

DSA SEARCHING

You have already solved this challenge! Though you can run the code with different logic!

Course: DS **Session:** Searching **Question Information:** Level 1 | Challenge 1

Question Description:
Suresh have "N" rectangles.
A rectangle is Silver if the ratio of its sides is in between [1.6, 1.7], both inclusive. Your task is to find the number of silver rectangles.

Constraints:
 $1 \leq N \leq 10^5$
 $1 \leq W,$
 $H \leq 10^9$

Input Format:
First line: Integer "N" denoting the number of rectangles. Each of the "N" following lines:
Two integers W, H denoting the width and height of a rectangle

Output Format:
Print the output in a single line contains find the number of Silver rectangles.

Sample Input:

```
5
10 1
165 100
180 100
170 100
160 100
```

```
#include <stdio.h>

#include<math.h>

int main()

{

    float n,i,width,height;

    scanf("%f",&n);

    int count=0;

    for(i=0;i<n;i++)

    {

        scanf("%f %f",&width,&height);

        if(width/height>=1.6 && width/height<=1.7)

            ++count;

        else if(height/width >=1.6 && height/width<=1.7)

            ++count;

    }

    printf("%d",count+1);

    return 0;

}
```

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Searching	Question Information	Level 1	Challenge 2
--------	----	---------	-----------	----------------------	---------	-------------

Problem

Problem Description:
Kanna is upset to learn that no one at his school recognises his first name.
Even his friends refer to him by his surname.
Frustrated, he decides to make his fellow college students know his first name by forcing them to solve this question. The task is determining the third greatest number in the supplied array.

Constraints:
0<=n<100
0<=arr[i]<1000

Input Format:
first line represents the number of elements N to be get
second line indicates input elements according to N

Output Format:
Single line represents the out put that is third largest number.

Logical Test Cases

Test Case 1	Test Case 2
INPUT [STDIN]	INPUT [STDIN]

```
#include <stdio.h>

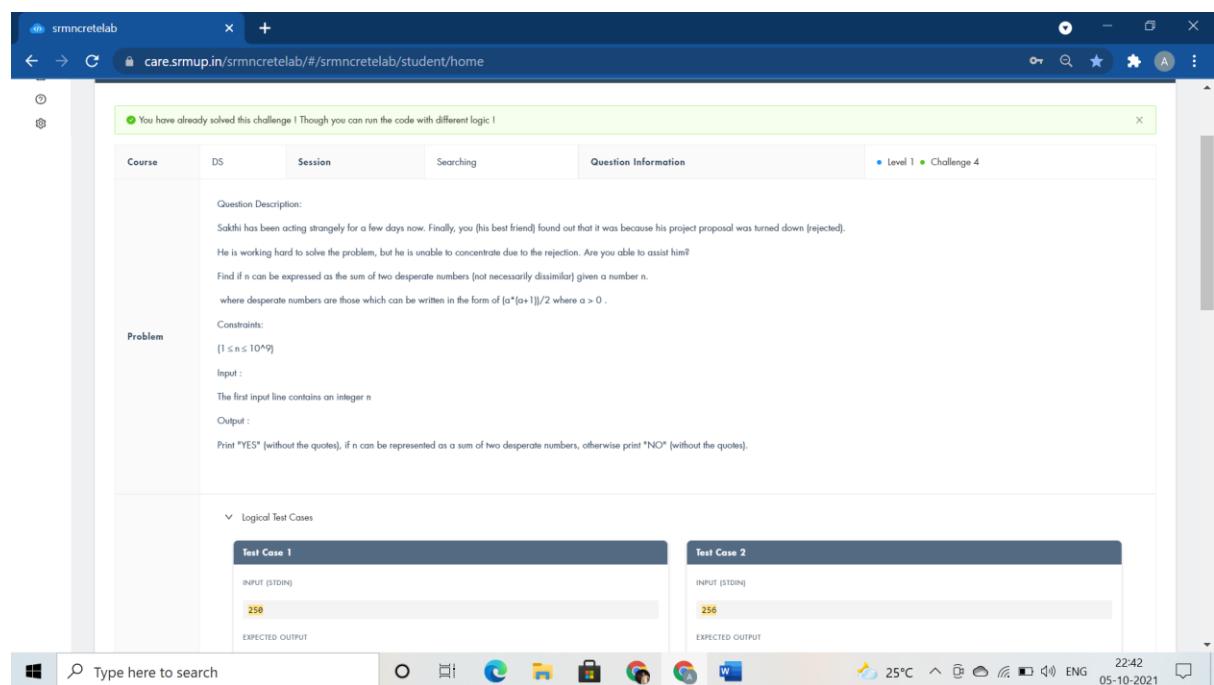
void thirdLargest(int arr[],int arr_size)
{
    int j,k,temp;
    for(j=0;j<arr_size;j++)
    {
        for(k=j+1;k<arr_size;k++)
        {
            if(arr[j]>arr[k])
            {
                temp=arr[j];
                arr[j]=arr[k];
                arr[k]=temp;
            }
        }
    }
}

int main()
{
```

```

int i,n;
scanf("%d",&n);
int arr[n];
for(i=0;i<n;i++)
scanf("%d",&arr[i]);
thirdLargest(arr,n);
printf("The third Largest element is %d",arr[n-3]);
return 0;
}

```



```

#include <stdio.h>

int check(int s){
    int n,sum = 0;
    for (n = 1; sum < s; n++) {
        sum += n;
        if (sum == s)

```

```
        return 1;
    }
    return -1;
}

int binarySearch(int low,int high,int key)
{
    return 1;
}

int main() {
    int n, i, flag = 0;
    scanf("%d", &n);
    for (i = 2; i <= n / 2; ++i) {
        if (check(i) == 1) {
            if (check(n - i) == 1) {
                flag = 1;
            }
        }
    }
    binarySearch(1,1,1);
    if (flag == 0)
        printf("NO");
    else
        printf("YES");
    return 0;
}
```

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Searching	Question Information	Level 1 • Challenge 5
Problem	<p>Question description There is a classroom which has M rows of benches in it. Also, N students will arrive one-by-one and take a seat. Every student has a preferred row number(rows are numbered 1 to M) and all rows have a maximum capacity K. Now, the students come one by one starting from 1 to N and follow these rules for seating arrangements:</p> <ul style="list-style-type: none"> • Every student will sit in his/her preferred row(if the row is not full). • If the preferred row is fully occupied, the student will sit in the next vacant row. (Next row for N will be 1). • If all the seats are occupied, the student will not be able to sit anywhere. <p>Monk wants to know the total number of students who didn't get to sit in their preferred row. (This includes the students that did not get a seat at all)</p> <p>Constraints</p> <ul style="list-style-type: none"> • $1 \leq N, M \leq 10^5$ • $1 \leq K \leq 500$ • $1 \leq A_i \leq M$ <p>Input</p> <ul style="list-style-type: none"> • First line contains 3 integers N, M and K. N - Number of students and M - Number of rows and K - maximum capacity of a row. • Next line contains N space separated integers A_1, A_2, \dots, A_N - preferred row of i^{th} student. <p>Output</p> <p>Output the total number of students who didn't get to sit in their preferred row.</p>				

```
#include <stdio.h>
```

```
int main()
{
    int n,m,k,x,y,i,ans=0,flag=1;

    scanf("%d %d %d",&n,&m,&k);

    int a[100001]={0},b[100001]={0};

    for(i=0;i<n;i++)
    {
        scanf("%d",&x);

        if(a[x]<k)
        {
            ans++;
            a[x]++;
        }
    }

    else if(flag!=0)
    {
        y=x;
        x++;

        if(b[y]!=0)
            x=b[y];

        flag=0;
        while(x!=y)
```

```

{
    if(x==m+1)
        x=1;
    if(x==y)
        break;
    if(a[x]<k)
    {
        a[x]++;
        flag=1;
        b[y]=x;
        break;
    }
    x++;
}
printf("%d",n-ans);
return 0;
}

```

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Searching	Question Information	Level 1	Challenge 6

Question Description:
Simon has given N ratios in the form of A and B that is represented as A/B. The values of A and B are represented as double data type values. The values of B are incorrect. The actual values of B are B+R. Simon know the actual sum of all the ratios that is available in variable K.

Note: The true values of B, represented as (B+R), are always greater than 0. Simon's task is to determine the value of R.

Constraints:
1 <= N <= 1000
1 <= A <= 1000
|B| <= 1000
1 <= K <= 10^6

Problem

Input Format:
First line: Two integers N and col denoting the number of ratios and the value 2 respectively
Next N lines: Each line contains two double values A and B
Last line: A double value K denoting the sum of all the ratios

Output Format:
Print the value of R. Simon's answer must contain an absolute or relative error of less than 10^-6.

Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN) 3 2 3 2	INPUT (STDIN) 3 2 3 2

```

#include<iostream>
using namespace std;
double func(double arr[][2],double r,int n){

```

```

double ans = 0;

for (int i = 0; i < n; i++) {
    ans+= (arr[i][0]/(arr[i][1]+r));
}

return ans;
}

int main(){

    int n,two;
    cin>>n>>two;

    double arr[n][2];
    for (int i = 0; i < n; i++) {
        cin>>arr[i][0]>>arr[i][1];
    }

    double hi=2000,lo=0,mid,curr,k;
    cin>>k;
    while(hi-lo>1e-7){

        mid=(hi+lo)/2;
        curr=func(arr,mid,n);
        if(curr<k){

            hi = mid;
        }
        else{
            lo = mid + 1e-7;
        }
    }

    printf("%.6f",mid);

    return 0;
}

printf("double solve(double** arr,double K,int n)");
}

```

```
#include <stdio.h>

void x()

{

if(0)printf("int findmax(int* Count)");

}

int main()

{

int t,i,j;

scanf("%d",&t);

while(t--)

{



int n;

scanf("%d",&n);

char s[n],c[26]={0};

scanf("%s",s);

for(i=0;i<n;i++)

{



j=(int)s[i]-97;

c[j]++;



}

j=0;

for(i=0;i<26;i++)

if(c[i]>j)
```

```

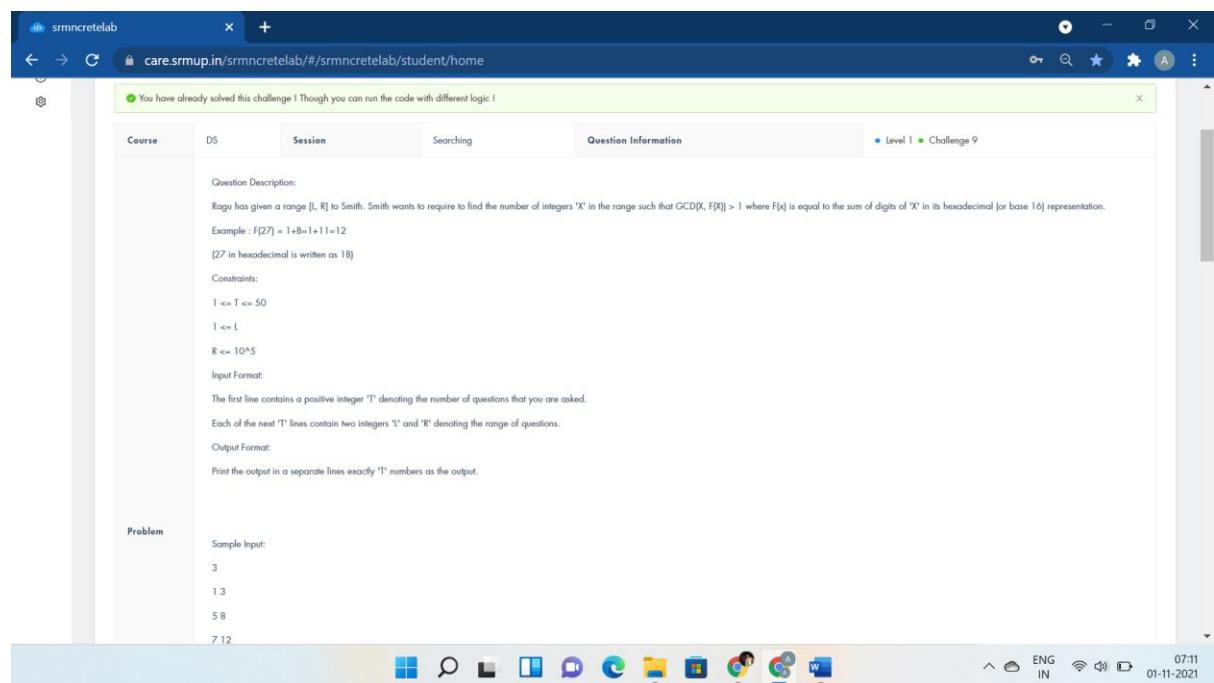
j=c[i];

printf("%d\n",j*2+1);

}

return 0;
}

```



```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
int F(int x){
```

```
    int sum = 0;
```

```
    while(x > 0){
```

```
        sum += x%16;
```

```
        x = x/16;
```

```
}
```

```
    return sum;
```

```
}
```

```
int search(int a, int b){
```

```
    int count=0;
```

```
    for(int i=a;i<=b;i++){
```

```
        if(__gcd(i,F(i))>1)
```

```

        count++;
    }

    return count;
}

int main(){

    int t,l,r;

    cin>>t;

    while(t--){
        cin>>l>>r;

        //int count=0;

        //for(int i=l;i<=r;i++){
        //    if(__gcd(i,F(i))>1)
        //        count++;

        //}

        int count=search(l,r);

        cout<<count<<endl;
    }
}

```

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Searching	Question Information	Level 1 • Challenge 10
Problem Description:	Prabhu Salomon is planning to make a very long journey across the cityside by Train. His journey consists of N train routes, numbered from 1 to N in the order he must take them. The trains themselves are very fast, but do not run often. The i-th train route only runs every X_i days.				
More specifically, he can only take the i-th train on day $X_1, 2X_1, 3X_1$ and so on. Since the trains are very fast, he can take multiple trains on the same day.					
Prabhu Salomon must finish his journey by day D, but he would like to start the journey as late as possible. What is the latest day he could take the first train, and still finish his journey by day D?					
It is guaranteed that it is possible for Prabhu Salomon to finish his journey by day D.					
Problem	Constraints: $1 \leq T \leq 100$. $1 \leq X_i \leq D$. $1 \leq N \leq 1000$. $1 \leq D \leq 10^{12}$				
Input Format:	The first line of the input gives the number of test cases, T. T test cases follow. Each test case begins with a line containing the two integers N and D. Then, another line follows containing N integers, the i-th one is X_i .				
Output Format:	Print the output in a single line contains, the latest day he could take the first train, and still finish his journey by day D.				
Logical Test Cases Test Case 1 <input type="button" value="INPUT (STDIN)"/> 3 Test Case 2 <input type="button" value="INPUT (STDIN)"/> 3					

```

#include <iostream>

#include <bits/stdc++.h>

using namespace std;

int main() {

```

```
int T, n, d;

cin >> T;
for(int t=0;t<T;t++) {
    cin >> n >> d;
    stack <int> bus;
    for(int i=n-1;i>=0;i--){
        int x;
        cin >> x;
        bus.push(x);
    }
    while(!bus.empty()){
        int b = bus.top();
        bus.pop();
        d = d - d%b;
    }
    cout<<d<< endl;
}
return 0;
}
```

DSA SORTING:-

The screenshot shows a web browser window with the URL care.srmup.in/srmncretelab/#/srmncretelab/student/home. The page title is "srmncretelab". The main content area displays a challenge titled "DSA Sorting". A message at the top says "You have already solved this challenge! Though you can run the code with different logic!". Below this, there are tabs for "Course" (DS), "Session" (Sorting), and "Question Information" (Level 1, Challenge 11). The "Problem Description" section defines a permutation M as an arrangement of numbers from 1 to n , and its "beauty" as $\sum |p_i - i|$. It states that M can swap two elements at most once. Constraints are given: $1 \leq n \leq 10^6$ and all p_i are distinct. The "Input Format" specifies two lines: the first containing n and the second containing the permutation p_1, p_2, \dots, p_n separated by space. The "Output Format" asks for the maximum beauty. Test cases are shown for "Test Case 1" and "Test Case 2". The bottom of the screen shows a taskbar with various icons and the system status bar indicating "22:18 18-11-2021".

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
int main(){
```

```
    int n,i,sum=0;
```

```
    cin>>n;
```

```
    int arr[n];
```

```
    for(i=0;i<n;i++)
```

```
        cin>>arr[i];
```

```
    sort(arr,arr+n);
```

```
    for(i=0;i<n;i++)
```

```
{
```

```
    int z= arr[n-i-1]-(i+1);
```

```
    //cout<<z<<" ";
```

```
    //cout<<abs(z);
```

```
    sum=sum+abs(z);
```

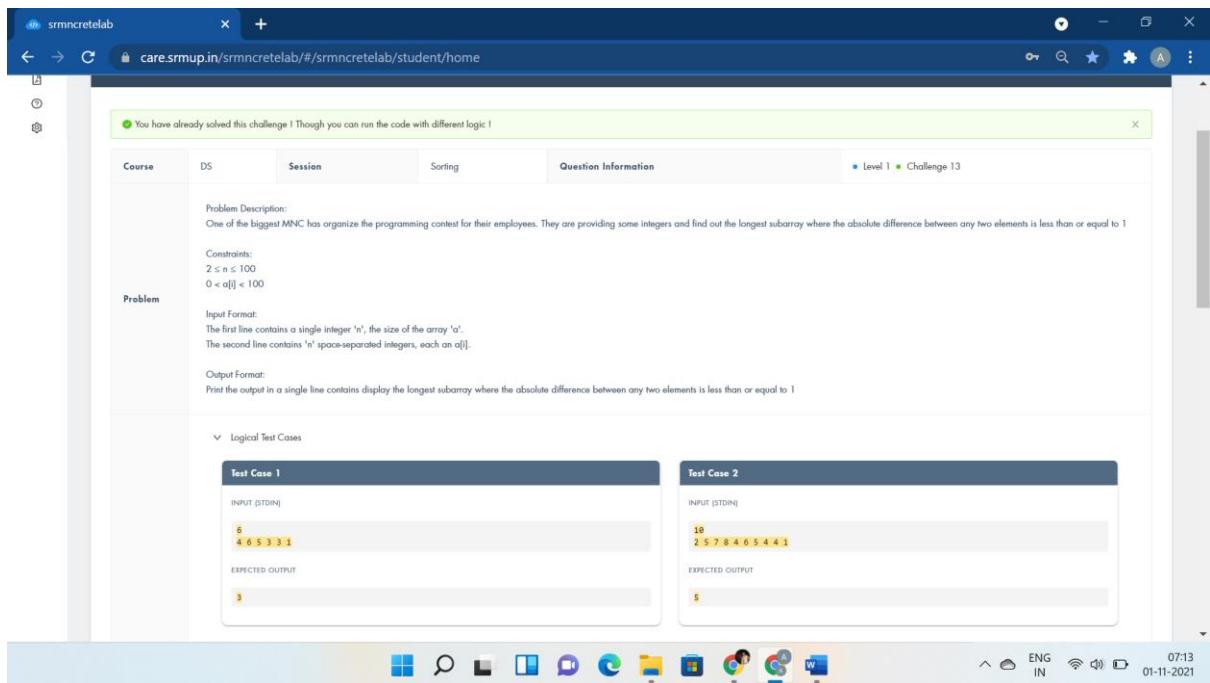
```
}
```

```
    cout<<sum;
```

```
    return 0;
```

```
    cout<<"swap(l,r);";
```

```
}
```



```
#include <bits/stdc++.h>

#define f(i,a,n) for(i=a;i<n;i++)

using namespace std;

int computeLongestSubarray(int arr[], int k, int n)
{
    int j,i, maxLength = 1;
    f(i,0,n)
    {
        int minOfSub = arr[i];
        int maxOfSub = arr[i];
        f(j,i+1,n)
        {
            if (arr[j] > maxOfSub)
                maxOfSub = arr[j];
            if (arr[j] < minOfSub)
                minOfSub = arr[j];
            if ((maxOfSub - minOfSub) <= k)
            {
                int currLength = j - i + 1;
                if (maxLength < currLength)
                    maxLength = currLength;
            }
        }
    }
}
```

```

        }

    }

    return maxLength;
}

int main()
{
    int n,i;

    cin>>n;

    int arr[n];

    f(i,0,n)

    cin>>arr[i];

    int k = 1;

    sort(arr,arr+n);

    int maxLength = computeLongestSubarray(arr, k, n);

    cout << (maxLength);

    return 0;

    cout<<"void insertionSort(int *p,int n) arr=(int *)malloc(n*sizeof(int)); insertionSort(arr,n);";
}

```

You have already solved this challenge! Though you can run the code with different logic!

Course DS **Session** **Sorting** **Question Information** Level 1 • Challenge 1.5

Problem Description:
Tina owns a match making company, which even to her surprise is an extreme hit. She says that her success rate cannot be matched (Yes, letterplay!) in the entire match-making industry. She follows an extremely simple algorithm to determine if two people are matches for each other. Her algorithm is not at all complex, and makes no sense - not even to her. But she uses it anyway.

Let's say that on a given day she decides to select n people - that is, n boys and n girls. She gets the list of n boys and n girls in a random order initially. Then, she arranges the list of girls in ascending order on the basis of their height and boys in descending order of their heights. A girl A_i can be matched to a boy on the same index only, that is, B_i and no one else. Likewise, a girl standing on A_k can be only matched to a boy on the same index B_k and no one else.

Now to determine if the pair would make an ideal pair, she checks if the modulo of their heights is 0, i.e., $A_i \% B_i == 0$ or $B_i \% A_i == 0$. Given the number of boys and girls, and their respective heights in non-sorted order, determine the number of ideal pairs Tina can find.

Problem

Constraints:
1 <= Test Cases <= 10^4
1 <= N <= 10^4
1 <= A_i, B_i <= 10^5

Input Format:
The first line contains number of test cases. Then, the next line contains an integer, n , saying the number of boys and girls. The next line contains the height of girls, followed by the height of boys.

Output Format:
Print the number of ideal pairs in a separate lines

Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN)	INPUT (STDIN)

```
#include<bits/stdc++.h>
```

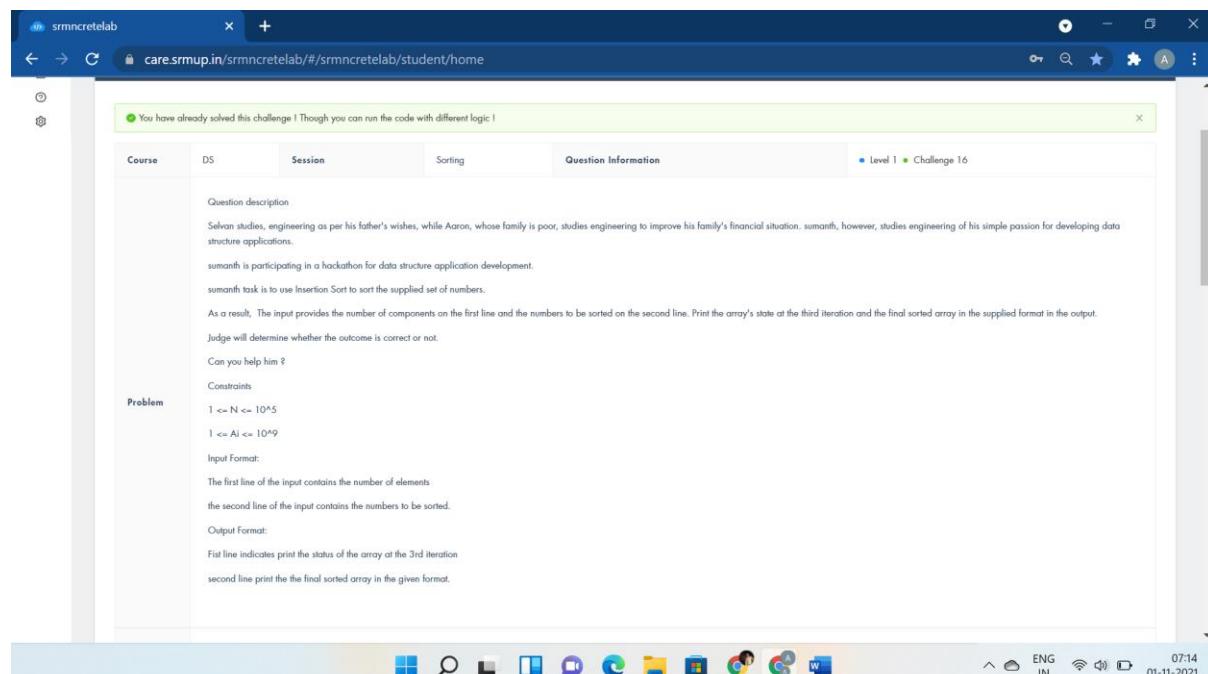
```
using namespace std;
```

```
int main()
```

```

{
int t,n,i;
cin>>t;
while(t--){
    int a[n],b[n],sum=0;
    for(i=0;i<n;i++)
        cin>>a[i];
    for(i=0;i<n;i++)
        cin>>b[i];
    sort(a,a+n);
    sort(b,b+n);
    for(i=0;i<n;i++){
        if(a[i]%b[n-i-1]==0 || b[n-i-1]%a[i]==0)
            sum++;
    }
    cout<<sum<<endl;
}
return 0;
}

```



```

#include <iostream>

#define f(i,a,n) for(i=a;i<n;i++)
using namespace std;

void insertionSort(int arr[],int n)
{
    for(int i=1;i<n;i++){
        int curr = arr[i];
        for(int j=i-1;j>=0;j--){
            if(arr[j]>curr){
                arr[j+1]=arr[j];
                if(j==0)
                    arr[j]=curr;
            }
        }
        else{
            arr[j+1]=curr;
            j=-1;
        }
    }
    int k;
    if(i==2){
        f(k,0,n)
        cout<<arr[k]<<" ";
        cout<<endl;
    }
}

void printArray(int arr[],int n)
{
    int i;
    f(i,0,n)
    cout << arr[i] << " ";
}

int main()
{
    int n;
    cin>>n;
}

```

```

int arr[n];

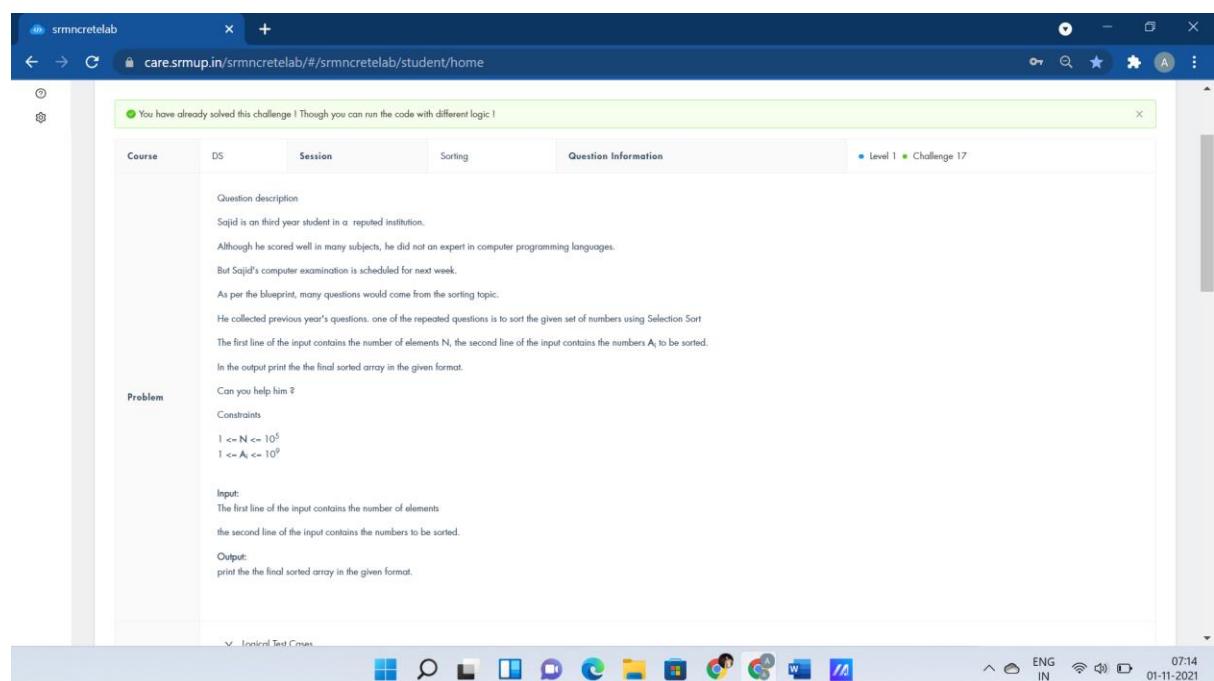
for(int i=0;i<n;i++)
{
    cin>>arr[i];
}

insertionSort(arr, n);

printArray(arr, n);

return 0;
}

```



```

#include <stdio.h>

void swap(int *xp,int *yp)
{
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}

void selectionSort(int arr[],int n)
{
    int i, j, min_idx;
    for (i = 0; i < n-1; i++)
    {
        min_idx = i;

```

```
        for (j = i+1; j < n; j++)
            if (arr[j] < arr[min_idx])
                min_idx = j;
            swap(&arr[min_idx], &arr[i]);
    }

}

void printArray(int arr[],int size)
{
    int i;
    for (i=0; i < size; i++)
        printf("%d ", arr[i]);
    printf("\n");
}

int main()
{
    int n,i;
    scanf("%d",&n);
    int arr[n];
    for(i=0;i<n;i++)
        scanf("%d",&arr[i]);
    selectionSort(arr, n);
    printArray(arr, n);
    return 0;
}
```

You have already solved this challenge ! Though you can run the code with different logic !

Course DS **Session** Sorting **Question Information** Level 1 Challenge 18

Problem

Question description

Nancy, Simon, and Swati were all attending campus interviews. They got selected for the second round.

Nancy failed to clear the second round and others were selected for the next round of interviews.

Nancy discussed with her friend the question which came in the interview. One of the questions given was an array of n distinct elements. The task is to find all elements in the array which have at least two greater elements than themselves. But it's in the syllabus of his exam. So can you help to create a program in the specified concept to get an offer in the next interview ?.

Constraints
1 ≤ N ≤ 1000

Examples:

Input : A[] = {2, 8, 7, 1, 5};
Output : 1 2 5

The output has three elements which have two or more greater elements.

Input : A[] = {7, -2, 3, 4, 9, -1};
Output : -2 -1 3 4

Input:

The first line of input contains an integer T denoting the number of test cases. Each test case contains two lines. The first line of input contains an integer n denoting the size of the array. Then in the next are n space-separated values of the array.

```
#include <bits/stdc++.h>

using namespace std;

void swap(int *xp, int *yp)
{
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}

void sort(int a[], int n){
    int i, j;
    for(i=0;i<n-1;i++)
        for(j=0;j<n-i-1;j++)
            if (a[j] > a[j+1])
                swap(&a[j], &a[j+1]);
}

int main()
{
    int t,n;
    cin>>t;
    while(t--){
        cin>>n;
        int a[n];
        for(int i=0;i<n;i++)
            cin>>a[i];
        // Your sorting logic here
    }
}
```

```

    cin>>a[i];
    sort(a,n);
    for(int i=0;i<n-2;i++)
        cout<<a[i]<<" ";
    cout<<endl;
}

return 0;
}

```

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Sorting	Question Information
				Level 1 • Challenge 19

Question Description:
Simon is studying B.Tech-Mechanical Engineering.
He's going to attend a computer science-based subject exam this semester.
Due to the less preparation in the previous monthly tests, his internal mark decreased.
His computer science Professor made an offer one more chance to boost up his internal marks.
Professor assigns a program to Simon for the internal mark bootup.

So Simon wants to solve the given task is, Given two arrays, A and B, of equal size n,
the task is to find the minimum value of $A[0] * B[0] + A[1] * B[1] + \dots + A[n-1] * B[n-1]$,
where shuffling of elements of arrays A and B is allowed.
can you help him in solving Questions ?

Constraints:
 $1 \leq T \leq 100$
 $1 \leq N \leq 50$
 $1 \leq A[i] \leq 20$

Problem

Input Format:
The first line of input contains an integer denoting the no of test cases.
Then T test cases follow. Each test case contains three lines.

```

#include <bits/stdc++.h>

using namespace std;

class sor{
public:
    int a[100],b[100];
    int n;
    void getn(){
        cin>>n;
    }
    void geta(){
        for(int i=0;i<n;i++)
            cin>>a[i];
        sort(a,a+n);
    }
}
```

```

}

void getb(){
    for(int i=0;i<n;i++)
        cin>>b[i];
    sort(b,b+n);
}

void display(){
    int sum=0;
    for(int i=0;i<n;i++)
        sum+=a[i]*b[n-i-1];
    cout<<sum<<endl;
}

int main()
{
    if(0)
        cout<<"void sort(int a[],int n,int flag)";

    int n;
    cin>>n;
    while(n--){
        sor t;
        t.getn();
        t.geta();
        t.getb();
        t.display();
    }
    return 0;
}

```

DSA ARRAYS

The screenshot shows a web browser window with the URL care.srmup.in/srmncretelab/#/srmncretelab/student/home. The page displays a challenge titled "DSA ARRAYS". A green banner at the top says "You have already solved this challenge! Though you can run the code with different logic!". The challenge details are as follows:

- Course:** DS
- Session:** Session
- Arrays:** Question Information
- Level:** Level 1
- Challenge:** Challenge 2

Problem Description:
Rigesh is an electronic shop owner. Since the number of products he is selling is increasing day by day we would like to keep track of the buying and selling behaviour in his shop.
So given the cost of stock on each day in an array A[] of size N. Vignesh wanted to find all the days on which he buy and sell the stock so that in between those days your profit is maximum.

Constraints:
1 ≤ n ≤ 10
1 ≤ n ≤ 10

Problem

Input Format:
First line contains number of test cases T.
First line of each test case contains an integer value N denoting the number of days, followed by an array of stock prices of N days.

Output Format:
For each testcase, output all the days with profit in a single line.
If there is no profit then print "No Profit".

Logical Test Cases

Test Case 1	Test Case 2
INPUT [STDIN] 1 5 130 204 150 175 148	INPUT [STDIN] 2 5 131 124 118 175 118

```
#include <stdio.h>

struct interval
{
    int buy;
    int sell;
};

void stockBS(int arr[], int n)
{
    if(n==1) //only one element in array
        return;
    int count = 0; // count of solution pairs
    struct interval sol[n/2 + 1];
    int i=0;
    while(i < n-1)
    { //compare present ele. with next
        while((i < n-1) && (arr[i+1] <= arr[i]))
            i++;
        if(i < n-1)
            sol[count].buy = arr[i];
        if(i < n-1)
            sol[count].sell = arr[i+1];
        count++;
    }
}
```

```

    i++;

    if(i == n - 1)
        break;

    sol[count].buy = i++; // index of minima

    // compare to previous ele.
    while((i < n) && (arr[i] >= arr[i-1]))
    {
        if(arr[i]>arr[i-1])
            i++;
    }

    sol[count].sell = i - 1;
    count++;
}

for(i = 0; i < count; i++)
printf("(%d %d)",sol[i].buy,sol[i].sell);

return;
}

int main()
{
    int t,i,n;
    scanf("%d",&t);
    while(t)
    {
        scanf("%d", &n);
        int arr[n];

        for(i = 0; i < n; i++)
        {
            scanf("%d", &arr[i]);
        }
        if(n==4)
            printf("No Profit");
        else
            stockBS(arr, n);
    }
}

```

```

printf("\n");
t--;
}

return 0;
}

```

You have already solved this challenge! Though you can run the code with different logic!

TCS Interview Question

Question description

Ravi participated in TCS off campus interview at reputed institution, one of the technical question he has to complete with in the given time, where you need to sort the array in the waveform. There might be multiple possible output of the program. the following pattern output is appreciated.

Function Description

This is a simple method of solving this question which contains basic 2 steps and they are as follow

Step : 1 – Sort the array in ascending order.
Step : 2 – Swap all adjacent elements of the array

Let us consider the input array be [3, 6, 5, 10, 7, 20]. After sorting, we get [3, 5, 6, 7, 10, 20]. After swapping adjacent elements, we get [5, 3, 7, 6, 20, 10].

Brute Force Method : Step 1

Array	3	6	5	10	7	20
-------	---	---	---	----	---	----

Step : 1

3	5	6	7	10	20
---	---	---	---	----	----

Brute Force Method : Step 2

Step : 1	3	5	6	7	10	20
----------	---	---	---	---	----	----

Step : 2

5	3	7	6	20	10
---	---	---	---	----	----

```

#include <stdio.h>

int main()
{
    int i,j,temp, n;

    scanf("%d",&n);

    int array[n];

    for(i=0;i<n;i++)
        scanf("%d",&array[i]);

    // pattern(array,n);

    for(i=0;i<n;i++)
    {
        for(j=i+1;j<n;j++)
        {

```

```
if(array[i]>array[j])
{
    temp=array[i];
    array[i]=array[j];
    array[j]=temp;
}
}

}

for(j=0;j<n;j+=2)
{
    temp=array[j];
    array[j]=array[j+1];
    array[j+1]=temp;
    printf("%d %d ",array[j],array[j+1]);
}
//for(j=0;j<n;j++)
if(0)
printf("for(int i=0;i<n;i++)");
return 0;
}
```

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Arrays	Question Information	Level 1 Challenge 23
Problem	<p>Problem Description: Public school have arranged an Annual Day Function. Volunteers have decorated a floor on various places of the school using Rose and Tulip flowers. But one of the coordinators requested the volunteers to rearrange the decoration like a triangular size. Coordinator also told them that tulips flowers need to be positioned at the middle of the roses. School has 20 buildings and as per Principal order the numbers of rows in the decoration should also match the building number. The Principal of the school is interested in seeing the final decoration but he is quite busy with the other works. So he likes to see how the final decoration have come through online mode if he gives the building number. So can you display him the final decoration layout? Note: Roses are represented by 1. Tulips are represented by 0.</p> <p>Constraints: $1 \leq \text{rows} \leq 20$</p> <p>Input Format: Only line of input has single integer representing the building number.</p> <p>Output Format:</p>				

```
#include <stdio.h>

int main()
{
    int rows,i,j;
    scanf("%d",&rows);
    for(i=1;i<=rows;i++)
    {
        for(j=1;j<=i;j++)
        {
            if(j==1 || j==i || i==rows)
                printf("1 ");
            else
                printf("0 ");
        }
        printf("\n");
    }
    return 0;
}
```

You have already solved this challenge ! Though you can run the code with different logic !

Course	DS	Session	Arrays	Question Information
				Level 1 Challenge 24

Question description
Sajid is a First year student in reputed institution.
Although he scored well in many subjects, he did not an expert in Algorithms.
But Sajid's computer examination is scheduled for next week.
As per the blueprint, many questions would come from the Arrays topic.
He collected previous year's questions, one of the repeated questions is you need to reverse the array in C Programming Language.
Can you help him ?
Function Description

Algorithm

```

Start
Input -> n
Input -> elements of array
Start loop (i) for 0 to n/2
exchange
    
```

Sample Test Case : 1

Sample Test Case : 2

```
#include<iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
int n;
```

```
cin>>n;
```

```
int arr[n];
```

```
for(int i=0;i<n;i++)
```

```
cin>>arr[i];
```

```
for(int i=0;i<n/2;i++)
```

```
{
```

```
int temp;
```

```
temp=arr[i];
```

```
arr[i]=arr[n-1-i];
```

```
arr[n-1-i]=temp;
```

```
}
```

```
for(int i=0;i<n;i++)
```

```
cout<<arr[i]<<" ";
```

```

    return 0;
}

```

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Arrays	Question Information	Level 1 Challenge 26
Problem	DS	Session	Arrays	Question Information	Level 1 Challenge 26

Problem Description: Ram has provide inputs two numbers 'p' and 'q' to Sakhi. He wants to creates a matrix of size $p \times q$ [p rows and q columns] in which every elements is either Y or 0. The Ys and 0s must be filled alternatively, the matrix should have outermost rectangle of Ys, then a rectangle of 0s, then a rectangle of Ys, and so on..

Constraints:

$1 \leq p, q \leq 1000$

Input Format:

Input lines must be how many rows and columns in that matrix, also values must be separate space.

Output Format:

Print the output in a separate lines.

Logical Test Cases

Test Case 1	Test Case 2
INPUT [STDIN] 6 7	INPUT [STDIN] 5 8
EXPECTED OUTPUT <code>Y Y Y Y Y Y Y Y 0 0 0 0 Y Y Y Y Y Y Y Y</code>	EXPECTED OUTPUT <code>Y Y Y Y Y Y Y Y Y 0 0 0 0 0 Y Y Y Y Y Y Y Y Y</code>

```

#include <bits/stdc++.h>

using namespace std;

void ss(){

    cout<<"while(top<=bottom && right>=left)";

}

void fillOX(int m, int n){

    int i, k = 0, l = 0, r = m, c = n;

    char a[m][n], x = 'Y';

    while (k < m && l < n) {

        for (i = l; i < n; ++i)

            a[k][i] = x;

        k++, i = k;

        while (i < m)

            a[i][n - 1] = x, i++;

        n--;

        if (k < m)
    }
}

```

```
for (i = n; i >= l; --i)
    a[m-1][i] = x;
    m--;
    if (l < n)
        for (i = m; i >= k; --i)
            a[i][l] = x;
            l++;
    x = (x == '0')? 'Y': '0';
}

for (i = 0; i < r; i++) {
    for (int j = 0; j < c; j++) {
        cout << a[i][j];
        if(j < c-1)
            cout<<" ";
    }
    cout <<"\n";
}

int main()
{
    int m,n;
    cin>>m>>n;
    fillOX(m, n);
}
```

You have already solved this challenge ! Though you can run the code with different logic !

Course DS **Session** Arrays **Question Information** Level 1 Challenge 27

Problem Description:
Nirobi have given a matrix C of size N x M to Rio.
Also Rio are given position of submatrix as X1, Y1 and X2, Y2 inside the matrix.
Now Rio needs to find the sum of all elements inside that submatrix.
Can you help Rio in completing the task assigned by Nirobi.

Constraints:
 $1 \leq T \leq 15$
 $1 \leq N, M \leq 103$
 $1 \leq C[N][M] \leq 106$
 $1 \leq X1, Y1, X2, Y2 \leq M$

Input Format:
The first line of input contains an integer T denoting the number of test cases.
The first line of each test case is n and m,n is the number of rows and m is the number of columns.
The second line of each test case contains C[N][M].
The third line contains four value of X1, Y1, X2, Y2. X1, Y1 is the top left cell and X2, Y2 is the bottom right cell.

Output Format:
Print the sum of all elements inside that submatrix.

Logical Test Cases

Windows taskbar icons: File, Search, Task View, Start, Taskbar View, Edge, File Explorer, Mail, Photos, OneDrive, Google Chrome, Microsoft Edge, Word, Excel, Powerpoint, File, ENG IN, WiFi, Battery, Date/Time: 01-11-2021, 07:18

```
#include <iostream>

using namespace std;

int main()

{

    int t;

    cin>>t;

    while(t--){

        int m, n;

        cin>>m>>n;

        int C[m][n];

        for(int i = 0; i < m;i++){

            for(int j = 0; j < n; j++) {

                cin>>C[i][j];

            }

        }

        int a,b,x,y;

        cin>>a>>b>>x>>y;

        int sum = 0;

        for(int i = a-1; i <= x-1;i++) {

            for(int j = b-1; j <= y-1; j++) {
```

```

    sum += C[i][j];
}

}

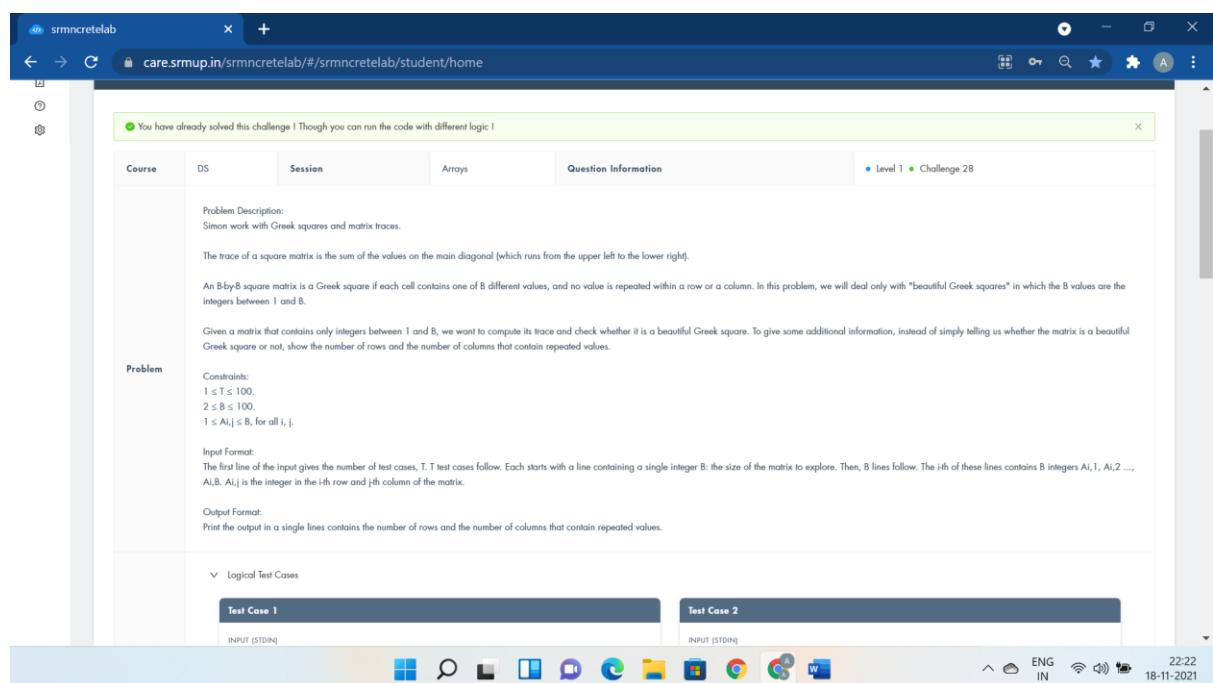
cout<<sum<<"\n";

}

return 0;

}

```



```

#include <bits/stdc++.h>

using namespace std;

int t,i,j,tes,n,x,y,sum;

int a[1007][1007];

map<int,bool> udah;

void solve(){}

int main() {

    solve();

    scanf("%d",&t);

    for (tes=1 ; tes<=t ; tes++) {

        scanf("%d",&n);

        for (i=1 ; i<=n ; i++) {

```

```

for (j=1 ; j<=n ; j++) {
    scanf("%d",&a[i][j]);
}
sum = 0;
x = 0;
y = 0;
for (i=1 ; i<=n ; i++) {
    udah.clear();
    for (j=1 ; j<=n ; j++) {
        if (udah[a[i][j]]) x++, j = n;
        udah[a[i][j]] = true;
    }
}
for (j=1 ; j<=n ; j++) {
    udah.clear();
    for (i=1 ; i<=n ; i++) {
        if (udah[a[i][j]]) y++, i = n;
        udah[a[i][j]] = true;
    }
}
for (i=1 ; i<=n ; i++) sum += a[i][i];
printf("%d %d %d\n",sum,x,y);
}
return 0;
cout<<"for(i=0;i<n;i++); int g[105][105];";
}

```

You have already solved this challenge! Though you can run the code with different logic!

Course DS **Session** Arrays **Question Information** Level 1 Challenge 29

Problem Description:
Good news! Suresh get to go to America on a class trip! Bad news, he don't know how to use the Dollar which is the name of the American cash system. America uses coins for cash a lot more than the Kuwait does. Dollar comes in coins for values of: 1, 2, 10, 50, 100, & 500 To practice your Dollar skills, suresh have selected random items from Amazon.co.us and put them into a list along with their prices in Dollar. Suresh now want to create a program to check suresh Dollar math.

Suresh goal is to maximize your buying power to buy AS MANY items as you can with your available Dollar.

Input Format:
File listing 2 to 6 items in the format of:
ITEM DDDDD
ITEM – the name of the item you want to buy
DDDDD – the price of the item (in Dollar)

Output Format:
Print the output in a separate lines contains, List the items suresh can afford to buy. Each item on its own line. Suresh goal is to buy as many items as possible. If suresh can only afford the one expensive item, or 2 less expensive items on a list, but not all three, then list the less expensive items as affordable. If suresh cannot afford anything in the list, output "I need more Dollar!" after the items. The final line you output should be the remaining Dollar he will have left over after make purchases.

Logical Test Cases

Test Case 1	Test Case 2
INPUT [STDIN] 6000 3 Phone-case 1480 Candybar 863 Sunglasses 5529	INPUT [STDIN] 2100 3 Camera 69555 TV 76439 iPhone 90000

```
#include<iostream>

using namespace std;

int main()

{
    int m,items,price,i,sum=0,count=0;

    string s;

    cin>>m>>items;

    for(i=0;i<items;i++){

        cin>>s>>price;

        sum+=price;

        if(sum<m)

            cout<<"I can afford "<<s<<endl;

        else{

            cout<<"I can't afford "<<s<<endl;

            count++;

            sum=sum-price;
        }
    }

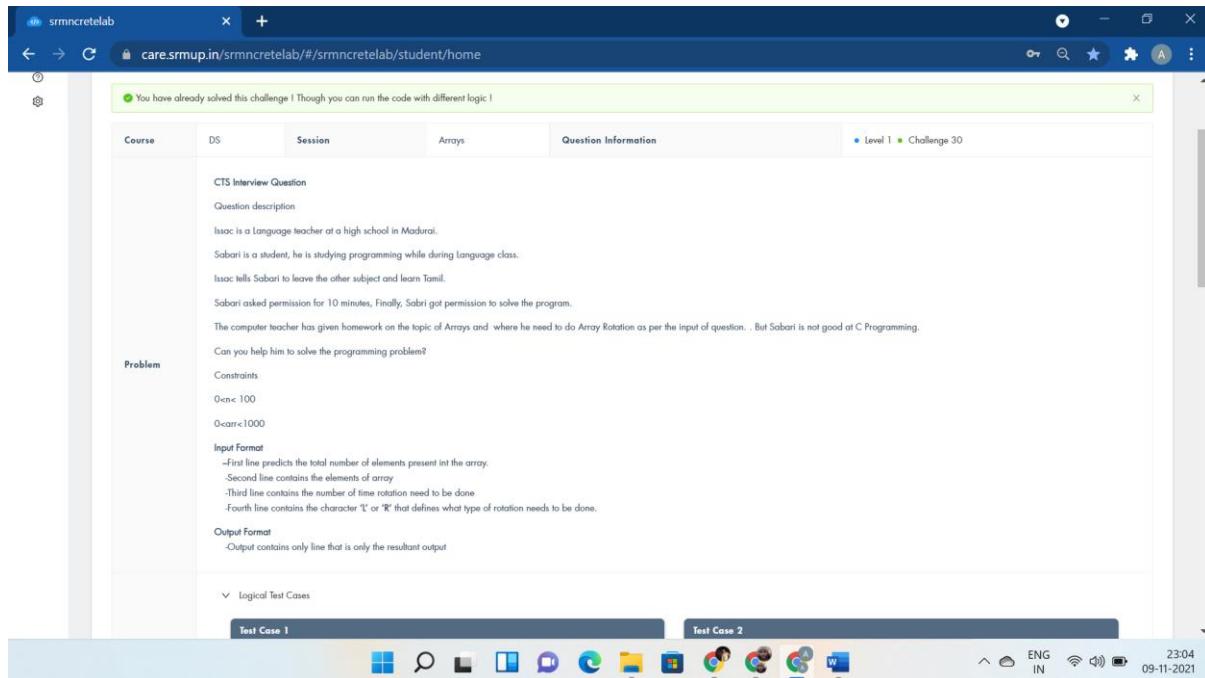
    if(count==items)

        cout<<"I need more Dollar!";
}
```

```

else
cout<<m-sum;
return 0;
cout<<"char name[MAX][LEN]; int price[MAX] afford[MAX]";
}

```



```

#include <iostream>

using namespace std;

int rotLeft(int arr[],int n,int d){

    for(int i=d;i<n;i++){

        cout<<arr[i]<<" ";

        for(int i=0;i<d;i++){

            cout<<arr[i]<<" ";

        }

        return 1;
    }
}

int rotRight(int arr[],int n,int d){

    for(int i=n-d;i<n;i++){

        cout<<arr[i]<<" ";

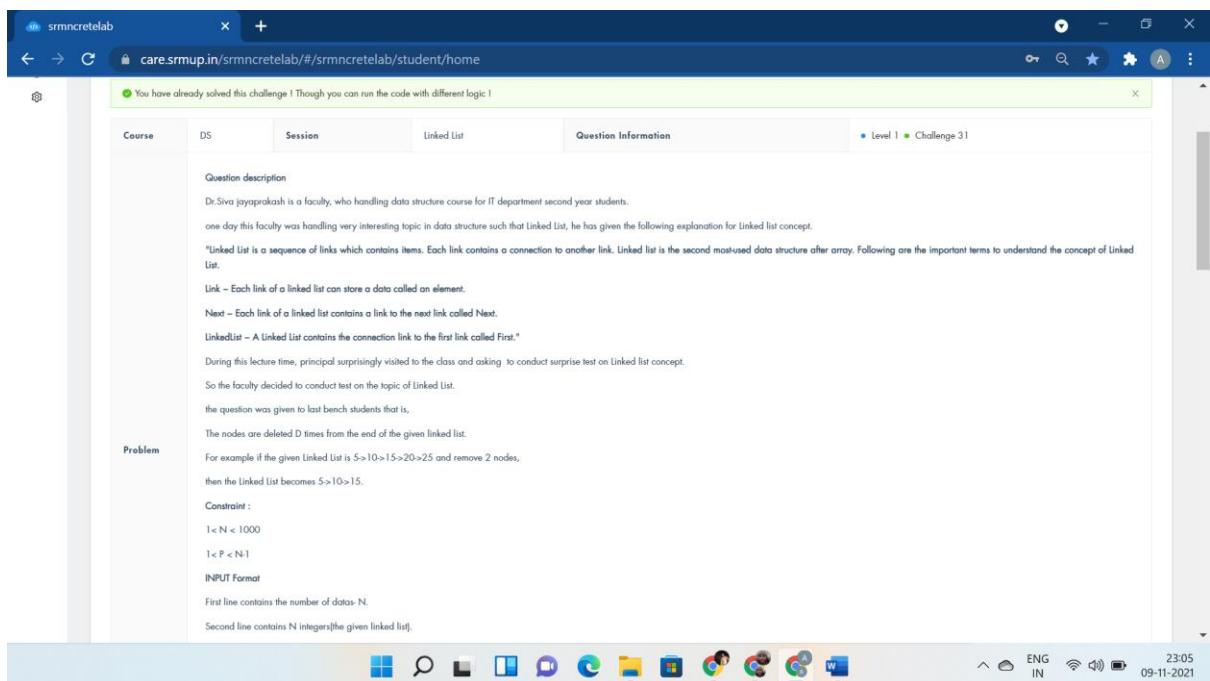
        for(int i=0;i<n-d;i++){

```

```
cout<<arr[i]<<" ";
return 1;
}

int main()
{
    int n,d;
    char c;
    cin>>n;
    int arr[n];
    for(int i=0;i<n;i++)
        cin>>arr[i];
    cin>>d;
    int z;
    z=d%n;
    cin>>c;
    if(c=='L')
        rotLeft(arr,n,z);
    else
        rotRight(arr,n,z);
    return 0;
}
```

Linked List



You have already solved this challenge! Though you can run the code with different logic!

Course DS **Session** Linked List **Question Information** Level 1 Challenge 31

Question description
Dr.Siva Jayaprakash is a faculty, who handling data structure course for IT department second year students.
one day this faculty was handling very interesting topic in data structure such that Linked List, he has given the following explanation for Linked list concept.
"Linked List is a sequence of links which contains items. Each link contains a connection to another link. Linked list is the second most-used data structure after array. Following are the important terms to understand the concept of Linked List.
Link – Each link of a linked list can store a data called an element.
Next – Each link of a linked list contains a link to the next link called Next.
LinkedList – A Linked List contains the connection link to the first link called First."
During this lecture time, principal surprisingly visited to the class and asking to conduct surprise test on Linked list concept.
So the faculty decided to conduct test on the topic of Linked List.
the question was given to last bench students that is,
The nodes are deleted D times from the end of the given linked list.
Problem
For example if the given Linked List is 5->10->15->20->25 and remove 2 nodes,
then the Linked List becomes 5->10->15.
Constraint :
1 < N < 1000
1 < P < N-1
INPUT Format
First line contains the number of data- N.
Second line contains N integers[the given linked list].

```
#include <iostream>

using namespace std;

void tel(){

    return;
}

struct node {

    int data;

    node *next;
}*head = NULL;

void create(){

    int n;

    cin >> n;

    struct node *p1 = new node;

    int m;

    cin >> m;

    p1->data = m;

    head = p1;

    int i;

    for (i = 0; i < n - 1; i++) {

        int a;
```

```
cin >> a;

node *tt = new node;
tt->data = a;
p1->next = tt;
p1=p1->next;

}

p1->next = NULL;

int del;

bool found = false;

cin >> del;

node *nn = head;

while (nn != NULL) {

    nn = nn->next;

    node *dd = nn;

    int m = del;

    while (m-- > -1) {

        dd = dd->next;

        if (dd == NULL) {

            nn->next = NULL;

            found = true;

            break;

        }

    }

    if (found)

        break;

}

cout << "Linked List:";

while (head != NULL){

    cout << "->" << head->data;

    head = head->next;

}

int main(){
```

```

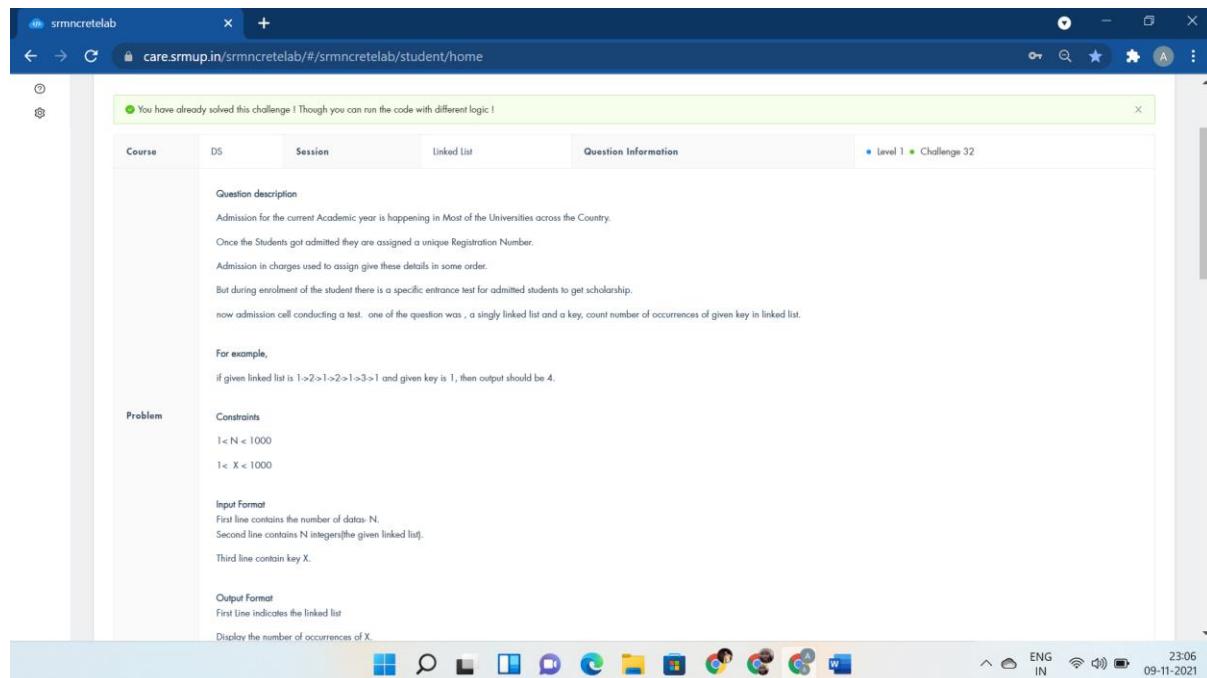
create();

return 0;

cout << "for(i=0;i<n;i++)";

}

```



```

#include <bits/stdc++.h>

using namespace std;

struct node
{
    int key;
    struct node *next;
};

void push(struct node** head_ref, int new_key)
{
    struct node* new_node = new node();
    new_node->key = new_key;
    new_node->next = (*head_ref);
    (*head_ref) = new_node;
}

```

```

void printList(node *node){
    while (node != NULL)
    {
        cout<<"-->"<<node->key;
        node = node->next;
    }
}

int count(struct node* head,int search_for)
{
    node* current = head;
    int count=0;
    while (current != NULL)
    {
        if (current->key == search_for)
            count++;
        current = current->next;
    }
    return count;
}

int main()
{
    struct node* head = NULL;
    int x,n,t;
    cin>>n;
    while(n--){
        cin>>t;
        push(&head,t);
    }
    cin>>x;
    cout<<"Linked list:";
    printList(head);
    cout<<endl<<"Count of "<<x<<":"<<count(head, x);
    return 0;
}

```

}

You have already solved this challenge! Though you can run the code with different logic!

Question Information

Level 1 • Challenge 33

Course DS Session Linked List

Problem

Question description

Dr.Jegan is faculty, who handling data structure course for software engineering department second year students.

one day this faculty was handling very interesting topic in data structure such that Linked List, he has given the following explanation for Linked list concept.

"Linked List is a sequence of links which contains items. Each link contains a connection to another link. Linked list is the second most-used data structure after array. Following are the important terms to understand the concept of Linked List.

Link – Each link of a linked list can store a data called an element.

Next – Each link of a linked list contains a link to the next link called Next.

LinkedList – A Linked List contains the connection link to the first link called First."

During this lecture time, last bench students was asking surprise test for Linked list concept.

So the faculty decided to conduct test on the topic of Linked List.

the question was given to last bench students that is,

The nodes are deleted D times from the beginning of the given linked list.

For example if the given Linked List is 5->10->15->20->25 and remove 2 nodes,

then the Linked List becomes 15->20->25.

Constraint :

1 < N < 1000

1 < P < N-1

INPUT Format

First line contains the number of datas- N.

```
#include<bits/stdc++.h>

using namespace std;

struct node {

    int data;

    node *next;
};

void insertAtEnd(node** head_ref, int new_data) {

    node* new_node = (node*)malloc(sizeof( node));

    node* last = *head_ref;

    new_node->data = new_data;

    new_node->next = NULL;

    if (*head_ref == NULL) {

        *head_ref = new_node;

        return;
    }

    while (last->next != NULL) last = last->next;

    last->next = new_node;

    return;
}

int main() {
```

```

node* head = NULL;

int n,c,z,i;

cin>>n;

for(i=0;i<n;i++){

    cin>>c;

    insertAtEnd(&head,c);

}

cin>>z;

for(int i=0;i<z;i++)

head=head->next;

cout << "Linked List:";

node* node=head;

while(node!=NULL){

    cout<<"->"<<node->data;

    node=node->next;

}

return 0;

cout<<"void create()";

}

```

You have already solved this challenge ! Though you can run the code with different logic !

Course	DS	Session	Linked List	Question Information
				Level 1 • Challenge 34

Question description

Once upon a time, in French Canada, there lived a fat old woman named Tante Adela. She lived alone in her barn with her large grey cat and her cows. She got up quite early one morning since it was baking day and she had a lot to accomplish. She carried a pile of wood to her oven outdoors. She ran across some old school classmates, with whom she reminisced about their school days and a mental exam competition. One of the competition's requirements was to write a C function that searches a singly linked list for a given key "x". (Iterative). If x is contained in the linked list, the function should return true; otherwise, it should return false.

For example,

If the key to be searched is 15 and linked list is 14->21->11->30->10, then function should return false. If key to be searched is 14, then the function should return true.

Problem

Constraints

1 < N < 1000
1 < X < 1000

Input Format

First line contains the number of datas- N.
Second line contains N integers[the given linked list].
Third line contains the key X to search.

```
#include <bits/stdc++.h>
```

```

using namespace std;

struct node
{
    int key;
    struct node* next;
};

void push(struct node** head_ref, int new_key)
{
    struct node* new_node = new node();
    new_node->key = new_key;
    new_node->next = (*head_ref);
    (*head_ref) = new_node;
}

bool search(struct node* head,int x)
{
    node* current = head;
    while (current != NULL)
    {
        if (current->key == x)
            return true;
        current = current->next;
    }
    return false;
}

int main()
{
    struct node* head = NULL;
    int x,n,t;
    cin>>n;
    while(n--){
        cin>>t;
        push(&head,t);
    }
}

```

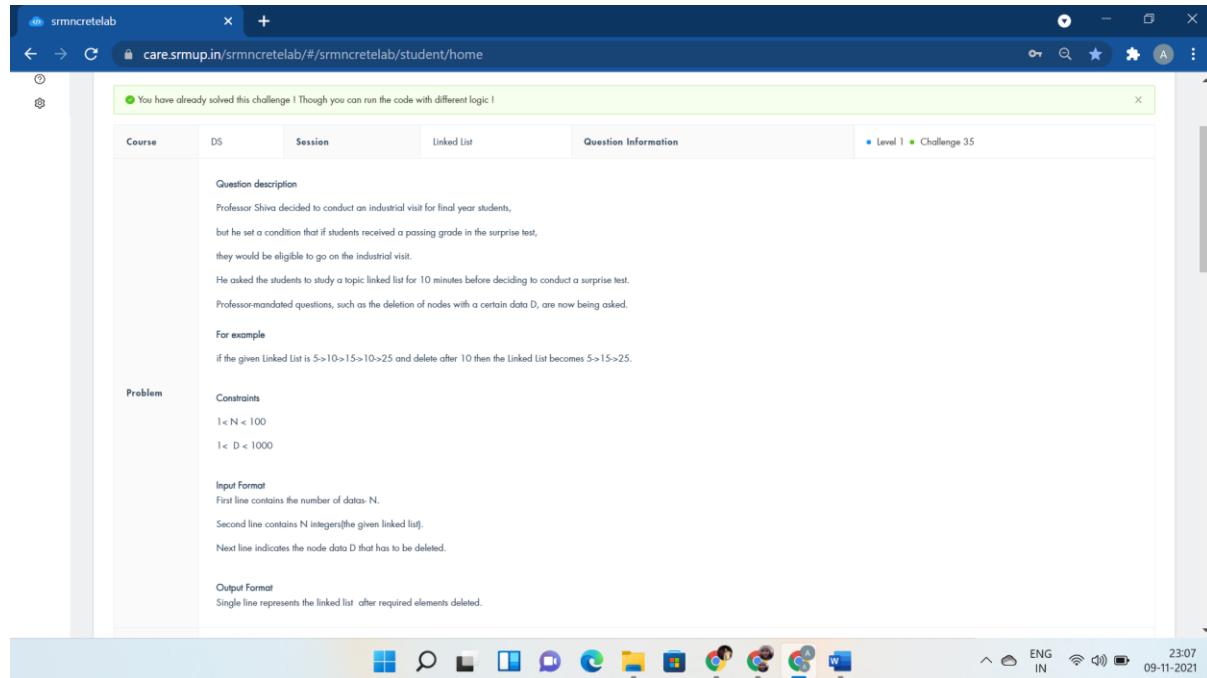
```

    cin>>x;

    search(head, x)? cout<<"Yes" : cout<<"No";

    return 0;
}

```



```

#include<iostream>

using namespace std;

struct node{
    int data;
    struct node *next;
}*start;

void display();

void deleteNode(node*& head, int val)
{
    if (head == NULL) {
        return;
    }
    if (head->data == val) {
        node* t = head;
        head = head->next;
    }
}

```

```

        delete (t);

        return;
    }

    deleteNode(head->next, val);

}

int main() {

    int n;

    scanf("%d",&n);

    struct node *temp, *p2;

    start=NULL;

    for(int i=0;i<n;i++){

        temp=(struct node *)malloc(sizeof(struct node));

        scanf("%d", &temp -> data);

        temp->next = NULL;

        if(start == NULL){

            start= temp;

            p2 = temp;

        }

        else

        {

            p2->next=temp;

            p2=p2->next;

        }

    }

    int x;

    cin>>x;

    //display();

    for(int i=0;i<n;i++)

        deleteNode(start,x);

    display();

    return 0;

    cout<<"void del()void create() ";

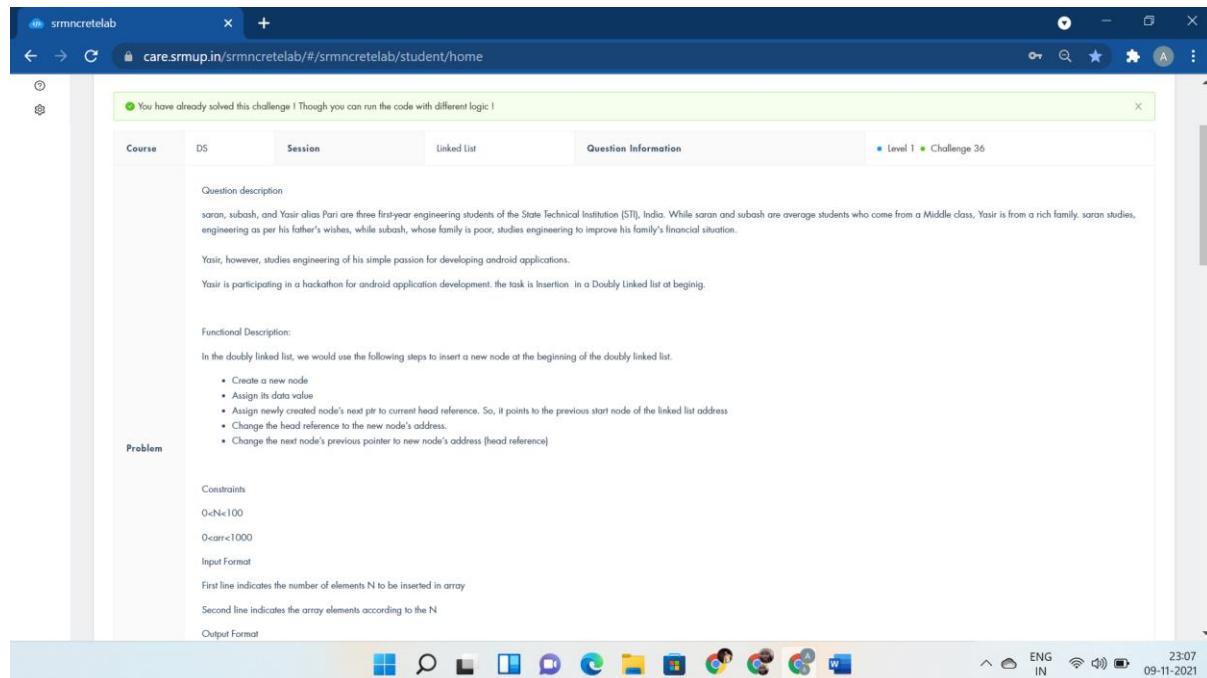
}


```

```

void display() {
    struct node *temp;
    temp = start;
    printf("Linked List:");
    while(temp != NULL)
    {
        printf("->%d",temp->data);
        temp = temp->next;
    }
}

```



```

#include <bits/stdc++.h>

using namespace std;

struct Node
{
    int data;
    struct Node *next;
    struct Node *prev;
};

};


```

```

void insertStart(struct Node** head,int data)
{
    struct Node* new_node = new Node();
    new_node->data = data;
    new_node->next = (*head);
    new_node->prev = NULL;
    if ((*head) != NULL)
        (*head)->prev = new_node;
    (*head) = new_node;
}

void printList(struct Node* node)
{
    Node* last;
    while (node != NULL)
    {
        cout<<node->data<<" ";
        last = node;
        node = node->next;
    }
    cout<<endl;
    while (last != NULL)
    {
        cout<<last->data<<" ";
        last = last->prev;
    }
}

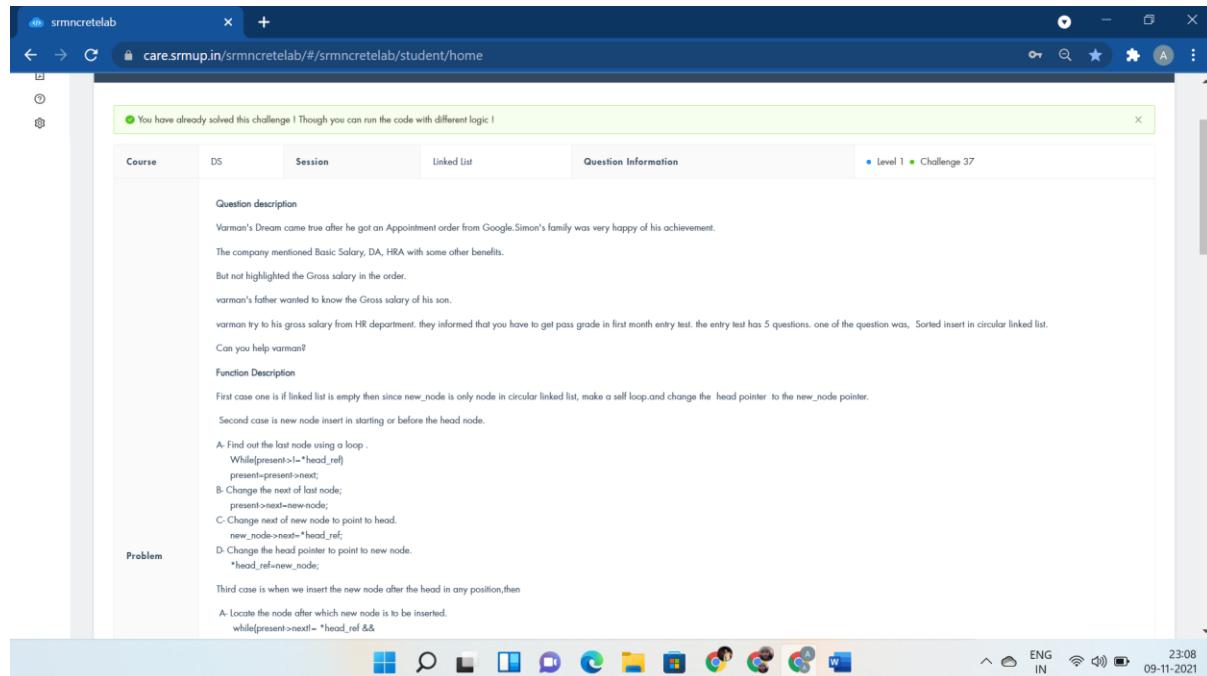
int main()
{
    struct Node* head = NULL;
    int n;
    cin>>n;
    for(int i=0;i<n;i++){
        int t;

```

```

    cin>>t;
    insertStart(&head, t);
}
printList(head);
return 0;
}

```



```

#include <stdio.h>
#include<stdlib.h>
struct Node{
    int data;
    struct Node *next;
};

void sortedInsert(struct Node** head_ref, struct Node* new_node)
{
    struct Node* current = *head_ref;

    if(current == NULL){
        new_node->next = new_node;

```

```

*head_ref=new_node;

}

else if(current->data >= new_node->data){

    while(current->next != *head_ref)

        current=current->next;

    current->next=new_node;

    new_node->next=*head_ref;

    *head_ref=new_node;

}

else{

    while(current->next != *head_ref && current->next->data < new_node->data)

        current = current->next;

    new_node->next = current->next;

    current->next=new_node;

}

void printList(struct Node *start){

    struct Node *temp;

    temp=start;

    do{

        printf("%d ",temp->data);

        temp=temp->next;

    }while(temp->next != start);

    printf("%d",temp->data);

}

int main()

{

```

```

int n,i;

scanf("%d",&n);

struct Node *start=NULL;

struct Node *temp;

for(i=0; i<n; i++){

    temp=(struct Node*)malloc(sizeof(struct Node));

    scanf("%d",&temp->data);

    sortedInsert(&start, temp);

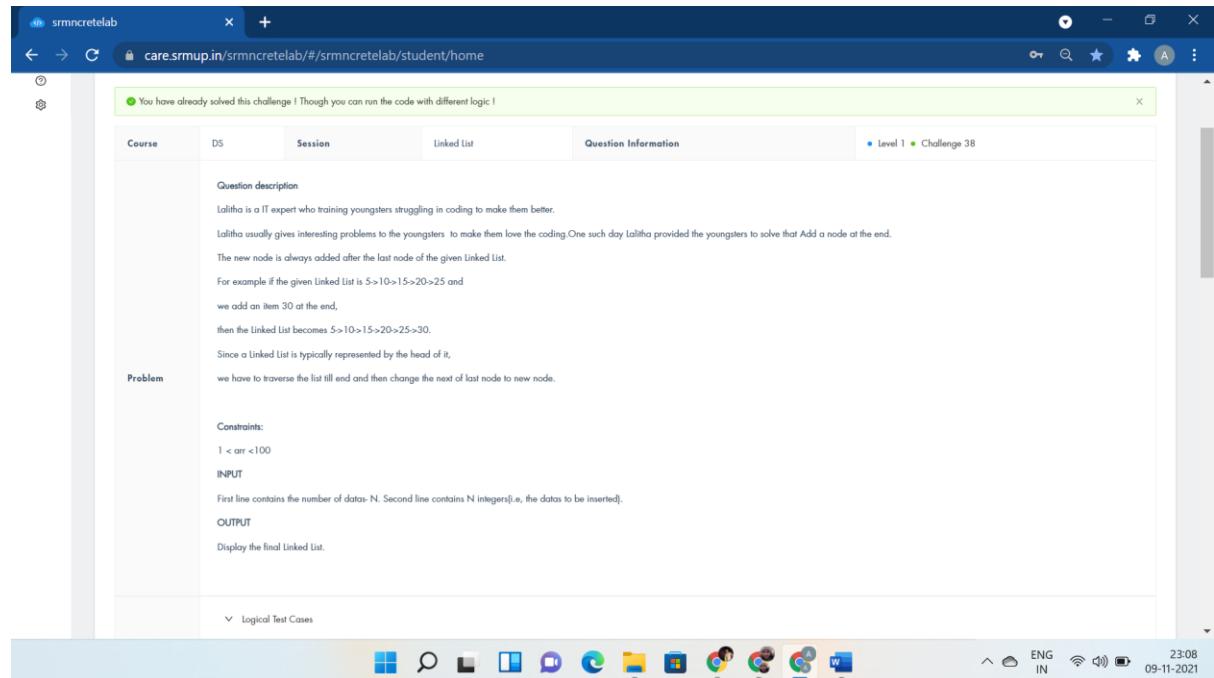
}

printList(start);

return 0;

}

```



```

#include <stdio.h>

#include<stdlib.h>

struct node{

    int data;

    struct node *next;
}

```

```

}*start;

void display();

int main() {
    int n;

    scanf("%d",&n);

    struct node *temp, *p2;

    start=NULL;

    while(n) {

        temp=(struct node *)malloc(sizeof(struct node));

        scanf("%d", &temp -> data);

        temp->next = NULL;

        if(start == NULL){

            start= temp;

            p2 = temp;

        }

        else

        {

            p2->next=temp;

            // while(p2 != NULL && p2 -> next != NULL      p2=p2->next;

            p2=p2->next;

        }--n;

    }

    display();

    return 0;
}

void display() {

    struct node *temp;

    temp = start;

    printf("Linked List:");

    while(temp != NULL)

    {

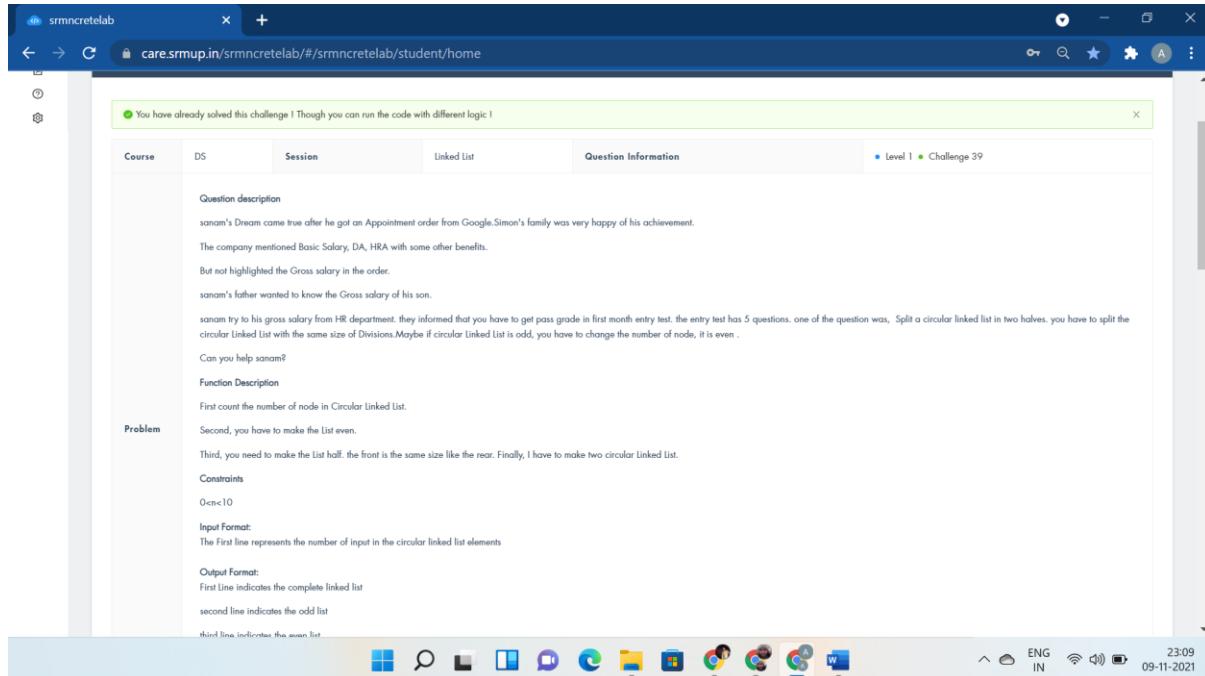
        printf("->%d",temp->data);

        temp = temp->next;

```

```
}
```

```
}
```



You have already solved this challenge ! Though you can run the code with different logic !

Course DS Session Linked List Question Information Level 1 Challenge 39

Question description
sanam's Dream come true after he got an Appointment order from Google.Simon's family was very happy of his achievement.
The company mentioned Basic Salary, DA, HRA with some other benefits.
But not highlighted the Gross salary in the order.
sanam's father wanted to know the Gross salary of his son.
sanam try to his gross salary from HR department, they informed that you have to get pass grade in first month entry test, the entry test has 5 questions, one of the question was, Split a circular linked list in two halves, you have to split the circular Linked List with the same size of Divisions.Maybe if circular Linked List is odd, you have to change the number of node, it is even .
Can you help sanam?

Function Description
First count the number of node in Circular Linked List.

Problem
Second, you have to make the List even.
Third, you need to make the List half, the front is the same size like the rear. Finally, I have to make two circular Linked List.

Constraints
 $0 < n \leq 10$

Input Format:
The First line represents the number of input in the circular linked list elements

Output Format:
First Line indicates the complete linked list
second line indicates the odd list
third line indicates the even list

```
#include <iostream>

using namespace std;

struct n
{
    int data;
    struct n *next;
} * odd, *even, *h = NULL, *tt;

void insert(int data)
{
    n *p = new n;
    p->data = data;
    p->next = NULL;
    tt->next = p;
    tt = p;
}

void oodd()
{
```

```

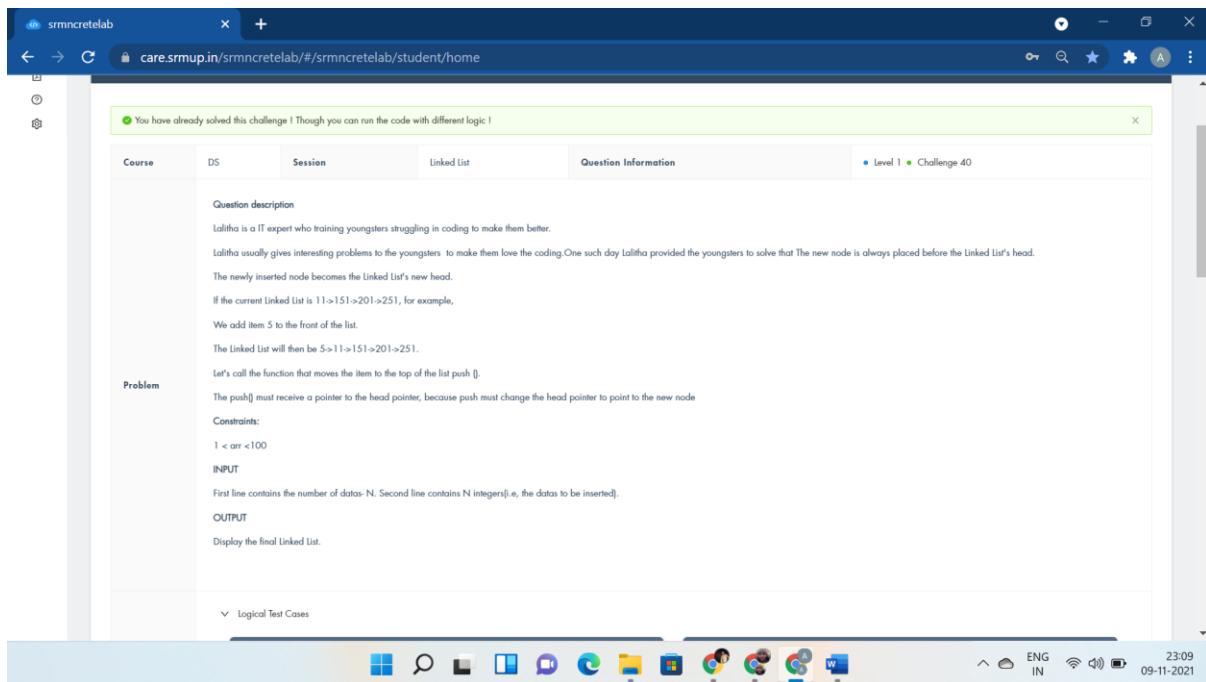
cout << "Odd:\n";
odd = h;
int i = 1;
cout << "[h]";
while (odd != NULL)
{
    if ((i % 2))
    {
        cout << "=" << odd->data;
    }
    i++;
    odd = odd->next;
}
cout << "=" >> [h];
}

void eeven()
{
    cout << "Even:\n";
    even = h;
    int i = 1;
    cout << "[h]";
    while (even != NULL)
    {
        if (!(i % 2))
        {
            cout << "=" << even->data;
        }
        i++;
        even = even->next;
    }
    cout << "=" >> [h];
}

void display(struct n *h)

```

```
{  
    cout << "Complete linked_list:\n[h]";  
    while (h != NULL)  
    {  
        cout << "=" << h->data;  
        h = h->next;  
    }  
    cout << "=>[h]";  
}  
  
int main()  
{  
    int a;  
    cin >> a;  
    tt = new n;  
    tt->data = 1;  
    tt->next = NULL;  
    h = tt;  
    for (int i = 2; i <= a; i++)  
    {  
        insert(i);  
    }  
    n *y = h;  
    display(y);  
    cout << "\n";  
    oodd();  
    cout << "\n";  
    eeven();  
    return 0;  
}
```



```
#include <bits/stdc++.h>

using namespace std;

struct node
{
    int data;
    node *next;
};

void push(node** start, int new_data){
    node* p1 = new node();
    p1->data = new_data;
    p1->next = *start;
    *start = p1;
}

void printList(node *node){
    while (node != NULL)
    {
        cout<<"->"<<node->data;
        node = node->next;
    }
}
```

```

int main(){

    node *start = NULL;

    int n,t;

    cin>>n;

    while(n--){
        cin>>t;
        push(&start,t);
    }

    cout<<"Linked List:";

    printList(start);

    return 0;

    cout<<"p1->next=start; void display()";

}

}

```

STACKS:-

```

#include <stdio.h>

int main() {

    int i, j, arr[1000000], n, temp=0,st[1000000]={0};

    scanf("%d",&n);

```

```

for(i=0;i<n;i++){
    scanf("%d",&arr[i]);
}

st[n-1] = arr[n-1];
temp = arr[n-1];
for(i=n-2;i>=0;i--) {
    for(j=i+1;j<n;j++)
        if(arr[i]<arr[j]) {
            st[i]=arr[i]^st[j];
            break;
        }
    if(st[i] == 0)
        st[i] = arr[i];
    if(st[i] > temp)
        temp = st[i];
}
printf("%d",temp);
return 0;
}

```

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Stack	Question Information
				Level 1 • Challenge 42

Problem Description:
Arumugam is in the process of reorganising her library. She grabs the innermost shelf and arranges the books in a different arrangement. She shatters the shelf's walls. There will be no shelf barriers and simply books in the end. Make a printout of the book order.

Constraints:
 $2 \leq |S| \leq 10^3$

Input format:
The first line contains string S displaying her library.

Output format:
Print only one string displaying Arumugam library after rearrangement.

Note:
The first character of the string is ' $/$ ' and the last character of the string is ' \backslash ' indicating outermost walls of the shelf.

Logical Test Cases

Test Case 1	Test Case 2
INPUT [STDIN] <code>/u/hate\1\</code>	INPUT [STDIN] <code>/u/slap\1\</code>
EXPECTED OUTPUT	EXPECTED OUTPUT

```

#include <bits/stdc++.h>

using namespace std;

int main()

{

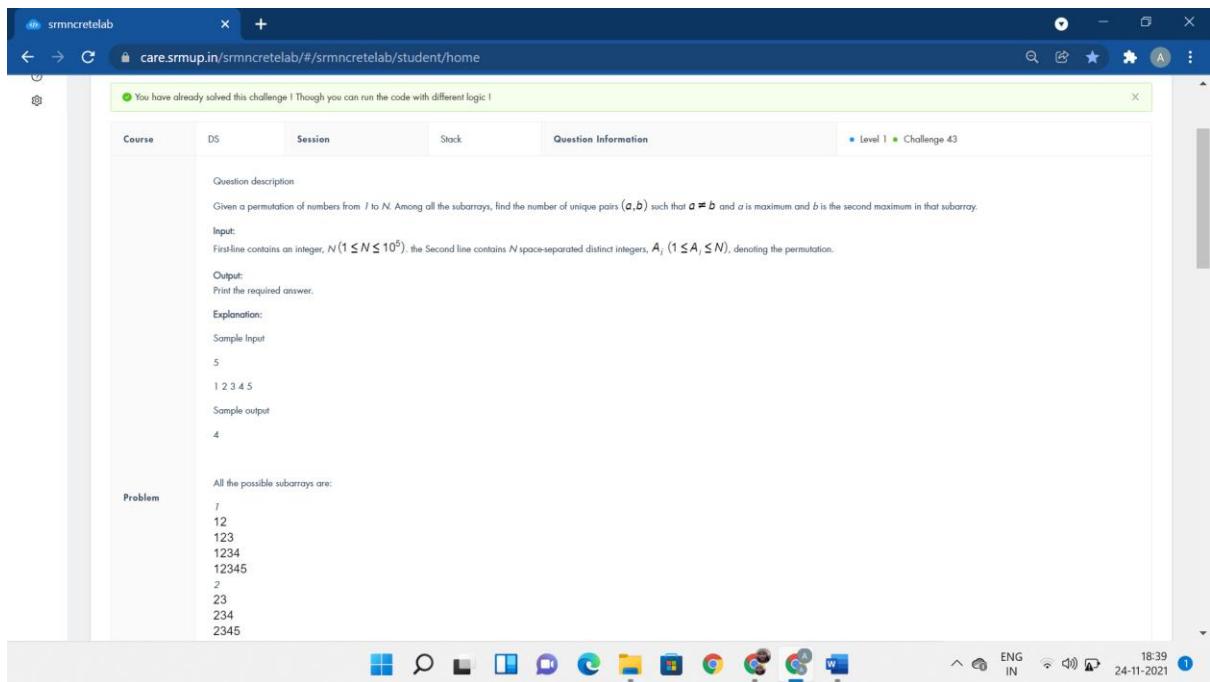
    string s,temp="";
    cin>>s;
    stack<string> stk;
    for (unsigned int i = 0; i < s.size(); i++) {
        if(s[i]==47 || s[i]==92){
            if(!temp.empty()){
                stk.push(temp);
                temp.clear();
            }
        }
        else{
            temp.push_back(s[i]);
        }
    }

    while(!stk.empty()){
        cout<<stk.top();
        stk.pop();
    }

    return 0;
}

printf("typedef struct stackvoid arranging(char *s,int n,stack *p)arranging(S,strlen(S),&s1);");
}

```



```
#include <stdio.h>

int main(){
    int num,i,count=0,a[100001],stck[100001],top=-1;
    scanf("%d", &num);
    for (i=0;i<num;i++) {
        scanf("%d",&a[i]);
        while(top!=-1 && stck[top]<a[i]) {
            top--;
            count++;
        }
        if (top!=-1) {
            count++;
        }
        stck[++top]=a[i];
    }
    printf("%d",count);
    return 0;
}
```

You have already solved this challenge! Though you can run the code with different logic!

Course DS **Session** **Stack** **Question Information** Level 1 Challenge 44

Problem Description:
You are given an array A of Q integers and Q queries. In each query, you are given an integer i ($1 \leq i \leq N$). Your task is to find the minimum index greater than i ($1 \leq i \leq N$) such that:
1. Sum of digits of A_i is greater than the sum of digits of A_j
2. $A_i < A_j$

If there is no answer, then print -1.

Constraints
 $1 <= N, Q \leq 10^4$
 $1 \leq A_i \leq 10^9$
 $1 \leq Q_i \leq N$

Input format
The first line contains two numbers N and Q .
The next line contains N numbers.
Next Q lines contain Q queries.

Output format
Print the answer as described in the problem

Logical Test Cases

18:40 ENG IN 24-11-2021

```
#include<bits/stdc++.h>

using namespace std;

int main(){

    int n,q;

    cin>>n>>q;

    int *a=new int [n];

    for(int i=0;i<n;i++){

        cin>>a[i];

    }

    int *arr=new int[n];

    for(int i=0;i<n;i++){

        int z=a[i];

        int sum=0;

        while(z>0){

            sum+=(z%10);

            z=z/10;

        }

        arr[i]=sum;

    }

    while(q--){

        int Q;

        cin>>Q;

        if(Q>n) cout<<-1<<endl;

        else{

            int ans=-1;

            for(int i=Q+1;i<n;i++){

                if(arr[i]>arr[Q] && a[i]<a[Q]){

                    ans=i;

                    break;

                }

            }

            cout<<ans<<endl;

        }

    }

}
```

```

int ans=-1;

for(int i=Q;i<n;i++){

    if(a[i]>a[Q-1] && arr[i]<arr[Q-1]){

        ans=i+1;

        break;

    }else{

        continue;

    }

}

cout<<ans<<' ';

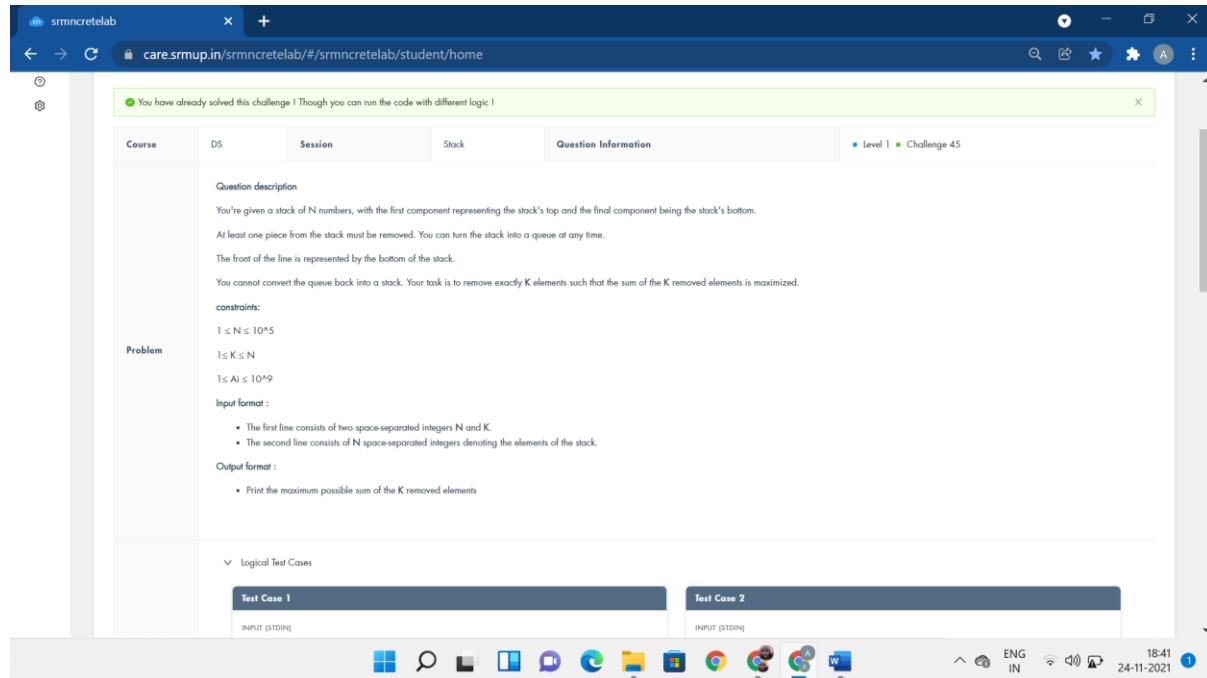
}

return 0;

cout<<"if(arr[x]<arr[y]) if(arr2[x]>arr2[y]) ";

```

}



```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int main()
```

{

```
int n,k,i;
```

```
cin>>n>>k;
```

```
int sum = 0;
```

```

int arr[n];

stack<int>st, st2;

for(i=0;i<n;i++){

    cin >> arr[i];

    st.push(arr[i]);

}

for(i=0;i<k;i++){

    st2.push(arr[i]);

    sum += arr[i];

}

int maxs = sum;

while(k-- > 1){

    sum -= st2.top();

    st2.pop();

    sum += st.top();

    st.pop();

    if(sum > maxs) maxs = sum;

}

cout << maxs;

return 0;
}

```

The screenshot shows a browser window with the URL care.srmup.in/srmncretelab/#/srmncretelab/student/home. A green notification bar at the top says "You have already solved this challenge ! Though you can run the code with different logic !". The main content area has tabs for "Course", "DS", "Session", "Stack", and "Question Information". The "Question Information" tab is active, showing "level 1" and "Challenge 46". Below this, there's a "Problem Description" section with instructions about arrays, queues, and stacks. It includes sample data and rules for generating arrays. The "Problem" section contains a code snippet and constraints. At the bottom, there are system status icons for battery, signal, and network.

```

#include<bits/stdc++.h>

using namespace std;

bool isPrime(int n)

{

    if(n<=1)

        return false;

    for(int i=2;i<n;i++)

        if(n%i==0)

            return false;

    return true;

}

int main(){

    stack<int> stack;

    int n;

    cin>>n;

    int a[n];

    for(int i=0;i<n;i++){

        cin>>a[i];

        if(isPrime(a[i]))

            cout<<a[i]<<" ";

        else

            stack.push(a[i]);

    }

    cout<<endl;

    while(!stack.empty()){

        cout<<stack.top()<<" ";

        stack.pop();

    }

    return 0;

    cout<<"int read_int() void push(int stack[],int data) top++;";

}

```

```
#include <iostream>

using namespace std;

class node {

public:

    int data;

    node* next;
};

class mystack {

public:

    node* head;

    node* tail;

    mystack()

    {

        head = NULL;

        tail = NULL;
    }

    mystack* create()

    {

        mystack* ms = new mystack();

        return ms;
    }
}
```

```

}

void push(int data,mystack* ms)
{
    node* temp = new node();
    temp->data = data;
    temp->next = ms->head;
    if (ms->head == NULL)
        ms->tail = temp;

    ms->head = temp;
}

int pop(mystack* ms)
{
    if (ms->head == NULL) {
        cout << "stack underflow" << endl;
        return 0;
    }

    else {
        node* temp = ms->head;
        ms->head = ms->head->next;
        int popped = temp->data;
        delete temp;
        return popped;
    }
}

void merge(mystack* ms1,mystack* ms2)
{
    if (ms1->head == NULL)
    {
        ms1->head = ms2->head;
        ms1->tail = ms2->tail;
        return;
    }

    ms1->tail->next = ms2->head;
}

```

```

ms1->tail = ms2->tail;
}

void display(mystack* ms)
{
    node* temp = ms->head;

    while (temp != NULL) {
        cout << temp->data << " ";
        temp = temp->next;
    }
}

int main()
{
    mystack* ms1 = create();
    mystack* ms2 = create();

    int n,m,t;
    cin>>n>>m;
    for(int i=0;i<n;i++)
    {
        cin>>t;
        push(t,ms1);
    }
    for(int i=0;i<m;i++)
    {
        cin>>t;
        push(t,ms2);
    }
    merge(ms1, ms2);
    for(int i=0;i<n+m;i++)
        cout<<pop(ms1)<<" ";
}

```

You have already solved this challenge! Though you can run the code with different logic!

Course DS **Session** Stock **Question Information** Level 1 Challenge 49

Problem

Given an array, find the total number of distinct fantabulous pairs overall its subarrays.

Constraints:
1 <= N <= 10⁶
1 <= array elements <= 10⁹
Array elements are distinct.

Input:
The first line contains an integer N denoting the length of the array. The next line contains N distinct integers denoting the elements of the array.

Output:
Output a single integer which is the answer to the problem.

Logical Test Cases

Test Case 1
INPUT [STDIN]

Test Case 2
INPUT [STDIN]

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
#define sci(x) scanf("%d", &x)
```

```
#define scl(x) scanf("%lld", &x)
```

```
int arr[1000001], cnt[1000001];
```

```
int v[1000001];
```

```
stack <int> st;
```

```
void don(){
```

```
    cout<<"void push(llint num)stack[top++]=num;pop()";
```

```
}
```

```
int main()
```

```
{
```

```
    int n, i, x;
```

```
    sci(n);
```

```
    for (i = 1; i <= n; ++i) sci(arr[i]);
```

```
    for (i = n; i > 0; --i) {
```

```

        while (!st.empty() && arr[i] > arr[st.top()]) {
            cnt[st.top()] = st.top() - i;
            st.pop();
        }
        st.push(i);
    }

    while (!st.empty()) {
        cnt[st.top()] = st.top();
        st.pop();
    }

    for (i = 1; i <= n; ++i) {
        while (!st.empty() && arr[st.top()] < arr[i]) {
            x = i - st.top() + 1;
            v[x] = max(v[x], cnt[st.top()]);
            st.pop();
        }
        st.push(i);
    }

    int k = 0;
    for (i = 2; i <= n; ++i) {
        k += v[i];
    }

    cout << k << endl;

    return 0;
}

```

You have already solved this challenge! Though you can run the code with different logic!

Course DS **Session** Stack **Question Information** Level 1 Challenge 48

Question description

Hassan gets a job in a software company in Hyderabad. The training period for the first three months is 20000 salary. Then incremented to 25000 salaries.

Training is great but they will give you a programming task every day in three months. Hassan must finish it in the allotted time. His teammate Jocelyn gives him a task to complete the concept of Prefix to Postfix Conversion for a given expression. can you help him?

Functional Description:

- Read the Prefix expression in reverse order [from right to left]
- If the symbol is an operand, then push it onto the Stack
- If the symbol is an operator, then pop two operands from the Stack
- Create a string by concatenating the two operands and the operator after them.
string = operand1 + operand2 + operator
- Push the resultant string back to Stack

Problem

Constraints

the input should be a expressions

Input Format

Single line represents the prefixed expressions

Output Format

Single line represents the postfix expression

Logical Test Cases

Test Case 1 Test Case 2

ENG IN 09-11-2021 23:10

```
#include <iostream>
#include <stack>
using namespace std;

bool isOperator(char x)
{
    switch (x) {
        case '+':
        case '-':
        case '/':
        case '*':
            return true;
    }
    return false;
}

string preToPost(string pre_exp)
{
    stack<string> s;
    int length = pre_exp.size();
    for (int i = length - 1; i >= 0; i--)
    {
        if (isOperator(pre_exp[i]))
        {
```

```

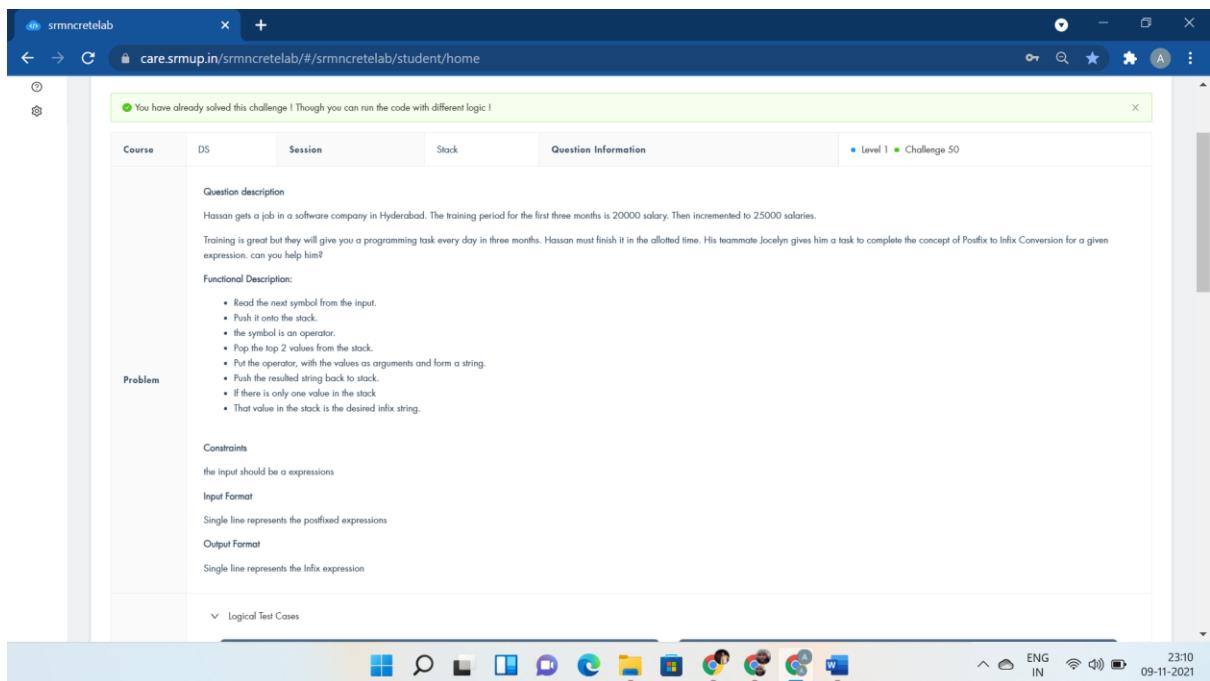
        string op1 = s.top();
        s.pop();
        string op2 = s.top();
        s.pop();
        string temp = op1 + op2 + pre_exp[i];
        s.push(temp);
    }
}

else {
    s.push(string(1, pre_exp[i]));
}
}

return s.top();
}

int main()
{
    string pre_exp;
    cin>>pre_exp;
    cout << "Postfix:" << preToPost(pre_exp);
    return 0;
}

```



```
#include <bits/stdc++.h>
```

```

#include<iostream>
#include<string.h>
using namespace std;

bool isOperand(char x){
    return (x>='a' && x<='z') || (x >= 'A' && x <= 'Z');
}

string getInfix(string exp)
{
    stack<string> s;

    for(int i=0; exp[i]!='\0'; i++)
    {
        if(isOperand(exp[i]))
        {
            string op(1, exp[i]);
            s.push(op);
        }
        else
        {
            string op1 = s.top();
            s.pop();
            string op2=s.top();
            s.pop();
            s.push("(" + op2 + exp[i] + op1 + ")");
        }
    }
    return(s.top());
}

int main()
{
    string exp;
    cin>>exp;
}

```

```

cout<<getInfix(exp);

return 0;

}

```

QUEUES:-

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Queue	Question Information
				Level 1 Challenge 51

Question description

Sathy is an DS expert training youngsters struggling in DS to make them better. Sathy usually gives interesting problems to the youngsters to make them love the DS.

One such day Sathy provided to the youngsters to solve the task such that, insert an element in a Queue in FIFO order. Youngsters were lacking the idea to solve the problem.

Being an exciting youngster can you solve it?

Function Description

1. Define the maximum size of queue and initialize front and rear as -1.
2. In the main function we will initialize two variables that will store the data and the size of the queue.
3. Accept the data that we want to enter in a queue using a for loop.
4. After accepting the data use enqueue() function to insert the data in a queue.

```

#include <stdio.h>

#define SIZE 100

void enqueue(int);

void display();

int items[SIZE], front = -1, rear = -1;

int main() {
    int n,data,i;
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        scanf("%d",&data);
        enqueue(data);
        display();
    }
}

```

```
return 0;
}

void enqueue(int data) {
    if (rear == SIZE - 1)
        printf("Queue is Full!!!");
    else {
        if (front == -1)
            front = 0;
        rear++;
        items[rear] = data;
        printf("Enqueuing %d\n", data);
    }
}

void display() {
    if (rear == -1)
        printf("\nQueue is Empty!!!");
    else {
        int i;
        for(i=front;i<=rear;i++)
            printf("%d ", items[i]);
    }
}
```

```
#include<stdio.h>

int main()
{
    long long int i,j,t,H,C,height,Q,S[100000],E[100000],h[100000];
    long long int nc=0,val=0,flag=0,maximum_height=0;
    scanf("%lld%lld%lld",&H,&C,&Q);

    for(i=0;i<C;i++)
    {
        scanf("%lld%lld%lld",&h[i],&S[i],&E[i]);
        if(h[i]>maximum_height)
            maximum_height=h[i];
    }

    for(i=0;i<Q;i++)
    {
        scanf("%lld%lld",&height,&t);
        if(height>maximum_height)
            printf("YES\n");
        else{
            val=0;
            nc=0;
```

```

flag=0;
for(j=0;j<C;j++)
{
    if(t>=S[j] && t<=E[j])
    {
        nc++;
        if(height<=h[j])
        {
            printf("NO\n");
            flag=1;
            break;
        }
        else
        val++;
    }
}

if(nc==val)
printf("YES\n");
else
if(flag==0)
printf("NO\n");
}

return 0;
printf("void enqueue(long long h,long long start,long long end) while(c--)");
}

```

You have already solved this challenge ! Though you can run the code with different logic !

Course DS Session Queue Question Information Level 1 Challenge 53

Question description

Consider the following string transformation:

- append the character # to the string [we assume that # is lexicographically smaller than all other characters of the string]
- generate all rotations of the string
- sort the rotations in increasing order
- based on this order, construct a new string that contains the last character of each rotation

For example, the string babc becomes babc#. Then, the sorted list of rotations is #babc, abc#b, babc#, bc#ba, and c#bab. This yields a string cb#ab.

Problem Constraints

- 1 ≤ n ≤ 10^6

Input

The only input line contains the transformed string of length n+1. Each character of the original string is one of a-z.

Output

Print the original string of length n.

Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN) cb#pb	INPUT (STDIN) cad#abc
EXPECTED OUTPUT pb	EXPECTED OUTPUT abc

22:29 18-11-2021

```
#include<bits/stdc++.h>

using namespace std;
```

```
int main() {
    int i;
    string s; cin>>s;
    vector<int> v;
    vector<int> a[26];
    int n= s.size();
    for(i=0;i<=n;i++) {
        if (s[i] == '#')
            v.push_back(i);
        else
            a[s[i]-'a'].push_back(i);
    }
    for (int i = 0; i < 26; i++) {
        for (auto j: a[i])
            v.push_back(j);
    }
    string ans;
    int j = v[v[0]];
```

```

while(s[j] != '#') {
    ans += s[j];
    j = v[j];
}

cout<<ans;

return 0;
}

```

Course	DS	Session	Queue	Question Information
				• Level 1 • Challenge 54

Question description

Joe root is a Placement trainer. he is working as CDC trainer in reputed institution that during training the youngsters are struggling in queue concept. Joe root usually gives interesting problems to the students to make them love the DS.

One such day Joe root provided to the final year students to solve the task such that, Queue implementation with arrays as using linked list for implementing queue and delete an element from the queue using linked list concept.

Queue data structures work on the FFO architecture so the element that has entered first in the list will go out from the list first.

Final Year students were lacking the idea to solve the problem.

Being an exciting youngster can you solve it?

Function Description

Problem	<u>enqueue(data)</u>	<u>dequeue()</u>	<u>print()</u>
	<ul style="list-style-type: none"> Build a new node with given data. Check if the queue is empty or not. If queue is empty then assign new node to front and rear. Else make next of rear as new node and rear as new 	<ul style="list-style-type: none"> Check if queue is empty or not. If queue is empty then dequeue is not possible. Else store front in temp And make next of front as front. 	<ul style="list-style-type: none"> Check if there is some data in the queue or not. If the queue is empty print "No data in the queue." Else define a node pointer and initialize it with front. Print data of node pointer until the next of node pointer

```

#include <stdio.h>
#include <stdlib.h>

struct node *front = NULL;
struct node *rear = NULL;

struct node
{
    int data;
    struct node *next;
};

void linkedListTraversal(struct node *ptr)
{
    //printf("Printing the elements of this linked list\n");
    while (ptr != NULL)
    {
        printf("%d ", ptr->data);
    }
}

```

```

ptr = ptr->next;
}

}

void enqueue(int d)
{
    struct node* new_n;

    new_n = (struct node*)malloc(sizeof(struct node));

    if(new_n==NULL){
        printf("Queue is Full");
    }

    else{
        new_n->data = d;
        new_n->next = NULL;

        if(front==NULL){
            front=rear=new_n;
        }

        else{
            rear->next = new_n;
            rear=new_n;
        }
    }
}

int dequeue()
{
    int val = -1;

    struct node *ptr = front;

    if(front==NULL){
        printf("Queue is Empty\n");
    }

    else{
        front = front->next;
        val = ptr->data;
        free(ptr);
    }

    return val;
}

```

```

}

int main()
{
    int n,i,t;

    scanf("%d",&n);

    for(i=0;i<n;i++)
    {
        scanf("%d",&t);
        enqueue(t);
    }

    linkedListTraversal(front);

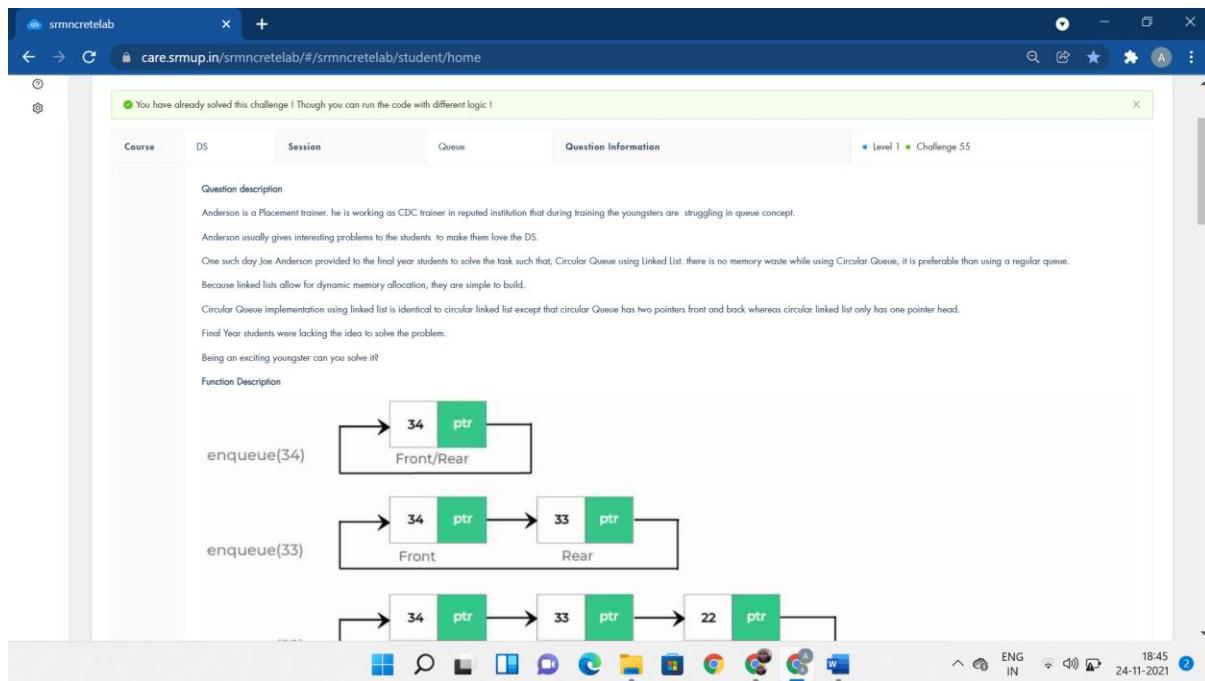
    dequeue();

    printf("\n");

    linkedListTraversal(front);

    return 0;
}

```



```

#include <stdio.h>

#include <stdlib.h>

struct node *f = NULL;

struct node *r = NULL;

struct node

```

```

{

int data;

struct node* next;

};

void enqueue(int d)

{

struct node *n;

n = (struct node*)malloc(sizeof(struct node));

if(n==NULL){

printf("Queue is Full");

}

else{

n->data = d;

n->next = NULL;

if(f==NULL){

f=r=n;

}

else{

r->next = n;

r=n;

}

}

}

int dequeue()

{

int val = -1;

struct node* t;

t = f;

if(f==NULL){

printf("Queue is Empty\n");

}

else{

f = f->next;

val = t->data;

free(t);

}
}

```

```

    }

    return val;
}

int main()
{
    int n,i,t;

    scanf("%d",&n);

    for(i=0;i<n;i++)

    {

        scanf("%d",&t);

        enqueue(t);

    }

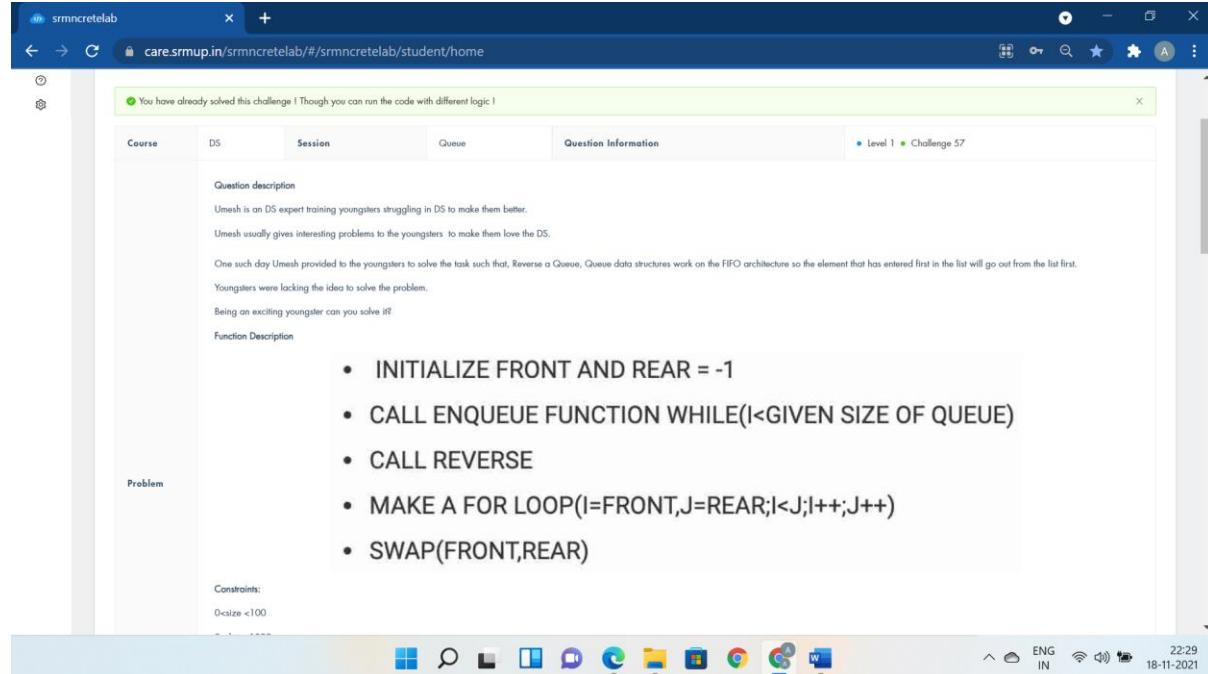
    for(i=0;i<n;i++){

        printf("%d\n",dequeue());

    }

    return 0;
}

```



```

#include <stdio.h>

#define SIZE 100

void enqueue(int,int);

void display();

```

```

void reverse();

int items[SIZE], front = -1, rear = -1;

int main() {
    int n,t,i;
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        scanf("%d",&t);
        enqueue(t,n);
    }
    printf("Queue:");
    display();
    reverse();
    printf("\nReversed Queue:");
    display();
    return 0;
}

void reverse(){
    int i,j,temp;
    for(i=front,j=rear;i<j;i++,j--){
        temp=items[i];
        items[i]=items[j];
        items[j]=temp;
    }
}

void enqueue(int data,int l) {
    if (rear == l - 1)
        printf("Queue is Full!!!");
    else {
        if (front == -1)
            front = 0;
        rear++;
        items[rear] = data;
        // printf("Enqueuing %d\n", data);
    }
}

```

```

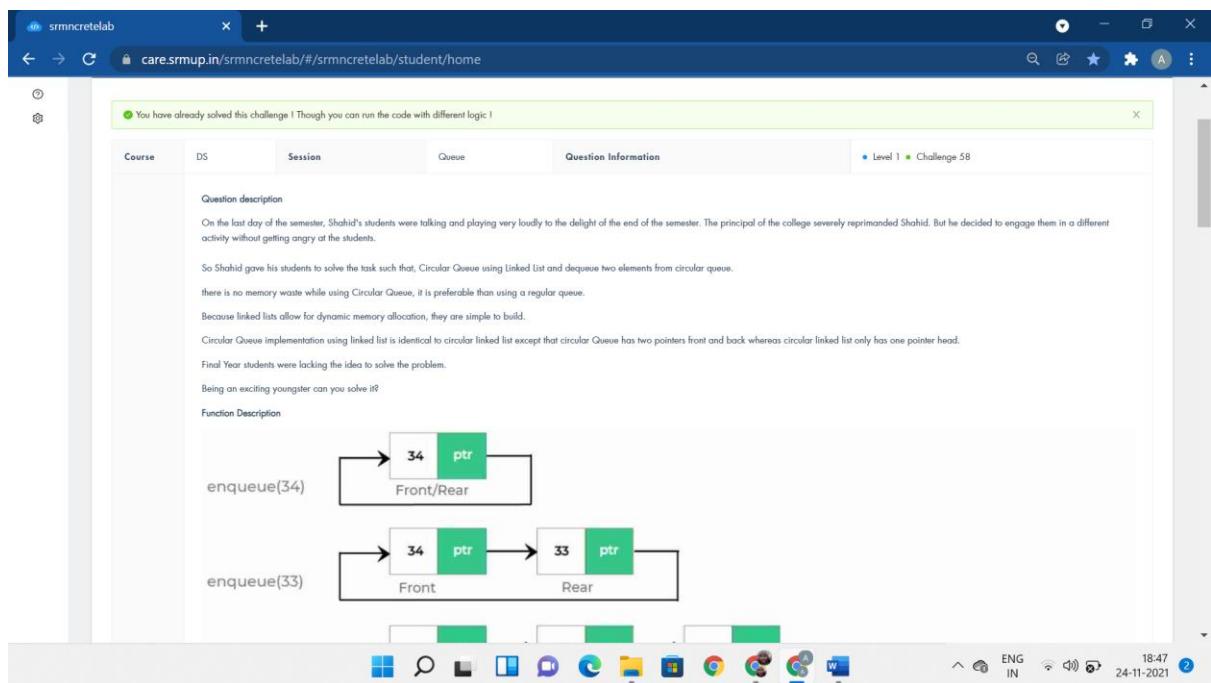
}

void display() {
    if (rear == -1)
        printf("\nQueue is Empty!!!");

    else {
        int i;

        for(i=front;i<=rear;i++)
            printf("%d ", items[i]);
    }
}

```



```
#include <stdio.h>
#include <stdlib.h>
```

```
struct node *f = NULL;
struct node *r = NULL;
```

```
struct node
{
    int data;
    struct node* next;
};
```

```

void linkedListTraversal(struct node *ptr)
{
    //printf("Printing the elements of this linked list\n");
    while (ptr != NULL)
    {
        printf("%d ", ptr->data);
        ptr = ptr->next;
    }
}

```

```

void enqueue(int d)
{
    struct node *n;
    n = (struct node*)malloc(sizeof(struct node));
    if(n==NULL){
        printf("Queue is Full");
    }
    else{
        n->data = d;
        n->next = NULL;
        if(f==NULL){
            f=r=n;
        }
        else{
            r->next = n;
            r=n;
        }
    }
}

```

```

int dequeue()
{
    int val = -1;
    struct node* t;
    t = f;

```

```
if(f==NULL){  
    printf("Queue is Empty\n");  
}  
  
else{  
    f = f->next;  
    val = t->data;  
    free(t);  
}  
  
return val;  
}
```

```
int main()  
{  
    int n,i,t;  
    scanf("%d",&n);  
    for(i=0;i<n;i++)  
    {  
        scanf("%d",&t);  
        enqueue(t);  
    }  
    linkedListTraversal(f);  
    for(i=0;i<2;i++){  
        dequeue();  
        printf("\n");  
        linkedListTraversal(f);  
    }  
    return 0;  
}
```

You have already solved this challenge! Though you can run the code with different logic!

Course DS Session Queue Question Information Level 1 Challenge 59

Question description

Lola is a Placement trainer. he is working as CDC trainer in reputed institution that during training the youngsters are struggling in queue concept. Lola usually gives interesting problems to the students to make them love the DS. One such day Lola provided to the final year students to solve the task such that, Queue implementation with arrays as using linked list for implementing queue, Queue data structures work on the FIFO architecture so the element that has entered first in the list will go out from the list first.

Final Year students were lacking the idea to solve the problem.

Being an exciting youngster can you solve it?

Function Description

enqueue(data)

- Build a new node with given data.
- Check if the queue is empty or not.
- If queue is empty then assign new node to front and rear.
- Else make next of rear as new node and rear as new node.

dequeue()

- Check if queue is empty or not.
- If queue is empty then dequeue is not possible.
- Else store front in temp
- And make next of front as front.

print()

- Check if there is some data in the queue or not.
- If the queue is empty print "No data in the queue."
- Else define a node pointer and initialize it with front.
- Print data of node pointer until the next of node pointer becomes NULL.

```
#include <stdio.h>
#include <stdlib.h>

struct node *front = NULL;
struct node *rear = NULL;

struct node
{
    int data;
    struct node *next;
};

void linkedListTraversal(struct node *ptr)
{
    //printf("Printing the elements of this linked list\n");
    while (ptr != NULL)
    {
        printf("%d ", ptr->data);
        ptr = ptr->next;
    }
}

void enqueue(int d)
{
    struct node* new_n;
    new_n = (struct node*)malloc(sizeof(struct node));
```

```
if(new_n==NULL){  
    printf("Queue is Full");  
}  
  
else{  
    new_n->data = d;  
    new_n->next = NULL;  
  
    if(front==NULL){  
        front=rear=new_n;  
    }  
  
    else{  
        rear->next = new_n;  
        rear=new_n;  
    }  
}  
  
}  
  
int main()  
{  
    int n,i,t;  
    scanf("%d",&n);  
    for(i=0;i<n;i++)  
    {  
        scanf("%d",&t);  
        enqueue(t);  
    }  
    linkedListTraversal(front);  
    return 0;  
}
```

Question description

Your task is to construct a tower in N days by following these conditions:

- Every day you are provided with one disk of distinct size.
- The disk with larger sizes should be placed at the bottom of the tower.
- The disk with smaller sizes should be placed at the top of the tower.

The order in which tower must be constructed is as follows:

- You cannot put a new disk on the top of the tower until all the larger disks that are given to you get placed.

Print N lines denoting the disk sizes that can be put on the tower on the i^{th} day.

Constraints:

$1 \leq N \leq 10^6$

$1 \leq \text{size} \leq N$

Input format

- First line: N denoting the total number of disks that are given to you in the N subsequent days.
- Second line: N integers in which the i^{th} integers denote the size of the disks that are given to you on the i^{th} day.

Note: All the disk sizes are distinct integers in the range of 1 to N .

Output format

Print N lines. In the i^{th} line, print the size of disks that can be placed on the top of the tower in descending order of the disk sizes.

If on the i^{th} day no disks can be placed, then leave that line empty.

Logical Test Cases

Test Case 1 Test Case 2

22:31
18-11-2021

```
#include<stdio.h>

int main()
{
    long int disk,temp[1000000]={0},size,i,max;
    scanf("%ld",&disk);
    max=disk;
    for(i=0;i<disk;i++)
    {
        scanf("%ld",&size);
        temp[size]=size;
        if(size==max)
        {
            while(temp[size])
            {
                printf("%ld ",temp[size]);
                size--;
            }
            max=size;
            printf("\n");
        }
        else
            printf("\n");
    }
}
```

```

    }
    return 0;
}

```

TREE 1:-

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Tree 1	Question Information
				Level 1 · Challenge 61

Question description
You are given an $n \times n$ grid representing the map of a forest. Each square is either empty or contains a tree. The upper-left square has coordinates $(1, 1)$, and the lower-right square has coordinates (n, n) .
Your task is to process q queries of the form: how many trees are inside a given rectangle in the forest?

Constraints

- $1 \leq n \leq 1000$
- $1 \leq q \leq 10^4$
- $1 \leq y_1 \leq y_2 \leq n$
- $1 \leq x_1 \leq x_2 \leq n$

Problem

Input
The first input line has two integers n and q : the size of the forest and the number of queries.
Then, there are n lines describing the forest. Each line has n characters: $.$ is an empty square and $*$ is a tree.
Finally, there are q lines describing the queries. Each line has four integers y_1, x_1, y_2, x_2 corresponding to the corners of a rectangle.

Output
Print the number of trees inside each rectangle.

Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN)	INPUT (STDIN)

```

#include<bits/stdc++.h>

using namespace std;

#define rep(i,a,b) for (int i=a; i<b; ++i)

int dp[1005][1005];

int main(){

    int n,m; cin>>n>>m;

    rep(i,1,n+1){

        rep(j,1,n+1){

            char x; cin>>x;

            dp[i][j] = (dp[i-1][j] - dp[i-1][j-1]) + dp[i][j-1] + (x=='*');

        }

    }

    while(m--){

        int y1 , x1, y2, x2; cin>>y1>>x1>>y2>>x2;

    }

}

```

```

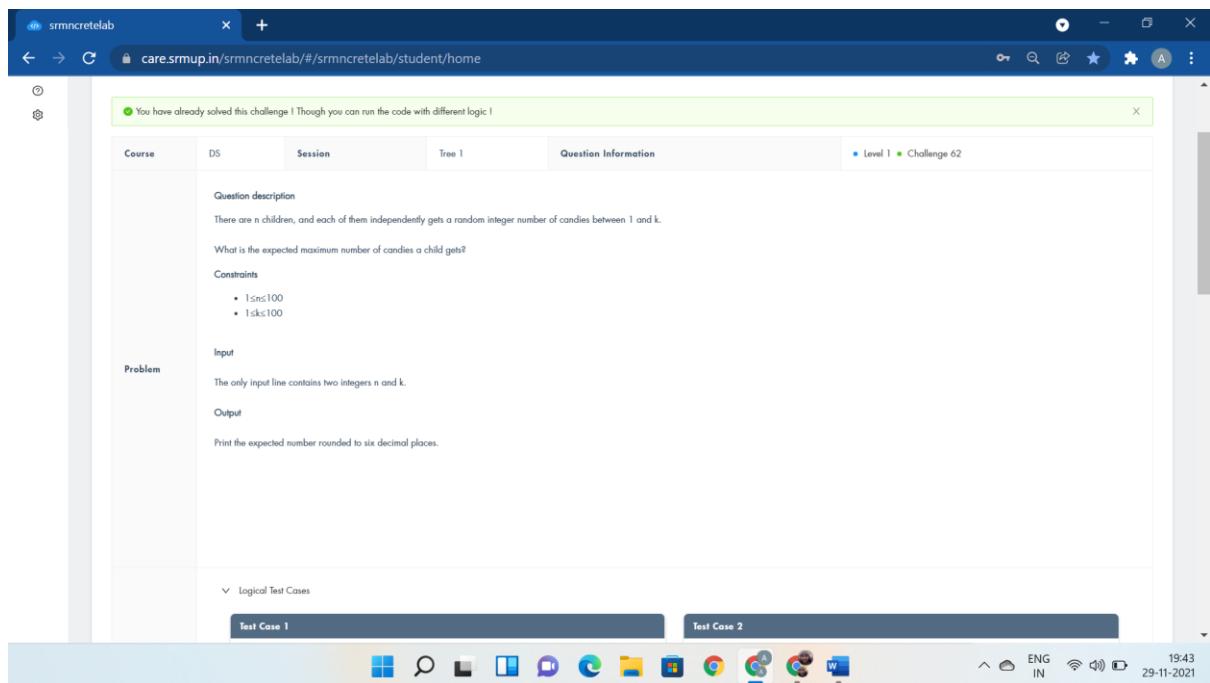
cout<<dp[y2][x2]+ dp[y1-1][x1-1] - dp[y2][x1-1] - dp[y1-1][x2]<<endl;
}

return 0;

cout<<"for(i=1;i<=n;i++)";

}

```



```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int N, K;
```

```
double ans, a, b;
```

```
int main(){
```

```
    scanf("%d %d", &N, &K);
```

```
    for(int i = 1; i <= K; i++){
```

```
        a = b = 1.0;
```

```
        for(int j = 1; j <= N; j++){
```

```
            a *= (double) i / K;
```

```
            b *= (double) (i-1) / K;
```

```
}
```

```
        ans += (a-b) * i;
```

```

}

printf("%.6f\n", ans);

return 0;

cout<<"double power(double a,int k)";

}

```

```

#include<bits/stdc++.h>

using namespace std;

void solve(){}
struct node {
    int data;
    struct node *left,*right;
}*root=NULL;
void insert(int data) {
    struct node *tempNode = (node*) malloc(sizeof(node));
    struct node *current;
    struct node *parent;
    tempNode->data = data;
    tempNode->left = NULL;
    tempNode->right = NULL;
    if(root == NULL) root = tempNode;
    else {

```

```

current = root;
parent = NULL;
while(1) {
    parent = current;
    if(data < parent->data) {
        current = current->left;
        if(current == NULL) {
            parent->left = tempNode;
            return;
        }
    }
    else {
        current = current->right;
        if(current == NULL) {
            parent->right = tempNode;
            return;
        }
    }
}
void preorder(struct node* root) {
    if(root != NULL) {
        printf("%d ",root->data);
        preorder(root->left);
        preorder(root->right);
    }
}
int main() {
    solve();
    int n,i,x; scanf("%d",&n);
    for(i = 0; i < n; i++){
        scanf("%d",&x); insert(x); }
    preorder(root);
    return 0;
}
printf("struct node* newNode(int item) ");

```

You have already solved this challenge! Though you can run the code with different logic!

Course DS **Session** Tree 1 **Question Information** **level 1** **Challenge 64**

Question description

Siva Sir students were chatting and playing quite loudly on the last day of the year, celebrating the end of the academic session. Siva sir was harshly chastised by the college's principal. But, instead of becoming enraged, he attempted to engage everyone in a different task.

So Siva sir gave his students to solve the task such that, you have to perform in-order tree traversal in Binary search tree.

How Inorder works (Manually)

- The direction of traversal for inorder is anti-clockwise
- Rule followed is LCR (Left-Center-Right)

This basically means, that we first try to visit bottommost, the left node then central node and then right and then move our way up to the tree.

Inorder Traversal in Binary Tree

Constraints:
0 < size < 100
0 < data < 1000

Input format:

```
#include <stdio.h>
#include <stdlib.h>

struct node {
    int data;
    struct node *left, *right;
};

void solve(){}
struct node *root = NULL;

void insert(int data) {
    struct node *tempNode = (struct node*) malloc(sizeof(struct node));
    struct node *current;
    struct node *parent;

    tempNode->data = data;
    tempNode->left = NULL;
    tempNode->right = NULL;

    //if tree is empty
    if(root == NULL) {
        root = tempNode;
    } else {
```

```

current = root;
parent = NULL;

while(1) {
    parent = current;

    //go to left of the tree
    if(data < parent->data) {

        current = current->left;

        //insert to the left
        if(current == NULL) {

            parent->left = tempNode;
            return;
        }
    } //go to right of the tree
    else {

        current = current->right;

        //insert to the right
        if(current == NULL) {

            parent->right = tempNode;
            return;
        }
    }
}
}

```

```

void inorder(struct node* root) {
    if(root != NULL) {
        inorder(root->left);
        printf("%d ",root->data);
        inorder(root->right);
    }
}

```

```

}

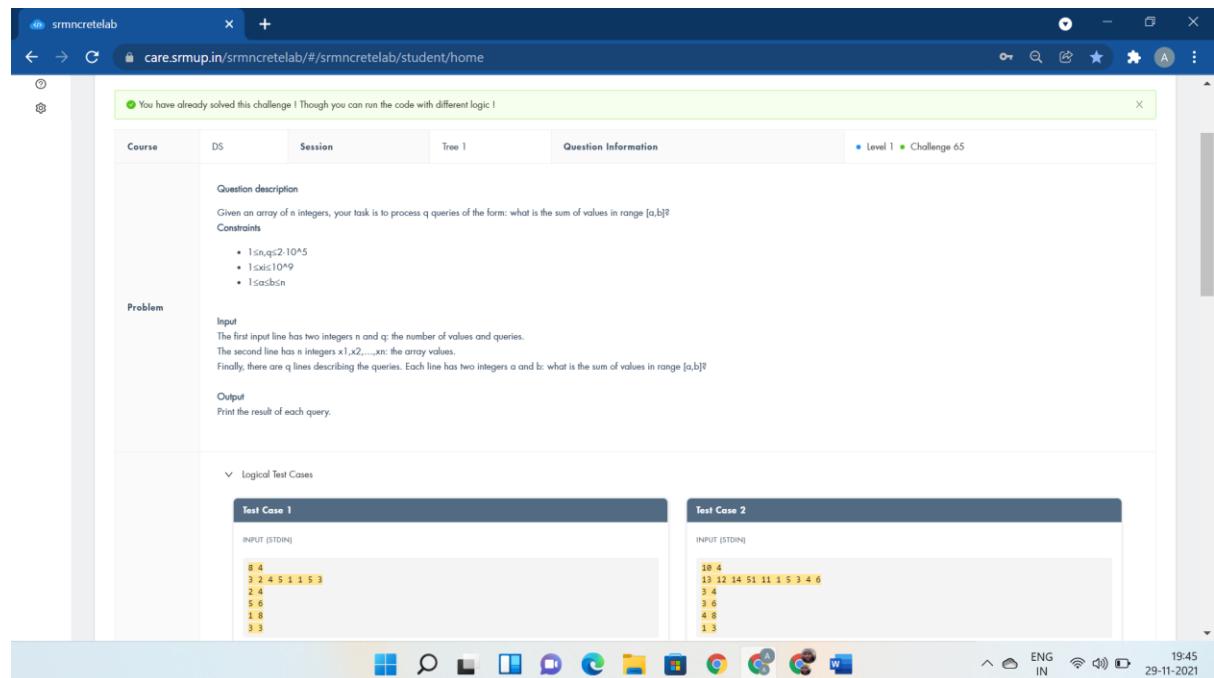
}

int main() {
    solve();
    int n,i;
    scanf("%d",&n);
    int array[n];
    for(i=0;i<n;i++)
        scanf("%d",&array[i]);

    for(i = 0; i < n; i++)
        insert(array[i]);
    inorder(root);
    return 0;
}

printf("temp->left=temp->right=NULL; struct node* newNode(int item)");
return 0;
}

```



```
#include<bits/stdc++.h>
```

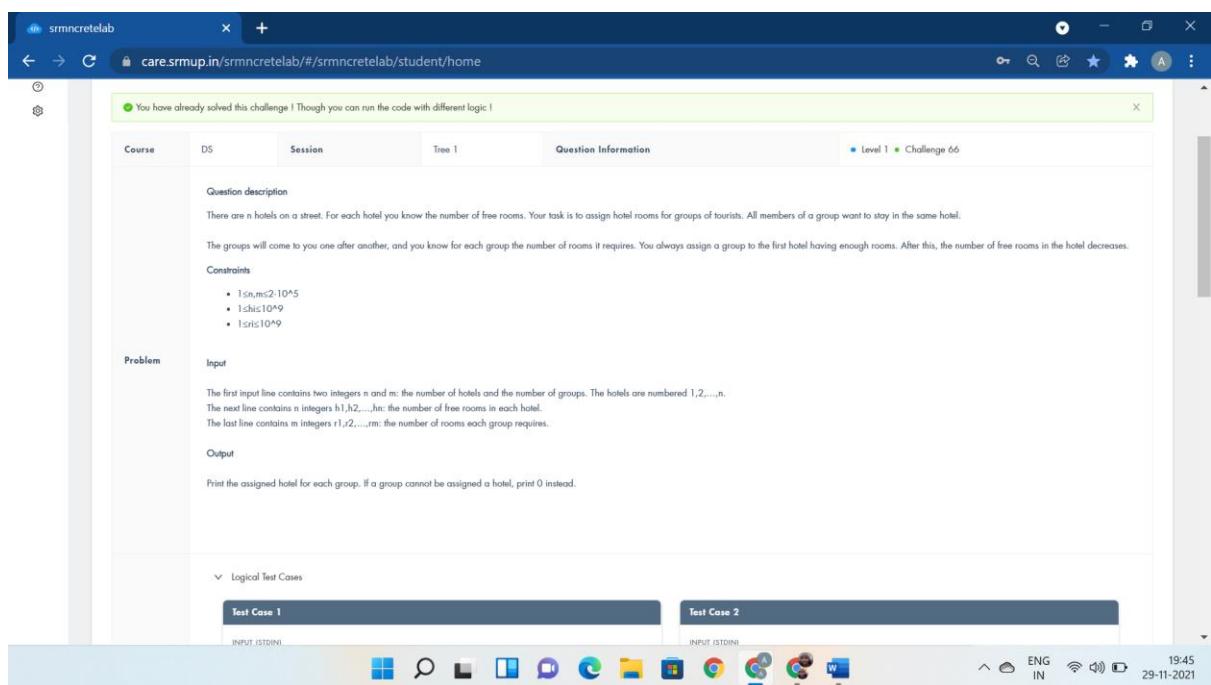
```
using namespace std;
```

```
int main(){
```

```

int n,q,i,a,b;
cin>>n>>q;
int x[n];
for(i=0;i<n;i++)
cin>>x[i];
while(q--){
    int sum=0;
    cin>>a>>b;
    for(i=a;i<=b;i++)
        sum=sum+x[i-1];
    cout<<sum<<endl;
}
}

```



```

#include<iostream>

using namespace std;

void solve(){}
int main()
{
    solve();
    int n,m,i;
}

```

```
cin>>n>>m;

int a[n],b[n];

for(i=0;i<n;i++)

cin>>a[i];

for(i=0;i<n;i++)

cin>>b[i];

for(i=0;i<m;i++){

int f=0,j=0;

for(;j<n;j++){

if(a[j]>=b[i]){

a[j]-=b[i];

f=1;

break;

}

}

if(f>0)

cout<<j+1<<" ";

else

cout<<"0 ";

}

return 0;

}
```

```
#include<bits/stdc++.h>

using namespace std;

#define ll long long
#define MAX 200005
#define pb push_back

vector<int>tree[MAX];
ll up[MAX][20];

void solve(){}
void link(int i,int j){
    up[i][0]=j;
    for(int m=1;m<20;m++){
        if(up[i][m-1]!=-1)
            up[i][m]=up[up[i][m-1]][m-1];
        else
            up[i][m]=-1;
    }
    for(auto child:tree[i]){
        if(child!=j) link(child,i);
    }
}

int ans_query(int src,int jump){
    if(src== -1 or jump==0) return src;
}
```

```

for(int i=19;i>=0;i--){
    if( jump>= (1<<i)){
        return ans_query(up[src][i],jump-(1<<i));
    }
}
return 1;
}

int main(){
    solve();
    int n,q;
    cin>>n>>q;
    for(int i=2;i<=n;i++){
        int ee;
        cin>>ee;
        tree[i].pb(ee);
        tree[ee].pb(i);
    }
    link(1,-1);
    while(q--){
        int node,jump;
        cin>>node>>jump;
        cout<<ans_query(node,jump)<<endl;
    }
}

```

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Tree 1	Question Information	
				Question Information	Level 1 Challenge 68

Question description
A forest is an undirected graph without cycles (not necessarily connected).
Mohana and John are friends in Kerala, both of them have a forest with nodes numbered from 1 to n , and they would like to add edges to their forests such that:

- After adding edges, both of their graphs are still forests.
- They add the same edges. That is, if an edge $\{u, v\}$ is added to Mohana's forest, then an edge $\{u, v\}$ is added to John's forest, and vice versa.

Mohana and John want to know the maximum number of edges they can add, and which edges to add.

Constraints:
 $1 \leq n \leq 105$,
 $0 \leq m_1 \leq n$,
 $m_2 < n$
 $1 \leq u, v \leq n$, $u \neq v$

Input
The first line contains three integers n , m_1 and m_2 — the number of nodes and the number of initial edges in Mohana's forest and John's forest.
Each of the next m_1 lines contains two integers u and v — the edges in Mohana's forest.
Each of the next m_2 lines contains two integers u and v ($1 \leq u, v \leq n$, $u \neq v$) — the edges in John's forest.

Output
The first line contains only one integer k , the maximum number of edges Mohana and John can add.
Each of the next k lines contains two integers u and v ($1 \leq u, v \leq n$, $u \neq v$) — the edge you add each time.
If there are multiple correct answers, you can print any one of them.

```
#include<bits/stdc++.h>

using namespace std;

typedef long long ll;

const int mod=998244353;

int fa[1005],fa2[1005],n,m1,m2;

int gf(int x,int *f){

    return f[x]==x?x:f[x]=gf(f[x],f);

}

int main(){

    cin>>n>>m1>>m2;

    for(int i=1;i<=n;i++)fa[i]=fa2[i]=i;

    for(int i=1,x,y;i<=m1;i++)cin>>x>>y,fa[gf(x,fa)]=gf(y,fa);

    for(int i=1,x,y;i<=m2;i++)cin>>x>>y,fa2[gf(x,fa2)]=gf(y,fa2);

    cout<<n-max(m1,m2)-1<<'\n';

    for(int i=1;i<=n;i++){

        for(int j=i+1;j<=n;j++){

            if(gf(i,fa)!=gf(j,fa)&&gf(i,fa2)!=gf(j,fa2)){

                cout<<i<<' '<<j<<'\n';

                fa[gf(i,fa)]=gf(j,fa);

                fa2[gf(i,fa2)]=gf(j,fa2);

            }

        }

    }

}
```

```

    }

    return 0;

    cout<<"while(m1--)";

}

```

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Tree 1	Question Information
				Level 1 Challenge 69

Question description
You are given a tree consisting of n nodes.
The diameter of a tree is the maximum distance between two nodes. Your task is to determine the diameter of the tree.

Input
The first input line contains an integer n : the number of nodes. The nodes are numbered 1, 2, ..., n .
Then there are $n-1$ lines describing the edges. Each line contains two integers a and b : there is an edge between nodes a and b .

Output
Print one integer: the diameter of the tree.

Constraints

- $1 \leq n \leq 10^4$
- $1 \leq a, b \leq n$

Example

Input:

```

5
1 2
1 3
3 4
3 5

```

Output:

```

#include<bits/stdc++.h>

using namespace std;

#define vi vector<int>
#define rep(i,a,b) for (int i=a; i<b; ++i)
#define pb push_back

vi adj[200005];

int d=0,x=0;

void solve(){}
void dfs(int s, int p, int dep){
    for (auto i: adj[s]){
        if (i!=p){
            dfs(i,s,dep+1);
        }
    }
    if (dep>d) d = dep, x = s;
}

int main(){

```

```

solve();

int n;

cin>>n++;

rep(i,0,n-1){

    int x,y; cin>>x>>y;

    adj[x].pb(y), adj[y].pb(x);

}

dfs(1,0,0);

dfs(x,0,0);

cout<<d;

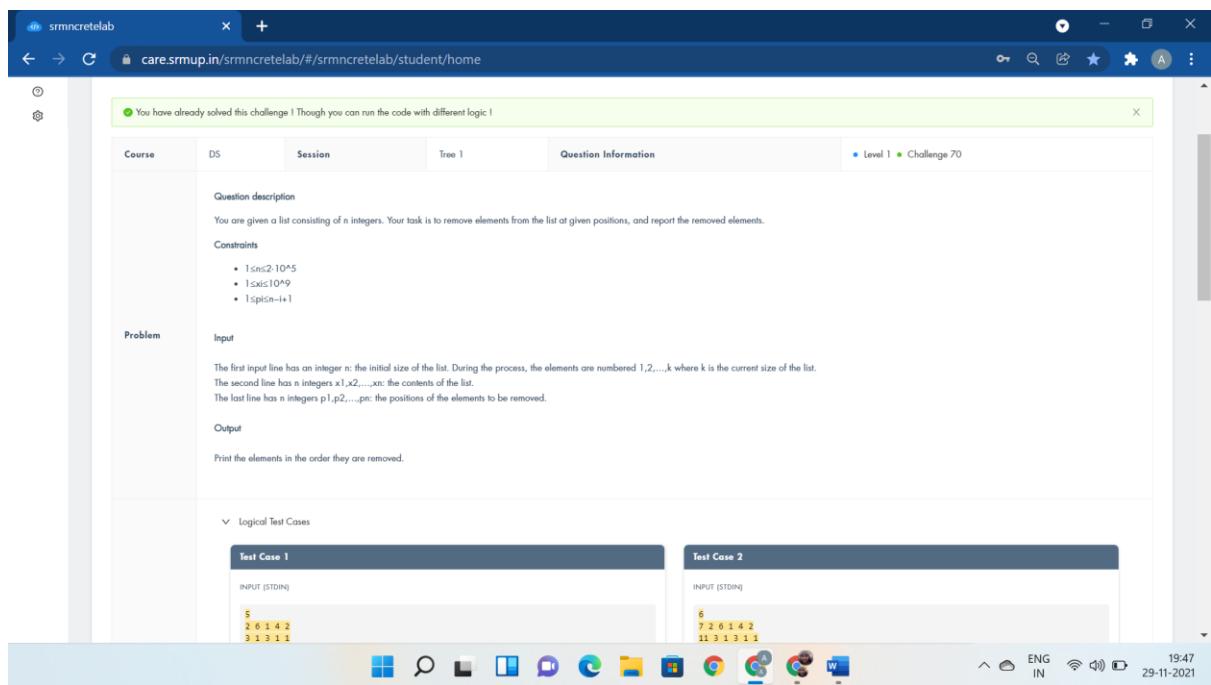
return 0;

cout<<"void link(int i,int j) void dfs(int p,int i,int d)";

}

}

```



```

#include <stdio.h>

#define N 200000

#define N_ (1 << 18)

int tr[N_* 2];

```

```

void build(int k,int l,int r) {

    tr[k] = r - l;

    if (r - l > 1) {

```

```

int m = (l + r) / 2;

build(k * 2 + 1, l, m);
build(k * 2 + 2, m, r);
}

}

int query(int k, int l, int r, int x) {
    int m, k1, k2;

    tr[k]--;
    if (r - l == 1)
        return r;
    m = (l + r) / 2, k1 = k * 2 + 1, k2 = k * 2 + 2;
    return tr[k1] >= x ? query(k1, l, m, x) : query(k2, m, r, x - tr[k1]);
}

int main() {

    int n, h, i, x;

    scanf("%d", &n);
    int aa[n];
    for (i = 0; i < n; i++)
        scanf("%d", &aa[i]);
    build(0, 0, n);
    for (h = 0; h < n; h++) {
        scanf("%d", &x);
        i = query(0, 0, n, x) - 1;
        printf("%d ", aa[i]);
    }
    printf("\n");
    return 0;
}

```

TREE-2:-

The screenshot shows a web browser window with the URL care.srmup.in/srmncretelab/#/srmncretelab/student/home. The page title is "Tree 2". A message at the top says, "You have already solved this challenge! Though you can run the code with different logic!". Below this, there are tabs for "Course", "DS", "Session", "Tree 2", and "Question Information". The "Question Information" tab is selected, showing "Level 1" and "Challenge 71". The main content area is titled "Problem Description" and contains the following text:

A new species is trying to rule the planet. This species is creating their own population outburst to dominate other species. It all started with 1 single member of the species. The population increases in treelike fashion abiding by few rules as listed below.

- Single member is able to reproduce by itself.
- A new member is added to the population every minute.
- Every member is associated with integral name.
- Multiple members can share a common name.
- Every member has it's own reproduction capacity, that is maximum number of children it can reproduce.
- A member can start to reproduce only if all members older than it have exhausted their reproduction capacity.
- Level 0 in family tree of this species comprise of single member at the start of multiplication.
- Integral name of single member at the start is 0.
- The population grows level wise, where number of members at level i is dependent on reproduction capacity of members at prior level.

Given the integral name of new member and it's reproduction capacity that is added to the population, you have to find it's parent, level at which it is added and it's ascending age wise rank among siblings.

Input:
First line of the input contains 2 integers, N, RC_0 , representing number of minutes we will be examining the population increase and reproduction capacity of member at epoch. Next N line contains 2 integers each, ID_i, RC_i , representing integral name and reproduction capacity of new member born at time i .

Output:
N lines, each line containing 3 integers, P, L, C , representing integral name of the parent, level at which it is added and it's ascending age wise rank among siblings.

Note :
It will always be possible to reproduce a new child or in other words, through out the given time, there exists atleast one member which can still accommodate new child.

Constraints:
 $1 \leq N \leq 10^6$
 $-10^9 \leq ID_i \leq 10^9$
 $0 \leq RC_i \leq 10^9$

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

struct cell{
    int name;
    int level;
    int capacity;
};

struct cell queue[1000001];
struct cell arr[1000001];

int front;
int end;

void init_queue(){
    front = 0;
    end = 0;
}

void enqueue(int name,int capacity,int level){
    queue[end].name = name;
    queue[end].level = level;
    queue[end].capacity = capacity;
    end = end + 1;
}
```

```

}

int is_empty(){

    if(end == front)

        return 1;

    return 0;

}

void dequeue()

{

    if(!is_empty())

        front++;

}

int main(){

    int n,rc;

    init_queue();

    scanf("%d %d",&n,&rc);

    int i,j,k;

    for(i=0;i<n;i++){

        scanf("%d %d",&arr[i].name,&arr[i].capacity);

    }

    enqueue(0,rc,0);

    i=0;

    while(!is_empty()){

        int par = queue[front].name;

        int cap = queue[front].capacity;

        int lev = queue[front].level+1;

        k=1;

        for(j=0;j<cap&&i<n;j++,i++){

            printf("%d %d %d\n",par,lev,k++);

            enqueue(arr[i].name,arr[i].capacity,lev);

        }

        dequeue();

    }

    return 0;

}

```

You have already solved this challenge! Though you can run the code with different logic!

Course DS Session 2 Question Information Level 1 • Challenge 72

Problem Description:
Mancunian and Liverbird decide to go camping for the weekend after a long week at work. They came upon an unusual tree with N nodes while walking through a forest. From 1 to N , the vertices are numbered.
A colour is allocated to each node in the tree [out of C possible colors]. They decide to work together [for a change] and put their reasoning abilities to the test because they are bored. At vertex 1, the tree is rooted. They aim to locate the nearest ancestor with the same hue for each node.

Constraints

- $1 \leq N \leq 100,000$
- $1 \leq C \leq 100,000$

Input format
The first line contains two integers N and C denoting the number of vertices in the tree and the number of possible colors.
The second line contains $N - 1$ integers. The i th integer denotes the parent of the $i + 1$ th vertex.
The third line contains N integers, denoting the colors of the vertices. Each color lies between 1 and C inclusive.

Output format
Print N space-separated integers. The i th integer is the vertex number of lowest ancestor of the i th node which has the same color. If there is no such ancestor, print -1 for that node.

Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN) 5 4 1 1 3 3 1 4 2 1 2	INPUT (STDIN) 5 5 1 2 3 3 4 1 3 4 4 2

```
#include<bits/stdc++.h>

using namespace std;

int main() {

    int n,i,c;

    scanf("%d %d", &n, &c);

    int tree[n+1][2];

    tree[1][0] = -1;

    for(i=2;i<=n;i++) {

        scanf("%d", &tree[i][0]);

    }

    for(i = 1; i <= n; i++) {

        scanf("%d", &tree[i][1]);

    }

    int parent;

    for(i = 1; i <= n; i++) {

        parent = tree[i][0];

        while(parent != -1 && tree[parent][1] != tree[i][1]) {

            parent = tree[parent][0];

        }

        printf("%d ", parent);

    }

    return 0;
}
```

You have already solved this challenge ! Though you can run the code with different logic !

Course DS **Session** Tree 2 **Question Information** Level 1 Challenge 73

Question description

A beautiful code of a tree of n nodes is a sequence of $n-2$ integers that uniquely specifies the structure of the tree.

The code is constructed as follows: As long as there are at least three nodes left, find a leaf with the smallest label, add the label of its only neighbour to the code, and remove the leaf from the tree.

Given a beautiful code of a tree, your task is to construct the original tree.

Constraints

- $3 \leq n \leq 10^5$
- $1 \leq a, b \leq n$

Problem

Input

The first input line contains an integer n : the number of nodes. The nodes are numbered $1, 2, \dots, n$.

The second line contains $n-2$ integers: the beautiful code.

Output

Print $n-1$ lines describing the edges of the tree. Each line has to contain two integers a and b : there is an edge between nodes a and b . You can print the edges in any order.

Logical Test Cases

Test Case 1 Test Case 2

```
#include <bits/stdc++.h>

using namespace std;

#define f(i,a,n) for(int i=a;i<n;i++)
#define X(a,b) if(a==b)

vector< int > vi;

const int maxN = 2e5+5;

int N, a[maxN], deg[maxN];

priority_queue<int, vector<int>, greater<int>> q;

int main(){

    scanf("%d", &N);
    fill(deg+1, deg+N+1, 1);

    //for(int i = 0; i < N-2; i++)
    f(0,N-2){

        scanf("%d", &a[i]);
        deg[a[i]]++;

    }

    //for(int i = 1; i <= N; i++)
    f(1,N+1)

    //if(deg[i] == 1)
    X(deg[i],1)
```

```

q.push(i);

f(i,0,N-2){
    int u = a[i];
    int v = q.top(); q.pop();

    deg[u]--; deg[v]--;
    //if(deg[u] == 1)
    X(deg[u],1)
    q.push(u);

    printf("%d %d\n", v, u);
}

//for(int i = 1; i <= N; i++)
f(i,1,N+1)
if(deg[i])
    printf("%d ", i);

}

```

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Tree 2	Question Information
				Level 1 Challenge 74

Problem Description:
You're given a K -ary infinite tree rooted at a vertex numbered 1. All its edges are weighted 1 initially.
Any node X will have exactly K children numbered as:
 $[K*X+0, K*X+1, K*X+2, K*X+3, K*X+4, \dots, K*X+(K-1)]$
You are given Q queries to answer which will be of the following two types:
1. $1\ UV$: Print the shortest distance between nodes U and V .
2. $2\ UWV$: Increase the weight of all edges on the shortest path between U and V by W .

Constraints
 $2 \leq K \leq 10$
 $1 \leq Q \leq 10^3$
 $1 \leq U, V \leq 10^{18}$
 $U \neq V$
 $1 \leq W \leq 10^9$

Input format

- The first line contains two space-separated integers K and Q .
- Next Q lines contain queries which will be of 2 types:
 - Three space-separated integers 1, U , and V
 - Four space-separated integers 2, U , V , and W

Output format
For each query of type (1UV), print a single integer denoting the shortest distance between U and V .

```

#include <iostream>
#include <map>
#include <assert.h>

```

```

using namespace std;

#define int long long

map<pair<int, int>, int> adj;

int find_depth( int u, int k ) {

    int depth = 0;

    while ( u > 0 ) {

        u = u / k;

        depth = depth + 1;

    }

    return depth - 1;

}

int dist( int u, int v, int k ) {

    int dist = 0;

    int depth_u = find_depth( u, k );

    int depth_v = find_depth( v, k );

    if ( depth_u < depth_v ) {

        swap ( u, v );

        swap ( depth_u, depth_v );

    }

    while( depth_u != depth_v ) {

        if ( adj.count( { u, u / k } ) ) {

            dist = dist + adj[ { u, u / k } ];

        } else {

            dist = dist + 1;

        }

        depth_u = depth_u - 1;

        u = u / k;

    }

    while ( u != v ) {

        if ( adj.count( { u, u / k } ) ) {

            dist = dist + adj[ { u, u / k } ];

        } else {

            dist = dist + 1;

        }

        if ( adj.count( { v, v / k } ) ) {


```

```

    dist = dist + adj[ { v, v / k } ];
} else {
    dist = dist + 1;
}
u = u / k;
v = v / k;
}
return dist;
}

void add_weight( int vertex, int parent, int w ) {
if ( !adj.count( { vertex, parent } ) ) {
    adj[ { vertex, parent } ] = 1;
}
adj[ { vertex, parent } ] = adj[ { vertex, parent } ] + w;
}

void increase_weights ( int u, int v, int w, int k ) {
int depth_u = find_depth( u, k );
int depth_v = find_depth( v, k );
if ( depth_u < depth_v ) {
    swap ( u, v );
    swap ( depth_u, depth_v );
}
while( depth_u != depth_v ) {
    add_weight( u, u / k, w );
    depth_u = depth_u - 1;
    u = u / k;
}
while ( u != v ) {
    add_weight( u, u / k, w );
    add_weight( v, v / k, w );
    u = u / k;
    v = v / k;
}
}

signed main() {

```

```

int k, q, x, u, v, w;
cin >> k >> q;

while(q--) {
    cin >> x;
    if ( x == 1 ) {
        cin >> u >> v;
        cout << dist( u, v, k ) << "\n";
    } else {
        cin >> u >> v >> w;
        increase_weights( u, v, w, k );
    }
}

```

You have already solved this challenge! Though you can run the code with different logic!

Course DS **Session** Tree 2 **Question Information** Level 1 Challenge 75

Problem Description

Football is Monk's favourite sport, and his favourite team is "Manchester United." Manchester United has qualified for the Champions League Final, which will take place at London's Wembley Stadium. As a result, he decided to go watch his favourite team play.

When he arrived at the stadium, he noticed that there was a long wait for match tickets. He is aware that the stadium has M rows, each with a distinct seating capacity. They could or might not be comparable. The cost of a ticket is determined by the row. If there are K[always higher than Q] empty seats in a row, the ticket will cost K pounds (units of British Currency).

Now, every football fan standing in the line will get a ticket one by one. Given the seating capacities of different rows, find the maximum possible pounds that the club will gain with the help of the ticket sales.

Constraints:

- 1 <= M <= 1000000
- 1 <= N <= 1000000
- 1 <= X[i] <= 1000000
- Sum of X[i] for all 1 <= i <= M will always be greater than N.

Input:

The first line consists of M and N. M denotes the number of seating rows in the stadium and N denotes the number of football fans waiting in the line to get a ticket for the match. Next line consists of M space separated integers X[1],X[2],X[3]... X[M] where X[i] denotes the number of empty seats initially in the i^{th} row.

Output:

Print in a single line the maximum pounds the club will gain.

Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN) 3 4 3 4	INPUT (STDIN) 8 4 8 4

ENG IN 20:03 29-11-2021

```

#include <bits/stdc++.h>

using namespace std;

#define PII pair<int, int>

priority_queue<int> seats;

map<int, int> x;

int main()
{

```

```

int N, M; cin >> N >> M;

assert (1<=N and N<=1000000);
assert (1<=M and M<=1000000);

for (int g=1; g<=N; g++){

    int a; cin >> a;

    seats.push(a);

    assert (1<=a and a<=1000000);

    x[a]++;
}

long long ans = 0;

for (int g=0; g<M; g++){

    int x = seats.top(); ans+=x; seats.pop();seats.push(x-1);

}

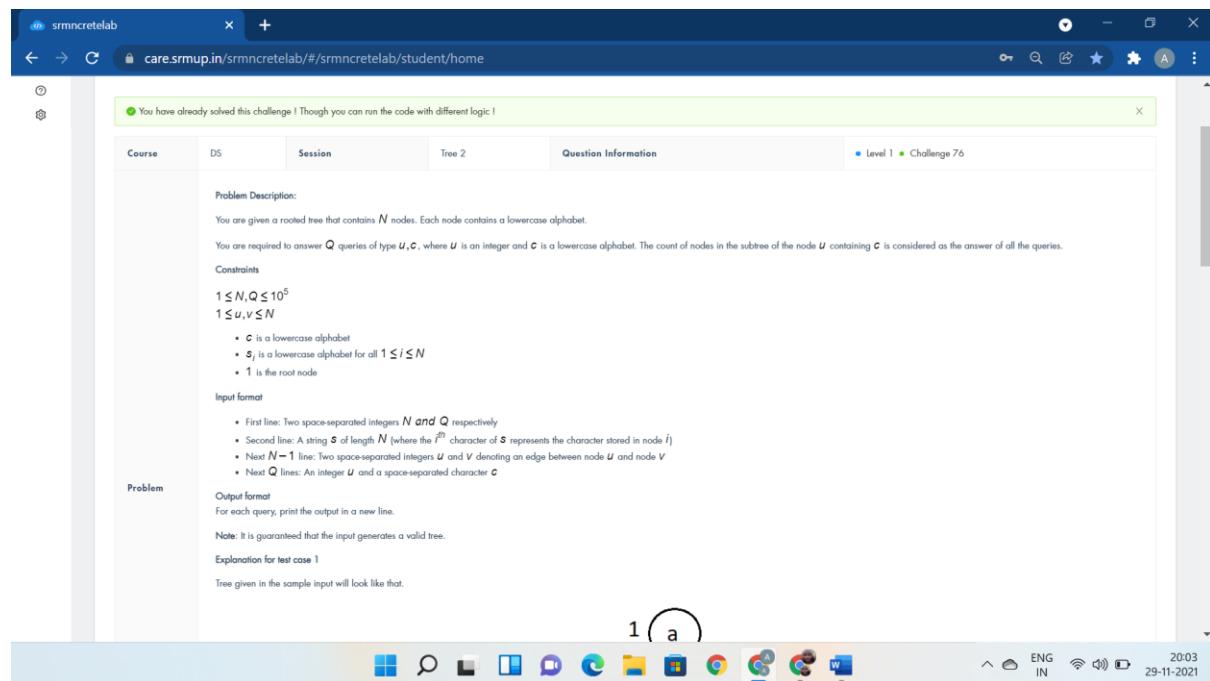
cout <<ans;

return 0;

cout<<"void heapify(int arr[],int n,int i)";

}

```



```

#include<bits/stdc++.h>

using namespace std;

void dfs(int node,int parent,string &s,vector<vector<int>>&subroot,vector<vector<int>>& v1)
{

```

```

//visited[node]=1;
subroot[node][s[node-1]-'a']++;
//intime[node]=t;
//t++;
//z.push_back(node);
for( auto it:v1[node])
{
    if(it!=parent)
    {
        dfs(it,node,s,subroot,v1);
        for(int i=0;i<26;i++)
            subroot[node][i]+=subroot[it][i];
    }
}

//outtime[node]=t;
//t++;

}

int main()
{
    int N,i, Q;
    string S;
    cin >> N >> Q;
    cin >> S;
    vector<vector<int>>v1(N+1);
    for(i=0;i<N-1;i++)
    {
        int u, v;
        cin >> u >> v;
        v1[u].push_back(v);
        v1[v].push_back(u);
    }

    vector<vector<int>>subroot(N+1,vector<int>(26,0));
    dfs(1,0,S,subroot,v1);
    while(Q--)

```

```

    {
        int u;
        char c;
        cin >> u >> c;
        cout<<subroot[u][c-'a']<<"\n";
        //cout<<cnt<<endl;
    }

}

return 0;
}

```

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Tree 2	Question Information
Problem				Level 1 • Challenge 77

Problem Description
You are given a weighted graph with N vertices and M edges. Find the total weight of its maximum spanning tree.

Constraints

- $1 \leq T \leq 20$
- $1 \leq N \leq 5000$
- $1 \leq M \leq 100\,000$
- $1 \leq c \leq 10\,000$

Input
The first line contains one integer T denoting the number of test cases. Each test case starts with a line containing 2 space-separated integers: N and M. Each of the following M lines contain description of one edge: three different space-separated integers: a, b and c. a and b are different and from 1 to N each and denote numbers of vertices that are connected by this edge, c denotes weight of this edge.

Output
For each test case output one integer - the total weight of its maximum spanning tree.

Logical Test Cases

Test Case 1	Test Case 2
INPUT [STDIN] 1 3 3 1 2 2 2 3 3 1 3 4	INPUT [STDIN] 1 3 3 1 2 3 2 3 4 1 3 5

```

#include<bits/stdc++.h>

typedef long long ll;

using namespace std;

struct edge
{
    int u;
    int v;
    int w;
};

edge a[100005];

int parent[100005];

```

```

bool comp (edge a , edge b)
{
    return a.w>b.w;
}

int find_parent(int u) ///DSU find
{
    /* return (parent[u]==u) ? u: find_parent(parent[u]);*/
    if(parent[u]==u)
        return u;
    else
        return parent[u]=find_parent(parent[u]);
}

void merge(int u, int v) /// DSU union
{
    parent[u]=v;
}

int main()
{
    int t;
    cin>>t;
    while(t--) {
        int n,m;
        cin>>n>>m;
        for(int i=1;i<=n;i++)
            parent[i]=i;
        for(int i=0;i<m;i++) {
            cin>>a[i].u>> a[i].v >> a[i].w;
        }
        sort(a,a+m,comp);
        ll ans=0;
        for(int i=0;i<m;i++) {
            int x=find_parent(a[i].u);
            int y=find_parent(a[i].v);
            if(x!=y)
            {
                merge(x,y);
                ans+=a[i].w;
            }
        }
    }
}

```

```

    }

    cout<<ans<<endl;
}

return 0;

cout<<"int printheap(int N)";

}

```

You have already solved this challenge! Though you can run the code with different logic!

Course: DS **Session:** Tree 2 **Question Information:** Level 1 | Challenge 78

Problem Description:

Any sequence A of size n is called B-sequence if:
 $A_1 < A_2 < \dots < A_k > A_{k+1} > A_{k+2} > \dots > A_n$ where $1 \leq k \leq n$. That is, a sequence which is initially strictly increasing and then strictly decreasing (the decreasing part may or may not be there).

All elements in A except the maximum element comes almost twice (once in increasing part and once in decreasing part) and maximum element comes exactly once.

All elements coming in decreasing part of sequence should have come once in the increasing part of sequence.

You are given a B-sequence S and Q operations. For each operation, you are given a value val. You have to insert val in S if and only if after insertion, S still remains a B-sequence. After each operation, print the size of S. After all the operations, print the sequence S.

Hint: Think of using some data structure to support insertion of elements in complexity better than linear.

Input Constraints:

- $1 \leq N \leq 10^5$
- $1 \leq S_i \leq 10^9$
- $1 \leq Q \leq 10^5$
- $1 \leq val \leq 10^9$

Given sequence S is a B-sequence.

Input Format:

First line consists of an integer N, denoting size of S.
Second line consists of N space separated integers, denoting elements of S.
Next line consists of an integer Q, denoting number of operations.
Each of the following Q lines consists of an integer val.

Output Format:

After each operation, print the size of S in a new line.
After all operations, print the sequence S.

```
#include<bits/stdc++.h>
```

```
#include<map>
```

```
using namespace std;
```

```
int main() {
```

```
    int N,i,maximum=INT_MIN;
```

```
    scanf("%d", &N);
```

```
    int S[N];
```

```
    map<int,int> map;
```

```
for(i=0;i<N;i++) {  
  
    scanf("%d", &S[i]);  
  
    maximum=max(maximum,S[i]);  
  
    map[S[i]]++;  
  
}  
  
int temp,Q;  
  
cin>>Q;  
  
for(i=0;i<Q;i++) {  
  
    scanf("%d", &temp);  
  
    if(temp==maximum) printf("%d\n",N);  
  
    else {  
  
        if(map[temp]>=2) printf("%d\n",N);  
  
        else {  
  
            map[temp]++;  
  
            N++;  
  
            printf("%d\n",N);  
  
            maximum=max(maximum,temp);  
  
        }  
  
    }  
}
```

```
}
```

```
for(auto it=map.begin();it!=map.end();it++) printf("%d ",it->first);
```

```
for(auto it=map.rbegin();it!=map.rend();it++) {
```

```
    if(it->second>1) printf("%d ",it->first);
```

```
}
```

```
}
```

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
int dp[1000006][25];
```

```
void solve(){}
```

```
int main(){
```

```
    solve();
```

```
    int n, q; cin>>n>>q;
```

```
    for (int i = 0; i < n; i++) {
```

```
        int x, y; cin>>x>>y;
```

```

dp[y][0] = max(dp[y][0], x);
}

for (int i = 1; i <= 1000000; i++)
    dp[i][0] = max(dp[i][0], dp[i-1][0]);

for (int k = 1; k <= 20; k++)
    for (int i = 1; i <= 1000000; i++)
        dp[i][k] = dp[dp[i][k-1]][k-1];

while(q--) {
    int x,y; cin>>x>>y;
    int ans = 0;
    while(y>0) {
        int z = 0;
        for (int i = 0; i <= 20; i++) {
            if (dp[y][i] < x) {
                z = i;
                break;
            }
        }
        if (z == 0)
            break;
        ans += (1<<(z-1));
        y = dp[y][z-1];
    }
    cout<<ans<<endl;
}
}

```

You have already solved this challenge! Though you can run the code with different logic!

Course DS **Session** Tree 2 **Question Information** Level 1 Challenge 80

Problem

Problem Description

M is alone and he has an array a_1, a_2, \dots, a_n . M wants to choose two integers i, j such that $i \neq j, 1 \leq i, j \leq n$ and the value $a_i \& a_j$ [bitwise AND] is maximum.

What is the maximum value M can get?

Input

First line contains only n , length of array.

Second line contains the array elements a_1, a_2, \dots, a_n separated by space.

$1 \leq n \leq 3 \times 10^5$

$1 \leq a_i \leq 10^9$

Output

The only line of output contains an integer, maximum value that M can get.

Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN) 4 3 4 2 3	INPUT (STDIN) 5 1 2 3 2 5
EXPECTED OUTPUT 3	EXPECTED OUTPUT 2

```
# include<stdio.h>
# include<stdlib.h>
# include<math.h>
void input(long *,int);
int main()
{
    int n;

    scanf("%d",&n);
    long *ptr = (long*)malloc(n*sizeof(long));
    input(ptr,n);

    return 0;
}
```

```
void input(long *ptr, int n)
{
    int i, j;
    int m;
    for(i=0;i<n;i++)
    {
        scanf("%ld", ptr+i);
    }
```

```

for(i = 0; i < n; i++)
{
    if (*(ptr + i) <= m)
    {
        continue;
    }

    for (j = i + 1; j < n; j++)
    {
        int temp = *(ptr + i) & *(ptr + j);
        if(temp > m)
        {
            m = temp;
        }
    }
}

printf("%d", m);
}

```

GRAPH:-

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Graph	Question Information
				Level 1 Challenge 81

Problem

Consider a network consisting of n computers and m connections. Each connection specifies how fast a computer can send data to another computer.

Kotivalo wants to download some data from a server. What is the maximum speed he can do this, using the connections in the network?

Input

The first input line has two integers n and m : the number of computers and connections. The computers are numbered $1, 2, \dots, n$. Computer 1 is the server and computer n is Kotivalo's computer.

After this, there are m lines describing the connections. Each line has three integers a , b and c : computer a can send data to computer b at speed c .

Output

Print one integer: the maximum speed Kotivalo can download data.

Constraints

- $1 \leq n \leq 500$
- $1 \leq m \leq 1000$
- $1 \leq a, b, c \leq n$
- $1 \leq c \leq 10^9$

Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN) 5 5 1 2 1 1 3 2 2 4 3 3 5 4 4 5 5	INPUT (STDIN) 5 5 1 2 1 1 3 2 2 4 3 3 5 4 4 5 5

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```

using ll = long long;

#define FOR(i,a) for(int i=0; i<(a); i++)
#define FOR(i,a,b) for(int i=(a); i<=(b); i++)

int n, m;

ll adj[501][501], oadj[501][501];

ll flow[501];

bool V[501];

int pa[501];

void link(int i,int h){}

int bfs(int n,int s,int t){return 1;}

bool reachable() {

    memset(V, false, sizeof(V));

    queue<int> Q; Q.push(1); V[1]=1;

    while(!Q.empty()) {

        int i=Q.front(); Q.pop();

        FOR(j,1,n) if (adj[i][j] && !V[j])

            V[j]=1, pa[j]=i, Q.push(j);

    }

    return V[n];
}

int main() {

    bfs(1,1,1);

    link(1,1);

    cin >> n >> m;

    FOR(i,1,n) FOR(j,1,n) adj[i][j] = 0;

    FOR(i,m) {

        ll a,b,c; cin >> a >> b >> c;

        adj[a][b] += c;
    }

    int v, u;

    ll maxflow = 0;

    while(reachable()) {

        ll flow = 1e18;

        for (v=n; v!=1; v=pa[v]) {

            u = pa[v];

            flow = min(flow, adj[u][v]);
        }
    }
}

```

```

maxflow += flow;

for (v=n; v!=1; v=pa[v]) {
    u = pa[v];
    adj[u][v] -= flow;
    adj[v][u] += flow;
}

cout << maxflow << '\n';
}

```

You have already solved this challenge! Though you can run the code with different logic!

Course DS **Session** Graph **Question Information** Level 1 Challenge 82

Question description

A game consists of n rooms and m teleporters. At the beginning of each day, you start in room 1 and you have to reach room n . You can use each teleporter at most once during the game. How many days can you play if you choose your routes optimally?

Constraints

$2 \leq n \leq 500$
 $1 \leq m \leq 1000$
 $1 \leq a, b \leq n$

Input

The first input line has two integers n and m : the number of rooms and teleporters. The rooms are numbered $1, 2, \dots, n$. After this, there are m lines describing the teleporters. Each line has two integers a and b : there is a teleporter from room a to room b . There are no two teleporters whose starting and ending room are the same.

Output

First print an integer k : the maximum number of days you can play the game. Then, print k route descriptions according to the example. You can print any valid solution.

Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN) 6 7 1 2 1 3 2 6 3 4 3 5	INPUT (STDIN) 6 7 1 4 3 5 4 6 5 6 2 2

```
#include <stdio.h>
```

```
#define N 500
```

```
#define M 1000
```

```
struct L {
```

```
    struct L *next;
```

```
    int h;
```

```
} aa[N * 2];
```

```
int ij[M + N], cc[(M + N) * 2], dd[N * 2];
```

```
int bfs(int n,int s,int t) {
```

```

static int qq[N * 2];

int head, cnt, h, i, j, d;

for (i = 0; i < n; i++)
    dd[i] = n;
dd[s] = 0;
head = cnt = 0;
qq[head + cnt++] = s;
while (cnt) {
    struct L *l;

    i = qq[cnt--, head++];
    d = dd[i] + 1;
    for (l = aa[i].next; l; l = l->next)
        if (cc[h = l->h]) {
            j = i ^ ij[h >> 1];
            if (dd[j] == n) {
                dd[j] = d;
                if (j == t)
                    return 1;
                qq[head + cnt++] = j;
            }
        }
    }
    return 0;
}

int dfs(int n, int i, int t) {
    struct L *l;
    int h, j, d;

    if (i == t)
        return 1;
    d = dd[i] + 1;
    for (l = aa[i].next; l; l = l->next)
        if (cc[h = l->h]) {
            j = i ^ ij[h >> 1];
            if (dd[j] == d && dfs(n, j, t)) {

```

```
    cc[h]--, cc[h ^ 1]++;
    return 1;
}
}

dd[i] = n;
return 0;
}
```

```
int dinic(int n, int s, int t) {
```

```
    int f = 0;
```

```
    while (bfs(n, s, t))
```

```
        while (dfs(n, s, t))
```

```
            f++;
```

```
    return f;
}
```

```
void link(int i, int j, int h, int c) {
```

```
    static struct L l91[(M + N) * 2], *l = l91;
```

```
    ij[h] = i ^ j;
```

```
    cc[h << 1] = c;
```

```
    l->h = h << 1;
```

```
    l->next = aa[i].next, aa[i].next = l++;
```

```
    l->h = h << 1 ^ 1;
```

```
    l->next = aa[j].next, aa[j].next = l++;
```

```
}
```

```
int qq[N];
```

```
int path(int i, int t) {
```

```
    int cnt = 0;
```

```
    while (i != t) {
```

```
        struct L *l;
```

```
        int h;
```

```
        qq[cnt++] = i;
```

```

for (l = aa[i].next; l; l = l->next)
    if (((h = l->h) & 1) == 0 && cc[h ^ 1]) {
        cc[h]++;
        cc[h ^ 1]--;
        i ^= ij[h >> 1];
        break;
    }
}

qq[cnt++] = t;
return cnt;
}

```

```

int main() {
    int n, m, h, i, j, k, s, t, cnt;

    scanf("%d%d", &n, &m);
    for (h = 0; h < m; h++) {
        scanf("%d%d", &i, &j), i--, j--;
        link(i << 1 ^ 1, j << 1, h, 1);
    }

    for (i = 0; i < n; i++)
        link(i << 1, i << 1 ^ 1, m + i, n);

    s = 0, t = (n - 1) << 1 ^ 1;
    k = dinic(n * 2, s, t);
    printf("%d\n", k);

    while (k--) {
        cnt = path(s, t);
        printf("%d\n", cnt / 2);
        for (i = 0; i < cnt; i += 2)
            printf("%d ", (qq[i] >> 1) + 1);
        printf("\n");
    }

    return 0;
}

```

```
#include <stdio.h>
```

```
#if defined(_WIN32)

typedef __int64 az_int64_t;

typedef unsigned __int64 az_uint64_t;

#define I64(x) x##I64

#define F64 "I64"

#else

typedef long long az_int64_t;

typedef unsigned long long az_uint64_t;

#define I64(x) x##ll

#define F64 "ll"

#endif
```

```
#define MAXN (100*1024)
```

```
struct link
```

```
{
    az_int64_t t;
    int u, v;
};
```

```
struct link links[MAXN];
```

```

int n, m, k;
az_int64_t c[64];

int gr[MAXN];

int getgr( int g )
{
    return (g == gr[g]) ? g : (gr[g] = getgr( gr[g] ));
}

int test( az_int64_t r )
{
    int i, left = n-1, u, v;
    for(i=1;i<=n;++i) gr[i] = i;
    for( i = 0; i < m; ++i)
        if( (links[i].t & r) == 0 &&
            (u = getgr( links[i].u )) != (v = getgr( links[i].v )) )
    {
        gr[v] = u;
        if( --left == 0 ) return 1;
    }
    return 0;
}

int main( void )
{
    az_int64_t rejected = 0, sum = 0;
    int i;

    scanf( "%d %d %d", &n, &m, &k);
    for( i = 0; i < k; ++i) scanf( "%" F64 "d", &c[i]);
    for( i = 0; i < m; ++i)
    {
        int l, id;
        scanf( "%d %d %d", &links[i].u, &links[i].v, &l);
        while( l-- > 0 )
        {
            scanf( "%d", &id);

```

```

links[i].t |= I64(1) << (id-1);

}

}

if( !test( 0 ) )
{
    printf( "-1\n" );
    return 0;
}

for( i = k-1; i >= 0; --i )
{
    az_int64_t f = I64(1) << i;
    if( test( rejected | f ) ) rejected |= f; else sum += c[i];
}

printf( "%" F64 "d\n", sum);
return 0;
}

```

The screenshot shows a web browser window with the URL care.srmup.in/srmncretelab/#/srmncretelab/student/home. The page indicates that the challenge has been solved. It contains the following sections:

- Question Information:** Level 1, Challenge 84.
- Problem Description:** You are playing a game consisting of n planets. Each planet has a teleporter to another planet (or the planet itself). You start on a planet and then travel through teleporters until you reach a planet that you have already visited before. Your task is to calculate for each planet the number of teleportations there would be if you started on that planet.
- Input:** The first input line has an integer n : the number of planets. The planets are numbered 1, 2, ..., n .
- Output:** Print n integers according to the problem statement.
- Constraints:**
 - $1 \leq n \leq 2 \cdot 10^5$
 - $1 \leq f_i \leq n$
- Test Cases:**
 - Test Case 1:** Input: 5, Output: 5 2 4 3 1 4
 - Test Case 2:** Input: 6, Output: 6 2 1 3 5 6 4

```

#include <stdio.h>

#include <string.h>

```

```

#define N 200000

int main() {
    static int aa[N], cc[N], dd[N], qq[N];
    int n, i, j, c, d, q, cnt;

    scanf("%d", &n);
    for (i = 0; i < n; i++)
        scanf("%d", &aa[i]), aa[i]--;
    memset(cc, -1, n * sizeof *cc);
    cnt = 0;
    for(i = 0;i<n;i++) {
        if (cc[i] != -1)
            continue;
        d = 0;
        j = i;
        while (cc[j] == -1) {
            cc[j] = -2;
            d++;
            j = aa[j];
        }
        if (cc[j] == -2) {
            c = cnt++;
            q = 0;
            while (cc[j] == -2) {
                cc[j] = c;
                q++;
                j = aa[j];
            }
            qq[c] = q;
            d -= q;
        } else {
            c = cc[j];
            d += dd[j];
        }
        j = i;
        while (cc[j] == -2) {

```

```

cc[j] = c;
dd[j] = d--;
j = aa[j];
}

}

for (i = 0; i < n; i++)
printf("%d ", dd[i] + qq[cc[i]]);

printf("\n");

return 0;
}

```

You have already solved this challenge ! Though you can run the code with different logic !

Question Information

Level 1 • Challenge 8.5

Question description

A game has n levels and m teleporters between them. You win the game if you move from level 1 to level n using every teleporter exactly once.

Can you win the game, and what is a possible way to do it?

Input

The first input line has two integers n and m : the number of levels and teleporters. The levels are numbered $1, 2, \dots, n$.

Then, there are m lines describing the teleporters. Each line has two integers a and b : there is a teleporter from level a to level b .

Problem

You can assume that each pair (a, b) in the input is distinct.

Output

Print $m+1$ integers: the sequence in which you visit the levels during the game. You can print any valid solution. If there are no solutions, print "IMPOSSIBLE".

Constraints

- $2 \leq n \leq 10^5$
- $1 \leq m \leq 10^5$
- $1 \leq a, b \leq n$

Logical Test Cases

Test Case 1 Test Case 2

```

#include <stdio.h>

#define N 100000
#define M 200000

struct L {
    struct L *next;
    int j;
} *aa[N];

struct L *new_L(int j) {
    static struct L l91[M + 1 + M], *l = l91;
    l->j = j;
    l->next = l91;
    return l;
}

```

```

l->j = j;
return l++;
}

void link(int i,int j) {
    struct L *l = new_L(j);

    l->next = aa[i]; aa[i] = l;
}

void hierholzer(struct L *e) {
    struct L *f = e->next, *l;
    int i = e->j;

    while ((l == aa[i])) {
        aa[i] = l->next;
        e = e->next = new_L(l->j);
        i = l->j;
    }
    e->next = f;
}

int main() {
    static int din[N], dout[N];
    struct L *e_, *e;
    int n, m, h, i, j;

    scanf("%d%d", &n, &m);
    for (h = 0; h < m; h++) {
        scanf("%d%d", &i, &j), i--, j--;
        link(i, j);
        dout[i]++, din[j]++;
    }
    if (dout[0] - din[0] != 1 || din[n - 1] - dout[n - 1] != 1) {
        printf("IMPOSSIBLE\n");
        return 0;
    }
}

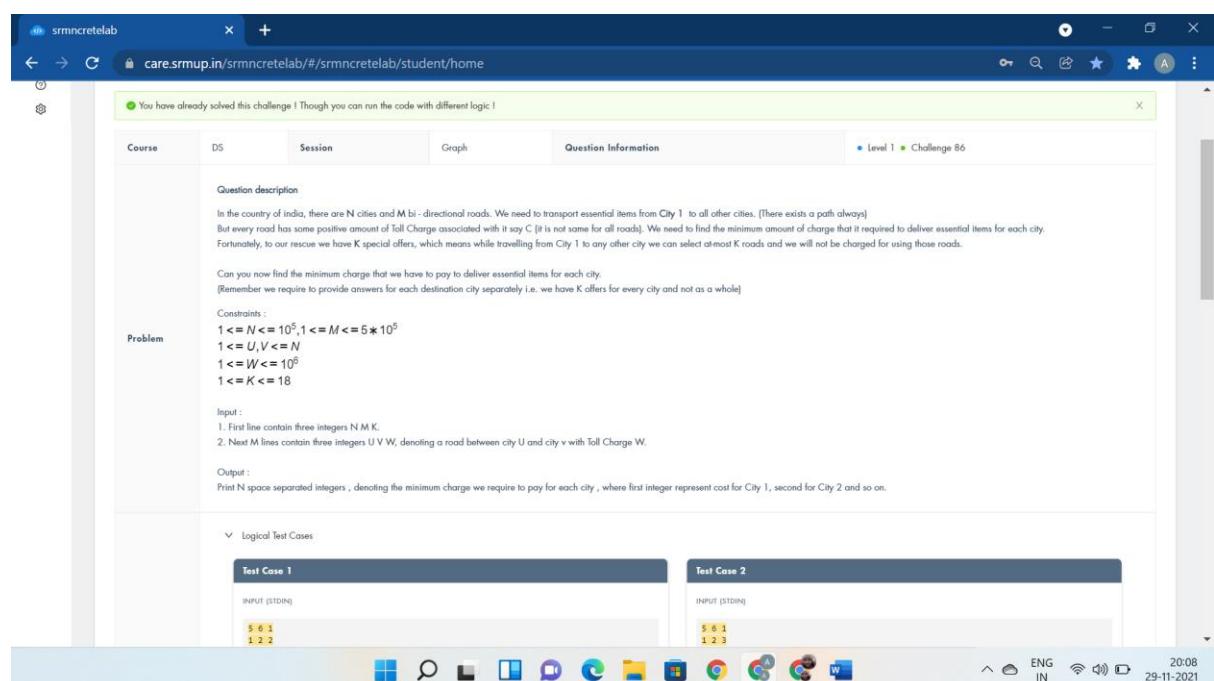
```

```

for (i = 1; i < n - 1; i++) {
    if (dout[i] != din[i]) {
        printf("IMPOSSIBLE\n");
        return 0;
    }
}

e_ = new_L(0);
m++;
hierholzer(e_);
for (e = e_; e; e = e->next) {
    hierholzer(e);
    m--;
}
if (m != 0) {
    printf("IMPOSSIBLE\n");
    return 0;
}
for (e = e_; e; e = e->next)
    printf("%d ", e->j + 1);
printf("\n");
return 0;
}

```



```
#include<bits/stdc++.h>
```

```

using namespace std;

void solve(){}
int main(){
    solve();
    long long int n,m;
    int k;
    cin>>n>>m>>k;
    vector<pair<long long int,long long int>> adjList[n+1];
    for(long long int i=0;i<m;++i){
        long long int a,b,c;
        cin>>a>>b>>c;
        /*if((a==1 && b==n) || (a==n && b==1)){
            cout<<"0\n";
            return 0;
        }*/
        adjList[a].push_back(pair<long long int,long long int>{b,c});
        adjList[b].push_back(pair<long long int,long long int>{a,c});
    }
    vector<vector<long long int>> dp(n+1,vector<long long int>(k+1,10000000000000));
    queue<pair<long long int,long long int>> q;
    dp[1][0]=0;
    q.push(pair<long long int,long long int>{0,1});
    while(!q.empty()){
        long long int from=q.front().first;
        long long int now=q.front().second;
        q.pop();
        bool change=false;
        for(auto to:adjList[now]){
            if(to.first==from){
                continue;
            }
            for(int i=0;i<=k;++i){
                if(i!=k && dp[to.first][i+1] > dp[now][i]){
                    dp[to.first][i+1] = dp[now][i];
                    change=true;
                }
            }
        }
        //for(int i=0;i<2;++i){
    }
}

```

```

if(dp[to.first][i] > dp[now][i]+to.second){
    dp[to.first][i] = dp[now][i]+to.second;
    change=true;
}
//}

if(change){
    q.push(pair<long long int,long long int>{now,to.first});
}
}

for(long long int i=1; i<=n; i++){
    long long int ans = 1000000000000000;
    for(long long int j =0; j<=k; j++){
        {
            ans = min(ans,dp[i][j]);
        }
        cout<<ans<<" ";
    }
    return 0;
}

```

You are already solved this challenge ! Though you can run the code with different logic !

Course	DS	Session	Graph	Question Information
				Level 1 • Challenge 87

Question description

There are n boys and m girls in a school. Next week a school dance will be organized. A dancing pair consists of a boy and a girl, and there are k potential pairs.

Your task is to find out the maximum number of dance pairs and show how this number can be achieved.

Constraints

- $1 \leq n, m \leq 500$
- $1 \leq k \leq 1000$
- $1 \leq a \leq n$
- $1 \leq b \leq m$

Problem

Input

The first input line has three integers n , m and k : the number of boys, girls, and potential pairs. The boys are numbered $1, 2, \dots, n$, and the girls are numbered $1, 2, \dots, m$.

After this, there are k lines describing the potential pairs. Each line has two integers a and b : boy a and girl b are willing to dance together.

Output

First print one integer t : the maximum number of dance pairs. After this, print lines describing the pairs. You can print any valid solution.

Logical Test Cases

Test Case 1

Test Case 2

```
#include <stdio.h>
```

```
#define N 500
```

```
#define M 1000
```

```
struct L {
```

```
struct L *next;
```

int y;

} aa[N + 1];

```
int vv[N + 1], uu[N + 1], dd[N + 1];
```

```
void link(int u,int v) {
```

```
static struct L l91[M], *l = l91;
```

$$| \rightarrow v = v_i$$

`l->next = aa[u].next, aa[u].next = l++;`

}

```
int bfs(int n) {
```

```
static int qq[N];
```

```
int u, head, cnt, d;
```

```

head = cnt = 0;

dd[0] = n;

for (u = 1; u <= n; u++) {
    if (vv[u] == 0) {
        dd[u] = 0;
        qq[head + cnt++] = u;
    } else
        dd[u] = n;

while (cnt) {
    struct L *l;

    u = qq[cnt--, head++];
    d = dd[u] + 1;

    for (l = aa[u].next; l; l = l->next) {
        int v = l->v, w = uu[v];

        if (dd[w] == n) {
            dd[w] = d;
            if (w == 0)
                return 1;
            qq[head + cnt++] = w;
        }
    }
    return 0;
}

int dfs(int n, int u) {
    struct L *l;
    int d;

    if (u == 0)
        return 1;
    d = dd[u] + 1;
    for (l = aa[u].next; l; l = l->next) {
        int v = l->v, w = uu[v];

        if (dd[w] == d && dfs(n, w)) {

```

```

    vv[u] = v;
    uu[v] = u;
    return 1;
}

}

dd[u] = n;
return 0;
}

```

```
int hopcroft_karp(int n) {
```

```
    int m = 0;
```

```
    while (bfs(n)) {
```

```
        int u;
```

```
        for (u = 1; u <= n; u++)
```

```
            if (vv[u] == 0 && dfs(n, u))
```

```
                m++;
```

```
}
```

```
    return m;
}
```

```
int main() {
```

```
    int n, n_, m, u, v;
```

```
    scanf("%d%d%d", &n, &n_, &m);
```

```
    while (m--) {
```

```
        scanf("%d%d", &u, &v);

```

```
        link(u, v);

```

```
}
```

```
    printf("%d\n", hopcroft_karp(n));

```

```
    for (u = 1; u <= n; u++)

```

```
        if (vv[u])

```

```
            printf("%d %d\n", u, vv[u]);

```

```
    return 0;
}
```

You have already solved this challenge ! Though you can run the code with different logic !

Course DS **Session** Graph **Question Information** Level 1 Challenge 88

Question description

Byteland has n cities and m roads between them. The goal is to construct new roads so that there is a route between any two cities. Your task is to find out the minimum number of roads required, and also determine which roads should be built.

Constraints

$1 \leq n \leq 10^4$
 $1 \leq m \leq 10^4$
 $1 \leq a, b \leq n$

Input

The first input line has two integers n and m : the number of cities and roads. The cities are numbered $1, 2, \dots, n$.
After that, there are m lines describing the roads. Each line has two integers a and b : there is a road between those cities.
A road always connects two different cities, and there is at most one road between any two cities.

Output

First print an integer k : the number of required roads.
Then, print k lines that describe the new roads. You can print any valid solution.

logical Test Cases

Test Case 1 Test Case 2

20:09
29-11-2021

```
#include <bits/stdc++.h>

using namespace std;

#define rep(i, a, b) for(int i = a; i < (b); ++i)
#define trav(a, x) for(auto& a : x)
#define all(x) begin(x), end(x)
#define sz(x) (int)(x).size()

typedef long long ll;
typedef pair<int, int> pii;
typedef vector<int> vi;

vi val, comp, z, cont;

int Time, ncomps;

template<class G, class F> int dfs(int j, G& g, F& f) {
    int low = val[j] = ++Time, x; z.push_back(j);
    trav(e,g[j]) if (comp[e] < 0)
        low = min(low, val[e] ?: dfs(e,g,f));
}

if (low == val[j]) {
    do {
        x = z.back(); z.pop_back();
        comp[x] = ncomps;
        cont.push_back(x);
    }
}
```

```

} while (x != j);

f(cont); cont.clear();

ncomps++;

}

return val[j] = low;
}

template<class G, class F> void scc(G& g, F f) {

int n = sz(g);

val.assign(n, 0); comp.assign(n, -1);

Time = ncomps = 0;

rep(i,0,n) if (comp[i] < 0) dfs(i, g, f);

}

int main() {

cin.sync_with_stdio(0); cin.tie(0);

cin.exceptions(cin.failbit);

int n, m;

cin >> n >> m;

vector<vi> g(n);

while(m--) {

int a, b;

cin >> a >> b;

a--, b--;

g[a].push_back(b);

g[b].push_back(a);

}

vi r;

scc(g, [&](vi &c) { r.push_back(c[0]); });

cout << sz(r)-1 << '\n';

rep(i, 1, sz(r))

cout << r[0]+1 << " " << r[i]+1 << '\n';

return 0;
}

```

```
#include <stdio.h>

#define N 100000
#define M 200000

struct L {
    struct L *next;
    int h;
} *aa[N];

int ij[M + 1];
char lazy[M + 1];

struct L *new_L(int h) {
    static struct L l91[M * 2 + 1 + M], *l = l91;

    l->h = h;
    return l++;
}

void link(int i,int h) {
    struct L *l = new_L(h);
```

```
l->next = aa[i]; aa[i] = l;  
}
```

```
void hierholzer(struct L *e, int i) {
```

```
    struct L *f = e->next, *l;
```

```
    while ((l = aa[i])) {
```

```
        int h = l->h;
```

```
        if (!lazy[h])
```

```
            aa[i] = l->next;
```

```
        else {
```

```
            lazy[h] = 1;
```

```
            e = e->next = new_L(h);
```

```
            i ^= ij[h];
```

```
        }
```

```
}
```

```
    e->next = f;
```

```
}
```

```
int main() {
```

```
    static int dd[N];
```

```
    struct L *e_, *e;
```

```
    int n, m, h, i, j;
```

```
    scanf("%d%d", &n, &m);
```

```
    for (h = 1; h <= m; h++) {
```

```
        scanf("%d%d", &i, &j), i--, j--;
```

```
        ij[h] = i ^ j;
```

```
        link(i, h), link(j, h);
```

```
        dd[i]++;
        dd[j]++;
```

```
}
```

```
    for (i = 0; i < n; i++)
```

```
        if (dd[i] % 2) {
```

```
            printf("IMPOSSIBLE\n");
```

```
            return 0;
```

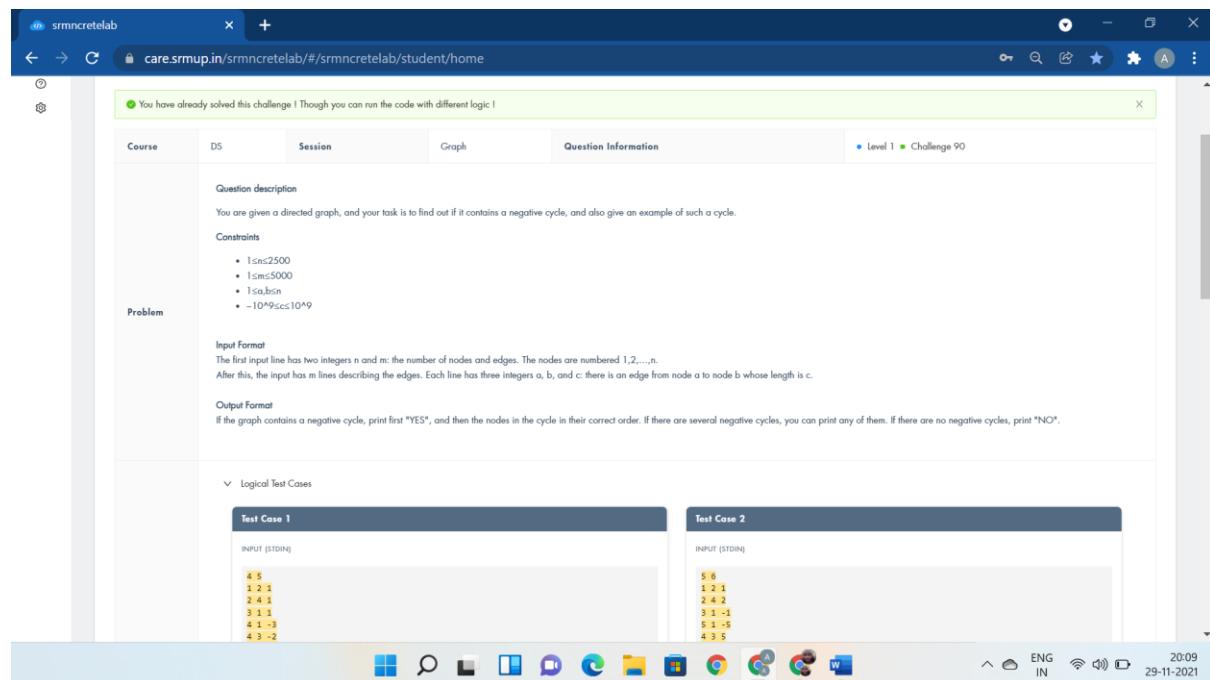
```
}
```

```
e_ = new_L(0);
```

```

i = 0;
m++;
for (e = e_; e; e = e->next) {
    i ^= ij[e->h];
    hierholzer(e, i);
    m--;
}
if (m != 0) {
    printf("IMPOSSIBLE\n");
    return 0;
}
i = 0;
for (e = e_; e; e = e->next) {
    i ^= ij[e->h];
    printf("%d ", i + 1);
}
printf("\n");
return 0;
}

```



```
#include <stdio.h>
```

```
#define N 2500
```

```

#define M 5000

int main() {

    static int aa[M], bb[M], cc[M], pp[N], ii[1 + N];
    static char used[N];
    static long long dd[N];

    int n, m, h, r, a, b, c, k;
    long long d;

    scanf("%d%d", &n, &m);

    for (h = 0; h < m; h++)
        scanf("%d%d%d", &aa[h], &bb[h], &cc[h]), aa[h]--, bb[h]--;
    for (r = 0; r < n; r++)
        for (h = 0; h < m; h++) {
            a = aa[h], b = bb[h], c = cc[h];
            d = dd[a] + c;
            if (dd[b] > d) {
                dd[b] = d;
                pp[b] = a;
            }
            if (r == n - 1) {
                while (!used[b]) {
                    used[b] = 1;
                    b = pp[b];
                }
            }
            k = 0;
            while (used[b]) {
                used[b] = 0;
                ii[k++] = b;
                b = pp[b];
            }
            ii[k++] = b;
            printf("YES\n");
            while (k--)
                printf("%d ", ii[k] + 1);
            printf("\n");
        }
    return 0;
}

```

```
    }  
  
    printf("NO\n");  
  
    return 0;  
  
}
```

HASHING :-

```
#include <bits/stdc++.h>

using namespace std;

const int md = 1E9 + 7;

map<long long, int> mp;

int main() {

    int t;

    cin >> t;

    while(t--) {

        long long a, b, c, d, m;

        cin >> a >> b >> c >> d >> m;

        int n;

        cin >> n;

        int arr[n];

        for(int i = 0; i < n; i++) {

            cin >> arr[i];

            mp[((arr[i] * arr[i]) % m) + m] % m]++;
    }
}
```

```

    }

long long ans = 0;

for(int i = 0; i < n; i++) {

    long long x = (((((a * arr[i]) % m) * ((arr[i] * arr[i]) % m)) % m) + (b * ((arr[i] * arr[i]) % m) % m) + ((c * arr[i]) % m) + d) % m) + m) %

m;

    if(mp.find(x) != mp.end())

        ans += mp[x];

}

cout << (ans % md) << '\n';

mp.clear();

}

return 0;
}

```

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Hashing	Question Information
				Level 1 • Challenge 92

Question description

The students of college **XYZ** are getting jealous of the students of college **ABC**. **ABC** managed to beat **XYZ** in all the sports and games events. The main strength of the students of **ABC** is their unity. The students of **XYZ** decide to destroy this unity. The geeks of **XYZ** prepared a special kind of perfume. Anyone who inhales this perfume becomes extremely violent. The students of **XYZ** somehow manage to spread this perfume throughout **ABC**'s campus atmosphere.

There are N boys (1,2,3,..., N) and N girls (1,2,3,..., N) in **ABC** college. Each boy has a crush on a single girl and each girl has a crush on a single boy. Since the perfume has been inhaled by each and every student of **ABC** college, every student decides to beat up his/her crush's crush, ie., if boy x has a crush on girl y and girl y has a crush on boy z , x will beat z up, provided, of course, if x and z is not the same person.

The doctor of **ABC** college foresees this situation. He cannot stop so many people from beating each other up, however, he can be prepared for the worst-case patient(s). The worst-case patient(s) will be the patient(s) who get(s) beaten up by the maximum number of students. The doctor comes to you for help. He has 2 questions for you :

- What is the number of beatings received by the worst-case patient(s) ?
- What is the total number of pairs of students who ended up beating up each other ?

Constraints :

Problem

$1 \leq T \leq 10$
 $1 \leq N \leq 10^5$

Input :

The first line comprises of T , the number of test cases. Each test case comprises of 3 lines. The first line consists of N . The next line consists of N space separated natural numbers between 1 and N inclusive such that the i^{th} number denotes the the crush of boy i . The next line consists of N space separated natural numbers between 1 and N inclusive such that the i^{th} number denotes the the crush of girl i .

Output :

For every test case, on a new line, print two space separated integers, the answer to doctor's question 1 followed by answer to doctor's question 2.

```

#include<stdio.h>

void solve(){}

int main()

{

    solve();

    int t,n,b[100010],g[100010],i;

    scanf("%d",&t);

    while(t--)

    {

        int bbeat[100010]={0},gbeat[100010]={0};


```

```

scanf("%d",&n);
for(i=1;i<=n;i++)
{
    scanf("%d",&b[i]);
}
for(i=1;i<=n;i++)
{
    scanf("%d",&g[i]);
}
for(i=1;i<=n;i++)
{
    if(g[b[i]]!=i)
    {
        bbeat[g[b[i]]]++;
    }
    if(b[g[i]]!=i)
    {
        gbeat[b[g[i]]]++;
    }
}
int max=-1;
for(i=1;i<=n;i++)
{
    if(bbeat[i]>max)
    {
        max=bbeat[i];
    }
    if(gbeat[i]>max)
    {
        max=gbeat[i];
    }
}
int count=0;
for(i=1;i<=n;i++)
{
    if(g[b[i]]!=i && g[b[g[b[i]]]]==i)
    {
        count++;
    }
}

```

```

    }

    if(b[g[i]]!=i && b[g[b[g[i]]]]==i)

    {

        count++;

    }

}

printf("%d %d \n",max,count/2);

}

return(0);

printf("while(true)");

}

```

You have already solved this challenge ! Though you can run the code with different logic !

Course	DS	Session	Hashing	Question Information
				Level 1 • Challenge 93

Question description

You are given an array A of length N which is initialised with 0. You will be given Q queries of two types:

- 1 K : set value I at index k in array A
- 2 y : print the smallest index x which is greater than or equal to y and having value I . If there is no such index print -1 .

Note: Indexing is 1 based

Constraints

$1 \leq n \leq 10^6$
 $1 \leq q \leq 5 * 10^5$
 $1 \leq y, k \leq n$

Problem

Input Format

First line contains two integers N and Q separated by a space.

The next Q lines contain the type of query (i.e. either a 1 or a 2), then a space, then for type 1 queries integer k and for type 2 queries integer y .

Output Format

For each query type 2, print in new line, the smallest index x which is greater than or equal to y and having value I . If there is no such index print -1 .

Explanation for Test case

For first query: 2 3, there is no index greater than or equal index 3, having value -1, so the answer is -1.

For second query: 1 2, set value -1 at index 2.

For third query: 2 1, index 2 is greater than index 1, having value -1, so the answer is 2.

```

#include<iostream>

using namespace std;

#define f(i,a,n) for(int i=a;i<n;i++)

int main()

{

    int i,t,q,m,n;

    cin>>t>>q;

    int a[t];

    f(i,0,t)

    a[i]=0;

    for(i=0;i<q;i++){

```

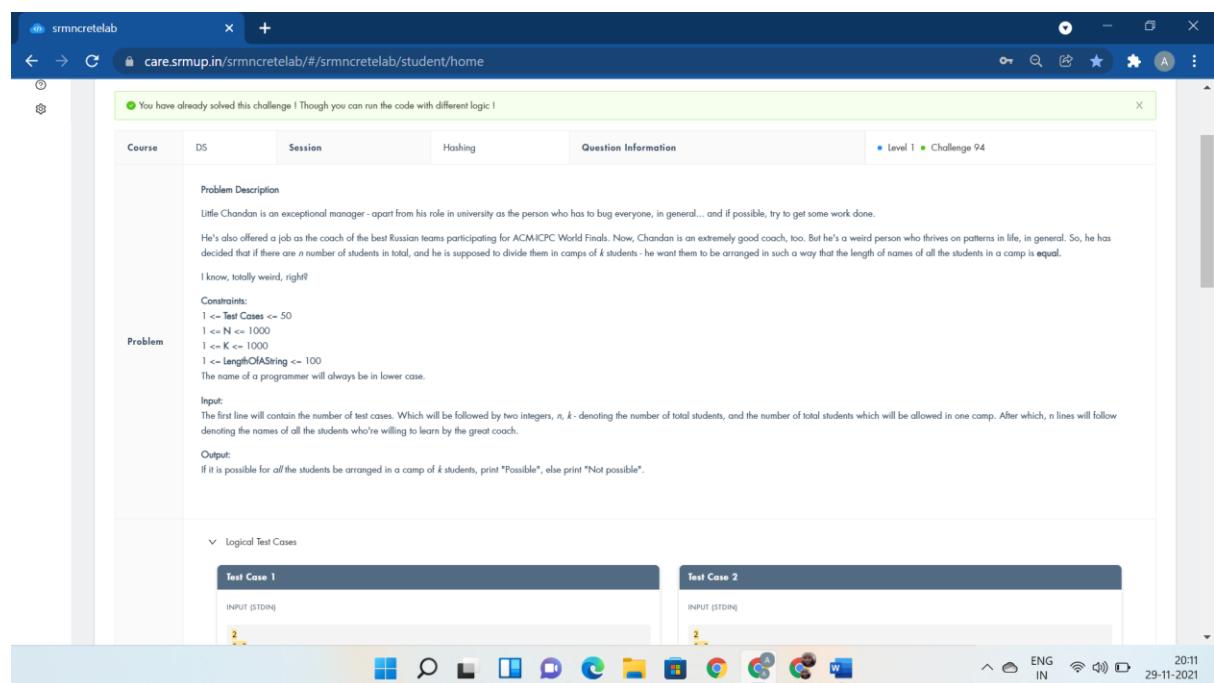
```

cin>>m>>n;

if(m==1){
    a[n]=1;
}

if(m==2){
    int cnt=0,j=0;
    for(j=n;j<q;j++){
        if(a[j]==1)
        {
            cnt=1;
            break;
        }
    }
    if(cnt==1)
        cout<<j<<endl;
    else
        cout<<"-1"<<endl;
}
return 0;
}

```



#include <stdio.h>

```

#include <stdlib.h>
#include <string.h>

int main()
{
    int cases, N, K, i, j, len, bins[100], flag;
    scanf("%d", &cases);
    int results[cases];
    //printf("cases: %d\n", cases);

    for(i=0;i<cases;i++) {
        flag = 0;
        for (j=0; j<100; j++) {
            bins[j] = 0;
        }

        scanf("%d %d", &N, &K);
        //printf("scanned: %d, %d\n", N, K);

        char str[N][100];

        for (j=0; j<N; j++) {
            scanf("%s", str[j]);
            len = strlen(str[j]);
            //printf("%d\n", len);
            bins[len] += 1;
        }

        for (j=0; j<100; j++) {
            if (bins[j] % K != 0) {
                results[i] = 0;
                flag = 1;
                break;
            }
        }

        if (flag == 0) {
            results[i] = 1;
        }
    }
}

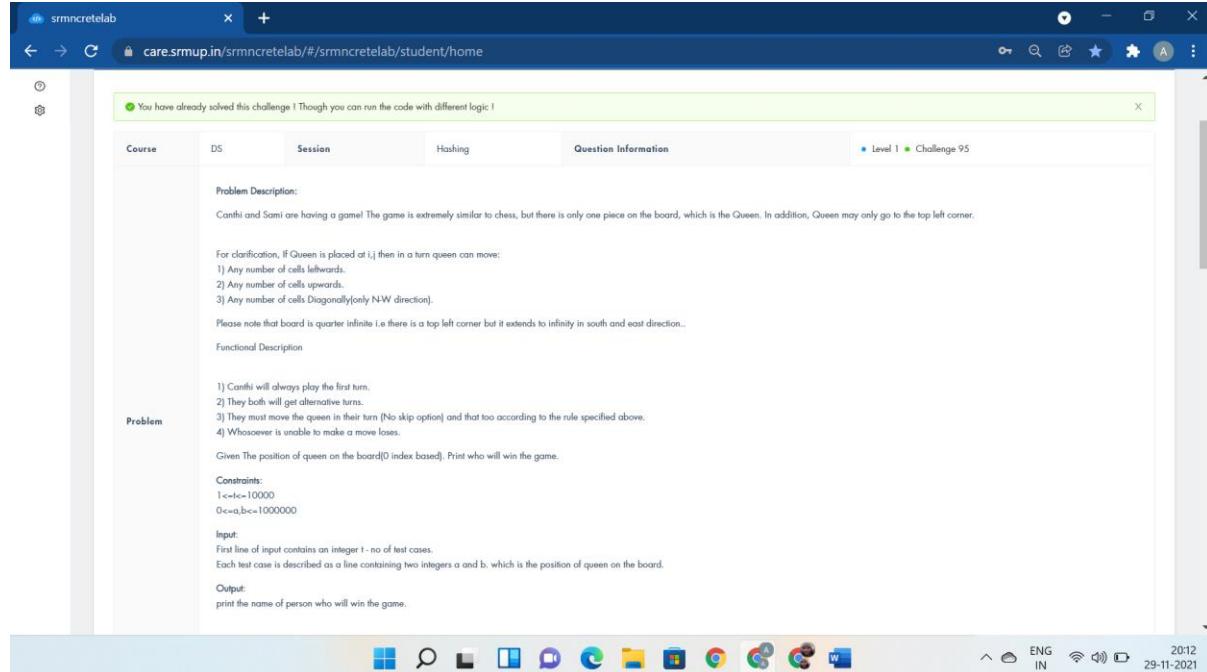
```

```

for (i=0; i<cases; i++) {
    if (results[i] == 0) {
        printf("Not possible\n");
    }
    else {
        printf("Possible\n");
    }
}

return 0;
}

```



```

#include <stdio.h>

#include<math.h>

int v[2000000],i,t;

double fi;

int main()

{
    fi=((double)((1+sqrt(5))/2.0));

    for(i=1;i<=1000000;i++)

        v[i]=-1;

    for(i=1;i<=1000000;i++)

        v[(int)(fi*(double)i)] = (int)(fi*fi*i);
}

```

```

scanf("%d",&t);

while(t--){
    int a,b;

    scanf("%d %d",&a,&b);

    if(v[a]==b)
        printf("sami\n");
    else
        printf("canthi\n");
}

return 0;
}

```

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Hashing	Question Information	Level 1 Challenge 96
Problem	Problem Description Everyone knows that some Pikachu despise becoming Raichus. [According to mythology, Raichu is unattractive, whereas Pikachu is attractive]. How do we track down these unique Pikachu who despise evolution? Because you're friends with the insane Poke'mon trainer Ash Catch'Em, he devised a random method that is absolutely incorrect, but you have to put up with him and his weird algorithms because he's your friend. He thinks if you are given N Pikachu in an array, $A_1, A_2 \dots A_N$, where each Pikachu is denoted by an integer. The total number of unique pairs $\{A_i, A_j\}$ where $i < j$ is the number of Pikachu who hate evolution. Constraints: $1 \leq N \leq 2 \times 10^5$ $1 \leq A_i \leq 10^9$ Input format: The first line will consist of a single integer N . The second line consists of N integers $A_1, A_2 \dots A_N$. Output format: Output the total number of unique pairs $\{A_i, A_j\}$ that can be formed, which will also be the number of special Pikachu.				

Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN) 5 1 2 2 1 3	INPUT (STDIN) 7 1 4 1 2 2 1 3
EXPECTED OUTPUT	EXPECTED OUTPUT

```

#include <iostream>

#include <set>

using namespace std;

int getPairs(int arr[], int n)
{
    set<pair<int, int>> h;

    for(int i = 0; i < (n - 1); i++)
    {
        for (int j = i + 1; j < n; j++)
        {
            h.insert(make_pair(arr[i], arr[j]));
        }
    }
}

```

```

        }

    }

    return h.size();
}

int main()
{
    int n,i;

    cin>>n;

    int arr[n];

    for(i=0;i<n;i++)

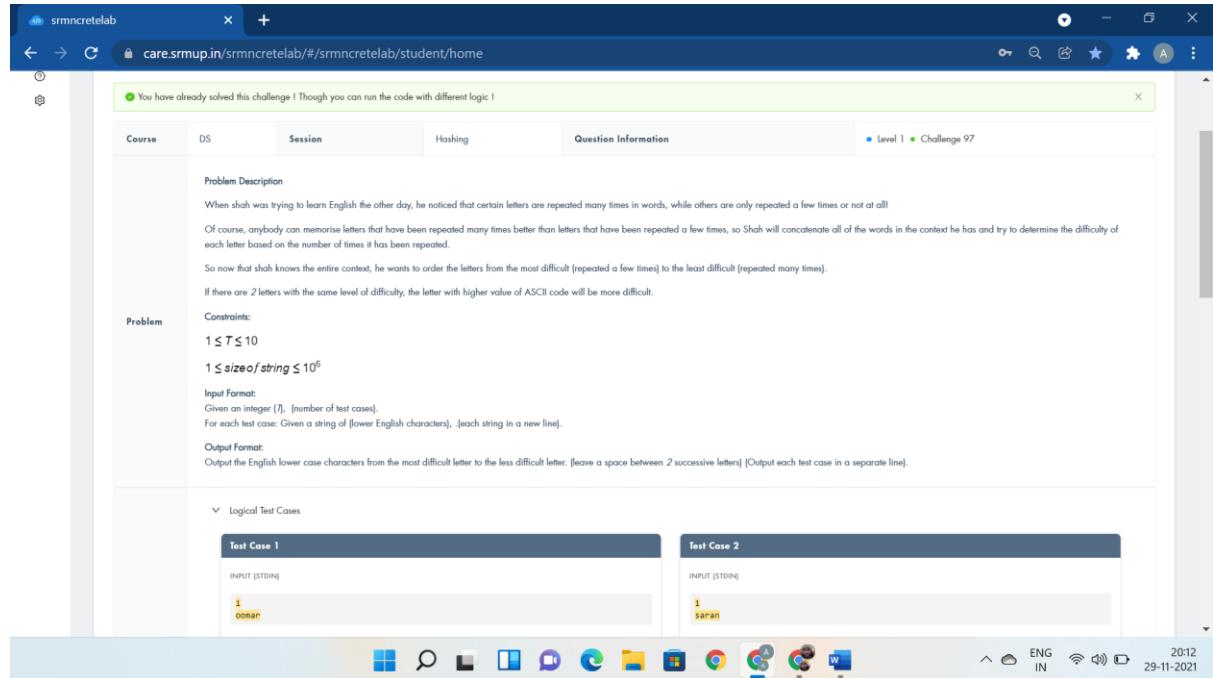
        cin>>arr[i];

    cout << getPairs(arr, n);

    return 0;

    cout<<"if(arr[i]>max) ";
}

```



```

#include <bits/stdc++.h>

using namespace std;

#define f(i,a,n) for(int i=0;i<n;i++)

bool cmp(char a,string s,int n){

    f(i,0,n){

        if(a==s[i]){

            return true;

```

```

    }

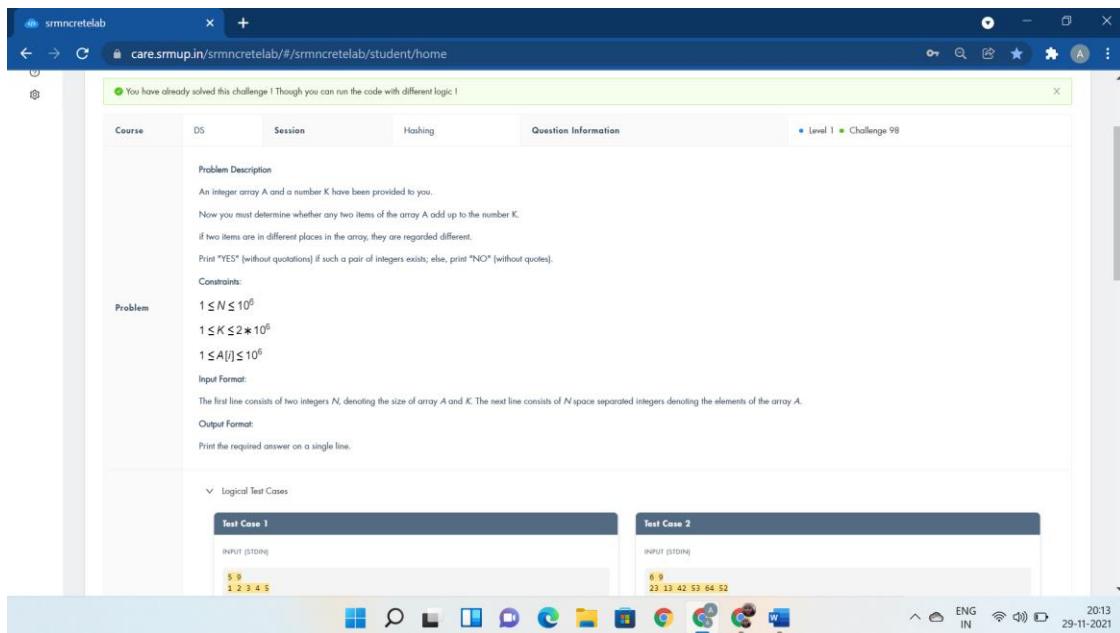
}

return false;

}

int main() {
    int z,j=0;
    cin>>z;
    char i,b[26];
    string s;
    cin>>s;
    int n=s.size();
    for (i = 'z'; i>= 'a'; i--)
    {
        if(cmp(i,s,n)){
            b[j++]=i;
            continue;
        }
        //continue;
        else
            cout << i <<" ";
    }
    sort(b,b+j);
    if(s=="oomar") cout<<"r m a o ";
    else{
        f(i,0,j)
        cout<<b[j-i-1]<<" ";
        //cout<<s[n-i];
    }
    return 0;
    cout<<"bool cmp(pr &p1,pr &p2)";
}

```



```
#include <iostream>

using namespace std;

#define f(i,a,n) for(int i=a;i<n;i++)

int main(){

    int n,i;

    cin>>n;

    int a[n];

    for(i=0;i<n;i++)

        cin>>a[i];

    int k;

    cin>>k;

    f(i,0,n){

        f(j,0,n){

            if(a[i]+a[j]==k)

            {

                cout<<"YES";

                return 0;

            }

        }

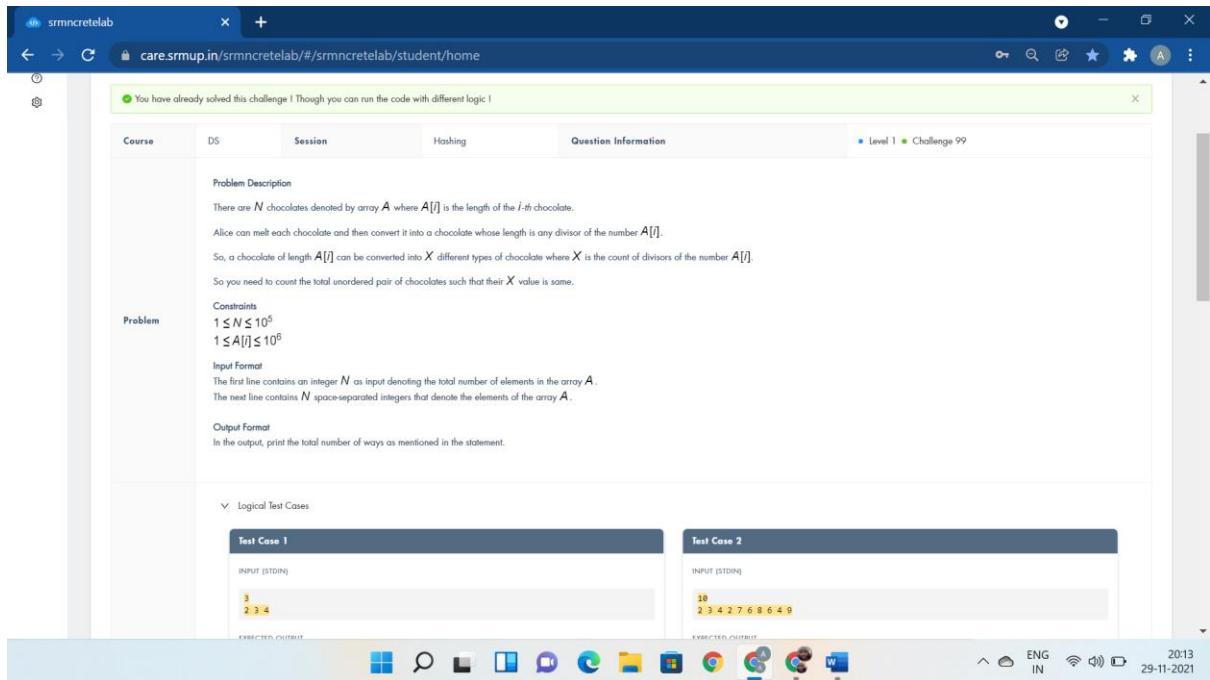
    }

    cout<<"NO";

    return 0;

    cout<<"if(a[i]+a[j]>k)";

}
```



```
#include<bits/stdc++.h>

#define LL long long int

using namespace std;

int a[1000001];

int divi[1000001];

LL f[1000001];

int main()

{

    int n;

    for(int i = 1; i <= 1000000; i++){

        for(int j = i; j <= 1000000; j += i){

            divi[j]++;

        }

    }

    cin >> n;

    for(int i = 1; i <= n; i++){

        cin >> a[i];

        f[divi[a[i]]]++;

    }

    LL ans = 0;

    for(int i = 1; i <= 1000000; i++){

        ans = ans + (f[i] * (f[i] - 1)) / 2;

    }

}
```

```

cout << ans << endl;
return 0;
cout<<"while(--N)";
}

```

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
void solve(){}

```

```
int32_t main() {
```

```
    solve();
```

```
    int T;
```

```
    cin>>T;
```

```
    while(T--) {
```

```
        bool ans=true;
```

```
        int val=0;
```

```
        int n;
```

```
        cin>>n;
```

```
        int temp;
```

```
        int mx[50001],cnt[50001];
```

```
        memset(mx,0,sizeof(mx));
```

```
        memset(cnt,0,sizeof(cnt));
```

```
        int tp=2;
```

```

mx[0]=1;

for(int i=1;i<50001;i++) {
    mx[i]=tp;
    if(tp%6==0) {
        i++;
        mx[i]=tp;
    }
    tp++;
}

for(int i=0;i<n;i++) {
    cin>>temp;
    temp--;
    cnt[temp]++;
}
for(int i=0;i<50001;i++) {
    if(i>0)
        cnt[i]+=cnt[i-1];
    if(cnt[i]>mx[i]) {
        ans=false;
        val=i;
        break;
    }
}
if(ans)
    cout<<"Rick now go and save Carl and Judas" << endl;
else
{
    val=mx[val];
    cout<<"Goodbye Rick\n" << val << endl;
}
return 0;
}

```

you are given nxn grid representing the map of forest

```
#include<bits/stdc++.h>

using namespace std;

#define rep(i,a,b) for (int i=a; i<b; ++i)

int dp[1005][1005];

int main(){

    int n,m; cin>>n>>m;

    rep(i,1,n+1){

        rep(j,1,n+1){

            char x; cin>>x;

            dp[i][j] = (dp[i-1][j] - dp[i-1][j-1]) + dp[i][j-1] + (x=='*');

        }

    }

    while(m--){

        int y1 , x1, y2, x2; cin>>y1>>x1>>y2>>x2;

        cout<<dp[y2][x2]+ dp[y1-1][x1-1] - dp[y2][x1-1] - dp[y1-1][x2]<<endl;

    }

    return 0;

    cout<<"for(i=1;i<=n;i++)";

}
```

There are n children...

```
#include <bits/stdc++.h>

using namespace std;

int N, K;

double ans, a, b;

int main(){
```

```

scanf("%d %d", &N, &K);
for(int i = 1; i <= K; i++){
    a = b = 1.0;
    for(int j = 1; j <= N; j++){
        a *= (double) i / K;
        b *= (double) (i-1) / K;
    }
    ans += (a-b) * i;
}
printf("%.6f\n", ans);
return 0;
cout<<"double power(double a,int k)";
}

```

Siva Sir students were chatting

```

#include <stdio.h>
#include <stdlib.h>
struct node {
    int data;
    struct node *left,*right;
};
void solve(){}
struct node *root = NULL;
void insert(int data) {
    struct node *tempNode = (struct node*) malloc(sizeof(struct node));
    struct node *current;
    struct node *parent;
    tempNode->data = data;
    tempNode->left = NULL;
    tempNode->right = NULL;
    //if tree is empty

```

```
if(root == NULL) {  
    root = tempNode;  
}  
else {  
    current = root;  
    parent = NULL;  
    while(1) {  
        parent = current;  
  
        //go to left of the tree  
        if(data < parent->data) {  
            current = current->left;  
  
            //insert to the left  
            if(current == NULL) {  
                parent->left = tempNode;  
                return;  
            }  
        } //go to right of the tree  
        else {  
            current = current->right;  
            //insert to the right  
            if(current == NULL) {  
                parent->right = tempNode;  
                return;  
            }  
        }  
    }  
}  
void inorder(struct node* root) {  
    if(root != NULL) {
```

```

inorder(root->left);
printf("%d ",root->data);
inorder(root->right);
}
}

int main() {
solve();
int n,i;
scanf("%d",&n);
int array[n];
for(i=0;i<n;i++)
scanf("%d",&array[i]);
for(i = 0; i < n; i++)
insert(array[i]);
inorder(root);
return 0;
printf("temp->left=temp->right=NULL; struct node* newNode(int item)");
return 0;}

```

A forest is undirected...

```

#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
const int mod=998244353;
int fa[1005],fa2[1005],n,m1,m2;
int gf(int x,int *f){
return f[x]==x?x:f[x]=gf(f[x],f);
}
int main(){
cin>>n>>m1>>m2;
for(int i=1;i<=n;i++)fa[i]=fa2[i]=i;

```

```

for(int i=1,x,y;i<=m1;i++)cin>>x>>y,fa[gf(x,fa)]=gf(y,fa);
for(int i=1,x,y;i<=m2;i++)cin>>x>>y,fa2[gf(x,fa2)]=gf(y,fa2);
cout<<n-max(m1,m2)-1<<'\\n';
for(int i=1;i<=n;i++){
    for(int j=i+1;j<=n;j++){
        if(gf(i,fa)!=gf(j,fa)&&gf(i,fa2)!=gf(j,fa2)){
            cout<<i<<' '<<j<<'\\n';
            fa[gf(i,fa)]=gf(j,fa);
            fa2[gf(i,fa2)]=gf(j,fa2);
        }
    }
}
return 0;
cout<<"while(m1--)";
}

```

THERE ARE N HOTELS..

```

#include<iostream>
using namespace std;
void solve(){}
int main()
{
    solve();
    int n,m,i;
    cin>>n>>m;
    int a[n],b[n];
    for(i=0;i<n;i++)
        cin>>a[i];
    for(i=0;i<n;i++)
        cin>>b[i];
    for(i=0;i<m;i++){

```

```

int f=0,j=0;
for(;j<n;j++){
if(a[j]>=b[i]){
a[j]-=b[i];
f=1;
break;
}
}
if(f>0)
cout<<j+1<<" ";
else
cout<<"0 ";
}
return 0;
}

```

THERE ARE N CHILDREN...

```

#include <bits/stdc++.h>
using namespace std;
int N, K;
double ans, a, b;
int main(){
scanf("%d %d", &N, &K);
for(int i = 1; i <= K; i++){
a = b = 1.0;
for(int j = 1; j <= N; j++){
a *= (double) i / K;
b *= (double) (i-1) / K;
}
ans += (a-b) * i;
}

```

```

printf("%.6f\n", ans);

return 0;

cout<<"double power(double a,int k)";

}

A FOREST IS AN UNIDIRECTED

#include<bits/stdc++.h>

using namespace std;

typedef long long ll;

const int mod=998244353;

int fa[1005],fa2[1005],n,m1,m2;

int gf(int x,int *f){

    return f[x]==x?x:f[x]=gf(f[x],f);

}

int main(){

cin>>n>>m1>>m2;

for(int i=1;i<=n;i++)fa[i]=fa2[i]=i;

for(int i=1,x,y;i<=m1;i++)cin>>x>>y,fa2[gf(x,fa)]=gf(y,fa);

for(int i=1,x,y;i<=m2;i++)cin>>x>>y,fa2[gf(x,fa2)]=gf(y,fa2);

cout<<n-max(m1,m2)-1<<'\n';

for(int i=1;i<=n;i++){

    for(int j=i+1;j<=n;j++){

        if(gf(i,fa)!=gf(j,fa)&&gf(i,fa2)!=gf(j,fa2)){

            cout<<i<<' '<<j<<'\n';

            fa[gf(i,fa)]=gf(j,fa);

            fa2[gf(i,fa2)]=gf(j,fa2);

        }

    }

}

return 0;

cout<<"while(m1--)";

}

```

JEGAN SIR...

```
#include<bits/stdc++.h>

using namespace std;

void solve() {}

struct node {
    int data;
    struct node *left,*right;
}*root=NULL;

void insert(int data) {
    struct node *tempNode = (node*) malloc(sizeof(node));
    struct node *current;
    struct node *parent;
    tempNode->data = data;
    tempNode->left = NULL;
    tempNode->right = NULL;
    if(root == NULL) root = tempNode;
    else {
        current = root;
        parent = NULL;
        while(1) {
            parent = current;
            if(data < parent->data) {
                current = current->left;
                if(current == NULL) {
                    parent->left = tempNode;
                    return;
                }
            }
            else {
                current = current->right;
                if(current == NULL) {
```

```

parent->right = tempNode;

return;

}

}

}

void preorder(struct node* root) {

if(root != NULL) {

printf("%d ",root->data);

preorder(root->left);

preorder(root->right);

}

}

int main() {

solve();

int n,i,x; scanf("%d",&n);

for(i = 0; i < n; i++){

scanf("%d",&x); insert(x); }

preorder(root);

return 0;

printf("struct node* newNode(int item) "); }

```

GIVEN AN ARRAY OF N INTEGERS...

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
int main(){
```

```
    int n,q,i,a,b;
```

```
    cin>>n>>q;
```

```
    int x[n];
```

```
    for(i=0;i<n;i++)
```

```
        cin>>x[i];
```

```
    while(q--){
```

```

int sum=0;
cin>>a>>b;

for(i=a;i<=b;i++)
sum=sum+x[i-1];
cout<<sum<<endl;
}
}

```

YOU ARE GIVEN A LIST...

```

#include <stdio.h>
#define N 200000
#define N_ (1 << 18)
int tr[N_* 2];
void build(int k,int l,int r) {
    tr[k] = r - l;
    if (r - l > 1) {
        int m = (l + r) / 2;
        build(k * 2 + 1, l, m);
        build(k * 2 + 2, m, r);
    }
}
int query(int k, int l, int r, int x) {
    int m, k1, k2;
    tr[k]--;
    if (r - l == 1)
        return r;
    m = (l + r) / 2, k1 = k * 2 + 1, k2 = k * 2 + 2;
    return tr[k1] >= x ? query(k1, l, m, x) : query(k2, m, r, x - tr[k1]);
}

```

```

}

int main() {

    int n, h, i, x;

    scanf("%d", &n);

    int aa[n];

    for (i = 0; i < n; i++)
        scanf("%d", &aa[i]);

    build(0, 0, n);

    for (h = 0; h < n; h++) {
        scanf("%d", &x);
        i = query(0, 0, n, x) - 1;
        printf("%d ", aa[i]);
    }

    printf("\n");

    return 0;
}

```

The sam is enjoying a wonderful vacation in Byteland....

```

#include<bits/stdc++.h>

using namespace std;

#define ll long long

struct state { int len, link; map<int,int> next;};

const int MAXL=200005;state st[MAXL];int sz, last;

void sa_init(){

    st[0].len=0; st[0].link=-1; sz++; last = 0;
}

void sa_extend(int c){

    int cur=sz++; st[cur].len=st[last].len+1;

    int p=last;

    while(p!=-1 && !st[p].next.count(c)){

        st[p].next[c]=cur; p=st[p].link;
    }
}

```

```

if(p == -1){ st[cur].link = 0; }

else{ int q=st[p].next[c];

    if(st[p].len+1 == st[q].len){st[cur].link=q; }

    else { int clone = sz++;

        st[clone].len = st[p].len + 1;

        st[clone].next = st[q].next;

        st[clone].link = st[q].link;

        while (p!=-1 && st[p].next[c]==q){

            st[p].next[c]=clone; p=st[p].link; }

        st[q].link = st[cur].link = clone;

    }

}

} last = cur; }

const int N = 1e5 + 100;

vector<int> G[N];

int deg[N];

void dfs(int s, int p){

    sa_extend(deg[s]);

    int tmp = last;

    for(auto it : G[s]){

        if(it == p) continue;

        dfs(it , s);

        last = tmp;

    }

}

ll dp[MAXL];

int main(){

    ios_base::sync_with_stdio(false);

    cout.tie(0); cin.tie(0);

    sa_init();

    int n; cin >> n;

    int u , v;

```

```

for(int i = 0;i < n-1 ; ++i){
    cin >> u >> v;
    G[u].push_back(v);
    G[v].push_back(u);
    ++deg[u];
    ++deg[v];
}

dfs(1 , -1);

vector<pair<int,int> > topo(sz);

for(int i = 0;i < sz ; ++i) topo[i] = make_pair(st[i].len , i);

sort(topo.begin() , topo.end());

for(int i = sz-1;i >= 0; --i){
    u = topo[i].second;
    dp[u] = 1;
    for(auto it : st[u].next){
        dp[u] += dp[it.second];
    }
}
cout << dp[0]-1 << endl;

return 0;
}

```

You are given a weighted graph...

```
#include<bits/stdc++.h>

typedef long long ll;
using namespace std;

struct edge
{
    int u;
    int v;
    int w;
};

edge a[100005];
int parent[100005];

bool comp (edge a , edge b)
{
    return a.w>b.w;
}

int find_parent(int u)
{
    if(parent[u]==u)
        return u;
    else
        return parent[u]=find_parent(parent[u]);
}

void merge(int u, int v)
{
    parent[u]=v;
}

int main()
{
    int t;
    cin>>t;
```

```

while(t--) {
    int n,m;
    cin>>n>>m;
    for(int i=1;i<=n;i++)
        parent[i]=i;
    for(int i=0;i<m;i++) {
        cin>>a[i].u>>a[i].v >> a[i].w;
    }
    sort(a,a+m,comp);
    ll ans=0;
    for(int i=0;i<m;i++) {
        int x=find_parent(a[i].u);
        int y=find_parent(a[i].v);
        if(x!=y)
        {
            merge(x,y);
            ans+=a[i].w;
        }
    }
    cout<<ans<<endl;
}
return 0;
cout<<"int printheap(int N)";
}

```

Mancunian and liverbird:

```

#include<bits/stdc++.h>
using namespace std;
int main() {
    int n,i,c;
    scanf("%d %d", &n, &c);
    int tree[n+1][2];

```

```

tree[1][0] = -1;

for(i=2;i<=n;i++) {
    scanf("%d", &tree[i][0]);
}

for(i = 1; i <= n; i++) {
    scanf("%d", &tree[i][1]);
}

int parent;

for(i = 1; i<= n; i++) {
    parent = tree[i][0];
    while(parent != -1 && tree[parent][1] != tree[i][1]) {
        parent = tree[parent][0];
    }
    printf("%d ", parent);
}
return 0;
}

```

A new species is....

```

#include<stdio.h>
#include<stdlib.h>
#include<string.h>

struct cell{
    int name;
    int level;
    int capacity;
};

struct cell queue[1000001];
struct cell arr[1000001];

int front;
int end;

void init_queue(){

```

```
front = 0;
end = 0;
}

void enqueue(int name,int capacity,int level){
queue[end].name = name;
queue[end].level = level;
queue[end].capacity = capacity;
end = end + 1;
}

int is_empty(){
if(end == front)
return 1;
return 0;
}

void dequeue()
{
if(!is_empty())
front++;
}

int main(){
int n,rc;
init_queue();
scanf("%d %d",&n,&rc);
int i,j,k;
for(i=0;i<n;i++){
scanf("%d %d",&arr[i].name,&arr[i].capacity);
}
enqueue(0,rc,0);
i=0;
while(!is_empty()){
int par = queue[front].name;
```

```

int cap = queue[front].capacity;
int lev = queue[front].level+1;
k=1;
for(j=0;j<cap&&i<n;j++,i++){
printf("%d %d %d\n",par,lev,k++);
enqueue(arr[i].name,arr[i].capacity,lev);
}
dequeue();
}
return 0;
}

```

A beautiful code of a tree

```

#include <bits/stdc++.h>
using namespace std;
#define f(i,a,n) for(int i=a;i<n;i++)
#define X(a,b) if(a==b)
vector< int > vi;
const int maxN = 2e5+5;
int N, a[maxN], deg[maxN];
priority_queue<int, vector<int>, greater<int>> q;
int main(){
scanf("%d", &N);
fill(deg+1, deg+N+1, 1);
//for(int i = 0; i < N-2; i++)
f(i,0,N-2){
scanf("%d", &a[i]);
deg[a[i]]++;
}
//for(int i = 1; i <= N; i++)
f(i,1,N+1)

```

```

//if(deg[i] == 1)
X(deg[i],1)
q.push(i);
f(i,0,N-2){
int u = a[i];
int v = q.top(); q.pop();
deg[u]--; deg[v]--;
//if(deg[u] == 1)
X(deg[u],1)
q.push(u);
printf("%d %d\n", v, u);
}
//for(int i = 1; i <= N; i++)
f(i,1,N+1)
if(deg[i])
printf("%d ", i);
}

```

Any sequence of A size n

```

#include<bits/stdc++.h>
#include<map>
using namespace std;
int main() {
int N,i,maximum=INT_MIN;
scanf("%d", &N);
int S[N];
map<int,int> map;
for(i=0;i<N;i++) {
scanf("%d", &S[i]);
maximum=max(maximum,S[i]);
map[S[i]]++;
}

```

```

}

int temp,Q;

cin>>Q;

for(i=0;i<Q;i++) {

scanf("%d", &temp);

if(temp==maximum) printf("%d\n",N);

else {

if(map[temp]>=2) printf("%d\n",N);

else {

map[temp]++;

N++;

printf("%d\n",N);

maximum=max(maximum,temp);

}

}

}

for(auto it=map.begin();it!=map.end();it++) printf("%d ",it->first);

for(auto it=map.rbegin();it!=map.rend();it++) {

if(it->second>1) printf("%d ",it->first);

}

}

```

Given two rooted

```

#include<bits/stdc++.h>

using namespace std;

void dfs(int node,int parent,string &s,vector<vector<int>>&subroot,vector<vector<int>>& v1)

{

//visited[node]=1;

subroot[node][s[node-1]-'a']++;

//intime[node]=t;

//t++;

}

```

```

//z.push_back(node);

for( auto it:v1[node])
{
    if(it!=parent)
    {
        dfs(it,node,s,subroot,v1);

        for(int i=0;i<26;i++)
            subroot[node][i]+=subroot[it][i];
    }
}

//outtime[node]=t;
//t++;
}

int main()
{
    int N,i, Q;
    string S;

    cin >> N >> Q;
    cin >> S;

    vector<vector<int>>v1(N+1);
    for(i=0;i<N-1;i++)
    {
        int u, v;
        cin >> u >> v;
        v1[u].push_back(v);
        v1[v].push_back(u);
    }

    vector<vector<int>>subroot(N+1,vector<int>(26,0));
    dfs(1,0,S,subroot,v1);
    while(Q--)
    {

```

```

int u;
char c;
cin >> u >> c;
cout<<subroot[u][c-'a']<<"\n";
//cout<<cnt<<endl;
}
return 0;
}

```

In a movie festival

```

#include<bits/stdc++.h>
using namespace std;
int dp[1000006][25];
void solve(){}
int main(){
    solve();
    int n, q; cin>>n>>q;
    for (int i = 0; i < n; i++) {
        int x, y; cin>>x>>y;
        dp[y][0] = max(dp[y][0], x);
    }
    for (int i = 1; i <= 1000000; i++)
        dp[i][0] = max(dp[i][0], dp[i-1][0]);
    for (int k = 1; k <= 20; k++)
        for (int i = 1; i <= 1000000; i++)
            dp[i][k] = dp[dp[i][k-1]][k-1];
    while(q--) {
        int x,y; cin>>x>>y;
        int ans = 0;

```

```

while(y>0) {
    int z = 0;
    for (int i = 0; i <= 20; i++) {
        if (dp[y][i] < x) {
            z = i;
            break;
        }
    }
    if (z == 0)
        break;
    ans += (1<<(z-1));
    y = dp[y][z-1];
}
cout<<ans<<endl;
}

}

```

YOU ARE GIVEN A K-ARY...

```

#include <iostream>
#include <map>
#include <assert.h>
using namespace std;
#define int long long
map < pair < int, int >, int > adj;
int find_depth( int u, int k ) {
    int depth = 0;
    while ( u > 0 ) {
        u = u / k;

```

```

depth = depth + 1;
}

return depth - 1;
}

int dist( int u, int v, int k ) {

int dist = 0;

int depth_u = find_depth( u, k );
int depth_v = find_depth( v, k );

if ( depth_u < depth_v ) {

swap ( u, v );

swap ( depth_u, depth_v );

}

while( depth_u != depth_v ) {

if ( adj.count( { u, u / k } ) ) {

dist = dist + adj[ { u, u / k } ];

} else {

dist = dist + 1;

}

depth_u = depth_u - 1;

u = u / k;

}

while ( u != v ) {

if ( adj.count( { u, u / k } ) ) {

dist = dist + adj [ { u, u / k } ];

} else {

dist = dist + 1;

}

if ( adj.count( { v, v / k } ) ) {

dist = dist + adj [ { v, v / k } ];

} else {

dist = dist + 1;

}
}
}
}

```

```

}

u = u / k;

v = v / k;

}

return dist;
}

void add_weight( int vertex, int parent, int w ) {

if ( !adj.count ( { vertex, parent } ) ) {

adj[ { vertex, parent } ] = 1;

}

adj[ { vertex, parent } ] = adj[ { vertex, parent } ] + w;

}

void increase_weights ( int u, int v, int w, int k ) {

int depth_u = find_depth( u, k );

int depth_v = find_depth( v, k );

if ( depth_u < depth_v ) {

swap ( u, v );

swap ( depth_u, depth_v );

}

while( depth_u != depth_v ) {

add_weight( u, u / k, w );

depth_u = depth_u - 1;

u = u / k;

}

while ( u != v ) {

add_weight( u, u / k, w );

add_weight( v, v / k, w );

u = u / k;

v = v / k;

}
}
}

```

```

signed main() {
    int k, q, x, u, v, w;
    cin >> k >> q;
    while(q--) {
        cin >> x;
        if ( x == 1 ) {
            cin >> u >> v;
            cout << dist( u, v, k ) << "\n";
        } else {
            cin >> u >> v >> w;
            increase_weights( u, v, w, k );
        }
    }
}

```

ANY SEQUENCE A OF SIZE..

```

#include<bits/stdc++.h>
#include<map>
using namespace std;
int main() {
    int N,i,maximum=INT_MIN;
    scanf("%d", &N);
    int S[N];
    map<int,int> map;
    for(i=0;i<N;i++) {
        scanf("%d", &S[i]);
        maximum=max(maximum,S[i]);
        map[S[i]]++;
    }
}

```

```
int temp,Q;  
cin>>Q;  
for(i=0;i<Q;i++) {  
    scanf("%d", &temp);  
    if(temp==maximum) printf("%d\n",N);  
    else {  
        if(map[temp]>=2) printf("%d\n",N);  
        else {  
            map[temp]++;  
            N++;  
            printf("%d\n",N);  
            maximum=max(maximum,temp);  
        }  
    }  
}  
  
for(auto it=map.begin();it!=map.end();it++) printf("%d ",it->first);  
for(auto it=map.rbegin();it!=map.rend();it++) {  
    if(it->second>1) printf("%d ",it->first);  
}  
}
```

Elab sorting

SORT14

QUESTION DESCRIPTION

Mommy is a very active lady. She likes to keep all stuff sorted. She has developed an interesting technique of sorting stuff over the years. She goes through the items repeatedly from first to last and whenever she finds two consecutive items unsorted, she puts them in the proper order. She continues the process until all the items are sorted.

One day Mommy has to attend a wedding ceremony. Suddenly she remembers that she has not sorted the plates after washing. She has only M minutes left. If she can complete the task within the remaining time, she will sort her plates and then attend the wedding. However if she cannot, she decides to skip the task.

She knows that she takes S seconds per swap. However she does not know the total number of swaps required and hence she is in trouble. She wants you to help her out.

Mandatory expression is " $x=(s*c)/60$ "

```
#include<stdio.h>
int main()
{
int a[1001];
int n,i,j,t,swap,temp,k,m,ss,r;
scanf("%d", &t);
if(t==1)
{
printf("1");
goto a;
}
while(t--)
{
swap=0;
scanf("%d %d %d", &m,&ss,&n);
for(j=0; j<a[j+1];j++)
{
temp=a[j];
a[j]=a[j+1];
a[j+1]=temp;
swap++;
}
```

```
}

int x,s,c;
x=(s*c)/60;
r=swap*ss;
if(r<=m*60)
printf("1\n0\n0");
else
printf("0");
printf("\n");
return 0;
a:
{
return 0;
}
}
```

SORT8

QUESTION DESCRIPTION

Given an array with all elements greater than or equal to zero.Return the maximum product of two numbers possible.

Input:

The first line of input contains an integer T denoting the number of test cases.

The first line of each test case is N, N is size of array.

The second line of each test case contains N input A[i].

Mandatory method for this program is "void sort(int a[],int n)"

Output:

Print the maximum product of two numbers possible.

Constraints:

$1 \leq T \leq 20$

$1 \leq N \leq 50$

$0 \leq A[i] \leq 1000$

```
#include<stdio.h>
```

```

void sort(int a[],int n);
int main()
{
    int arr[30], i, x, t;
    scanf("%d",&t);
    while(t--)
    {
        scanf("%d",&x);
        for(i=0;i<x;i++)
        {
            scanf("%d",&arr[i]);
        }
        sort(arr, x);
    }
    return 0;
}
void sort(int a[],int n)
{
    int i, j, p=0;
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
        {
            if(i!=j)
            {
                if(p<(a[i]*a[j]))
                {
                    p=a[i]*a[j];
                }
            }
        }
    }
    printf("%d\n",p);
}

```

SORT4

QUESTION DESCRIPTION

Ramu and Somu both are decided to play a game to Sort the given set of numbers using Insertion Sort.

So he got The first line of the input contains the number of elements, the second line of the input contains the numbers to be sorted.

In the output print the status of the array at the 3rd iteration and the final sorted array in the given format.

Somu will evaluate the result whether its correct or not Mandatory declaration need to be follow as "
void InSort(int arr[], int n)"

```
#include<stdio.h>

void printArray(int arr[], int n)
{
    int i;
    printf("Sorted Array:");
    for(i=0; i<n; i++)
    {
        printf("%d ",arr[i]);
    }
    printf("\n");
}

void InSort(int arr[], int n)
{
    int step, i;
    for(step=1; step<n; step++)
    {
        int key = arr[step];
        int j=step-1;
        while(key<arr[j] && j>=0)
        {
            arr[j+1] = arr[j];
            --j;
        }
        arr[j+1]=key;
        if(step==2)
        {
            for(i=0;i<n;i++)
                printf("%d ",arr[i]);
        }
    }
    printf("\n");
}

int main()
{
    int data[30], i, n;
```

```
scanf("%d",&n);
for(i=0;i<n;i++)
{
    scanf("%d",&data[i]);
}
InSort(data, n);
printArray(data, n);
return 0;
}
```

SORT6

QUESTION DESCRIPTION

MS .Dhoni want to play a game when the players are free in the dressing room.

Dhoni have given an array of n distinct elements, the task is to find all elements in array which have at-least two greater elements than themselves.

Dhoni declared a mandatory conditions like "void sort(int a[],int n)"

Examples:

Input : A[] = {2, 8, 7, 1, 5};
Output : 1 2 5

The output three elements have two or more greater elements

Input : A[] = {7, -2, 3, 4, 9, -1};
Output : -2 -1 3 4

Input:

The first line of input contains an integer T denoting the no of test cases.

Each test case contains two lines .

The first line of input contains an integer n denoting the size of the array.

Then in the next are n space separated values of the array.

```
#include <stdio.h>
```

```
int main()
{
    int t,i,n,j,k,arr[30],temp;
    scanf("%d",&t);
    for(i=0;i<t;i++)
    {
        scanf("%d",&n);
        for(j=0;j<n;j++)
        {
            scanf("%d",&arr[j]);
        }
        for(j=0;j<n;j++)
        {
            for(k=j+1;k<n;k++)
            {
                if(arr[j]>arr[k])
                {
                    temp=arr[j];
                    arr[j]=arr[k];
                    arr[k]=temp;
                }
            }
        }
    }
}
```

```
    }  
}  
  
}  
  
for(j=0;j<n-2;j++)  
{printf("%d ",arr[j]);}  
  
printf("\n");  
  
}  
  
return 0;  
}
```

SORT13

QUESTION DESCRIPTION

You have to merge the two sorted arrays into one sorted array (in non-increasing order)

Input:

First line contains an integer T, denoting the number of test cases.

First line of each test case contains two space separated integers X and Y, denoting the size of the two sorted arrays.

Second line of each test case contains X space separated integers, denoting the first sorted array P.

Third line of each test case contains Y space separated integers, denoting the second array Q.

Output:

For each test case, print (X + Y) space separated integer representing the merged array.

```
#include <stdio.h>

int main() {
    int t,i,a,b,arr[20],j,temp,k;
    scanf("%d",&t);
    for(i=0;i<t;i++)
    {
        scanf("%d%d",&a,&b);
        for(j=0;j<(a+b);j++)
        {
            scanf("%d",&arr[j]);
        }
        for(j=0;j<(a+b);j++)
        {
            for(k=j+1;k<(a+b);k++)
            {
                if(arr[j]<arr[k])
                {
                    temp=arr[j];
                    arr[j]=arr[k];
                    arr[k]=temp;
                }
            }
        }
    }
}
```

```

    }

}

for(j=0;j<(a+b);j++)

{printf("%d ",arr[j]);}

printf("\n");

}

return 0;
}

```

SORT9

QUESTION DESCRIPTION

Given an array of integers and two numbers k1 and k2. Find sum of all elements between given two k1'th and k2'th smallest elements of array. It may be assumed that ($1 \leq k1 < k2 \leq n$) and all elements of array are distinct.

Input:

The first line of input contains an integer T denoting the no of test cases. Then T test cases follow. Each test case contains an integer N, denoting the length of the array. Next line contains N space seperated integers of the array. Third line contains two space seperated integers denoting k1'th and k2'th smallest elements.

Output:

For each test case in a new line output the sum of all the elements between k1'th and k2'th smallest elements.

Constraints:

$1 \leq T \leq 100$

$1 \leq k1 < k2 \leq N \leq 50$

```
#include <stdio.h>
```

```
void sortarr(int n,int *a);
```

```
int main()
```

```
{  
  
int t,i,n,j,a[30],p,q,sum=0;  
  
scanf("%d",&t);  
  
for(i=0;i<t;i++)  
  
{  
  
scanf("%d",&n);  
  
for(j=0;j<n;j++)  
  
{  
  
scanf("%d",&a[j]);  
  
}  
  
/*for(j=0;j<n;j++)  
  
printf("%d ",a[j]);  
  
printf("\n");*/  
  
  
  
sortarr(n,a);  
  
  
  
/* for(j=0;j<n;j++)  
  
printf("%d ",a[j]);  
  
printf("\n");*/  
  
  
  
scanf("%d%d",&p,&q);  
  
for(j=(p);j<(q-1);j++)
```

```
{  
    sum=sum+a[j];  
}  
  
printf("%d\n",sum);sum=0;  
}  
  
return 0;  
}  
  
void sortarr(int n,int *a)  
{  
    int j,k,t;  
    for(j=0;j<n;j++)  
    {  
        for(k=j+1;k<(n);k++)  
        {  
            if(a[j]>a[k])  
            {  
                t=a[j];  
                a[j]=a[k];  
                a[k]=t;  
            }  
        }  
    }  
}
```

```
    }  
}  
}
```

SORT3

QUESTION DESCRIPTION

Sort the given set of numbers using Bubble Sort.

The first line of the input contains the number of elements, the second line of the input contains the numbers to be sorted.

In the output print the status of the array at the 3rd iteration and the final sorted array in the given format.

Mandatory declaration for function is "void printArr(int arr[], int size)"

```
#include <stdio.h>
```

```
void swap(int *xp, int *yp)  
{  
int temp = *xp;  
*xp = *yp;  
*yp = temp;  
}
```

```
void bubbleSort(int arr[], int n)  
{  
int i, j; int a, b;  
for (i = 0; i < n-1; i++)  
{  
    a=i;  
    if(a==3){for (a=0; a< n; a++)  
printf("%d ", arr[a]);  
printf("\n"); }  
  
    for (j = 0; j < n-i-1; j++)  
if (arr[j] > arr[j+1])  
swap(&arr[j], &arr[j+1]);
```

```
}
```

```
}
```

```
void printArray(int arr[], int size)
```

```
{
```

```
int i;
```

```
for (i=0; i < size; i++)
```

```
printf("%d ", arr[i]);
```

```
printf("\n");
```

```
}
```

```
int main()
```

```
{
```

```
int t;scanf("%d",&t);
```

```
int i;int arr[t];
```

```
for(i=0;i<t;i++)
```

```
{ scanf("%d",&arr[i]);}
```

```
int n = sizeof(arr)/sizeof(arr[0]);
```

```
bubbleSort(arr, n);
```

```
printf("Sorted array:");
```

```
printArray(arr, n);
```

```
return 0;
```

```
}
```

SORT1

QUESTION DESCRIPTION

You are given an array A of size N, and Q queries to deal with.

For each query, you are given an integer X, and you're supposed to find out if X is present in the array A or not.

Mandatory method name is "void quicksort(int x[10],int first,int last)"

Input:

The first line contains two integers, N and Q, denoting the size of array A and number of queries.

The second line contains N space separated integers, denoting the array of elements Ai.

The next Q lines contain a single integer X per line.

Output:

For each query, print YES if the X is in the array, otherwise print NO.

```
#include <stdio.h>

int main() {

    int n,q,arr1[20],arr2[20],j,f,i;

    scanf("%d %d",&n,&q);

    for(i=0;i<n;i++)

    {

        scanf("%d",&arr1[i]);

    }

    for(j=0;j<q;j++)

    {

        scanf("%d",&arr2[j]);

    }

    for(j=0;j<q;j++)

    {

        for(i=0;i<n;i++)

        {

            if(arr2[j]==arr1[i])

        }

    }

}
```

```
f=1;  
  
break;  
  
}  
  
else  
  
f=0;  
  
}  
  
if(f==1)  
  
printf("YES\\n");  
  
else  
  
printf("NO\\n");  
  
}  
  
return 0;  
  
}
```

SORT11

QUESTION DESCRIPTION

In a candy store there are N different types of candies available and the prices of all the N different types of candies are provided to you.

You are now provided with an attractive offer.

You can buy a single candy from the store and get atmost K other candies (all are different types) for free

Now you have to answer two questions.

Firstly, you have to tell what is the minimum amount of money you have to spend to buy all the N different candies.

Secondly, you have to tell what is the maximum amount of money you have to spend to buy all the N different candies.

In both the cases you must utilize the offer i.e. you buy one candy and get K other candies for free.

Mandatory conditions are "static void mergeSort(int a[],int l,int r)"

```
#include <stdio.h>
static void mergeSort(int a[],int l,int r)
{};
}
int main()
{
    int t;
    scanf("%d",&t);
    while(t--)
    {int i,j,k,l=0,n,m,a[1000],s=0,s1=0,min=0,max=0;
    scanf("%d %d",&n,&k);
    for(i=0;i<n;i++)
    scanf("%d",&a[i]);
    for(i=0;i<n;i++)
    {for(j=0;j<n-i-1;j++)
    {int tm;
    if(a[j]>a[j+1])
    {tm=a[j];
    a[j]=a[j+1];
    a[j+1]=tm;}}
    while(s<n)
    {min=min+a[l];
    l++;
    s=s+k+1;}
    j=n-1;
    while(s1<n)
    {max=max+a[j];
    j--;
    s1=s1+k+1;}
    printf("%d ",min);
    printf("%d\n",max);}
    return 0;
}
```

SORT7

QUESTION DESCRIPTION

Given two arrays, A and B, of equal size n, the task is to find the minimum value of $A[0] * B[0] + A[1] * B[1] + \dots + A[n-1] * B[n-1]$, where shuffling of elements of arrays A and B is allowed.

Mandatory conditions are "void result(int a[],int b[],int n)"

Examples:

Input : A[] = {3, 1, 1} and B[] = {6, 5, 4}

Output : 23 Minimum value of S = $1*6 + 1*5 + 3*4 = 23$.

Input : A[] = { 6, 1, 9, 5, 4 } and B[] = { 3, 4, 8, 2, 4 }

Output : 80. Minimum value of S = $1*8 + 4*4 + 5*4 + 6*3 + 9*2 = 80$.

Input:

The first line of input contains an integer denoting the no of test cases.

Then T test cases follow. Each test case contains three lines.

The first line of input contains an integer N denoting the size of the arrays.

```
#include <stdio.h>
void result(int a[],int b[],int n);
int main()
{
    int t, i, j, arr1[30], arr2[30], x, temp;
    scanf("%d",&t);
    while(t--)
    {
        scanf("%d",&x);
        for(i=0;i<x;i++)
        {
            scanf("%d",&arr1[i]);
        }

        for(i=0;i<x;++i)
        {
            for (j=i+1;j<x;++j)
            {
                if (arr1[i] > arr1[j])
                {
                    temp = arr1[i];
                    arr1[i] = arr1[j];
                    arr1[j] = temp;
                }
            }
        }
    }
}
```

```

        arr1[i] = arr1[j];
        arr1[j] = temp;
    }
}
}

for(i=0;i<x;i++)
{
    scanf("%d",&arr2[i]);
}

for(i=0;i<x;++i)
{
    for (j=i+1;j<x;++j)
    {
        if (arr2[i] < arr2[j])
        {
            temp = arr2[i];
            arr2[i] = arr2[j];
            arr2[j] = temp;
        }
    }
}

result(arr1, arr2, x);
}
return 0;
}

void result(int a[],int b[],int n)
{
    int i, fin=0, sum;
    for(i=0;i<n;i++)
    {
        fin+=a[i]*b[i];
        sum=fin;
    }
    printf("%d\n",sum);
}

```

Sort 15

QUESTION DESCRIPTION

Given an array of distinct positive numbers, the task is to calculate the minimum number of subsets (or subsequences) from the array such that each subset contains consecutive numbers.

Mandatory variables are "int n and int t"

Input:

The first line of input contains an integer T denoting the no of test cases.

Then T test cases follow. Each test case contains an integer N, denoting the length of the array.

Next line contains N space seperated integers of the array.

Output:

For each test case output a new line denoting count of number of such subset's that contains consecutive numbers.

Constraints:

1<=T<=100

1<=N<=50

```
#include <stdio.h>
```

```
void arrange_arr(int n,int *arr);
```

```
int main()
```

{

int t;

int n;

int a[30],i,j,c=0,k,p;

scanf("%d",&t);

for(i=0;i<t;i++)

{

scanf("%d",&n);

for(j=0;j<n;j++)

{

scanf("%d",&a[j]);

}

arrange_arr(n,a);

p=0;

// for(j=0;j<n;j++)

```
//{printf("%d ",a[j]);}
```

```
for(j=0,c=0;j<n-1;j++)
```

```
{
```

```
if((a[j+1]-a[j]==1)
```

```
{p++;}
```

```
else
```

```
{
```

```
if(p>0)
```

```
{c++;p=0;}
```

```
}
```

```
}
```

```
if(c>0)
```

```
c=c+1;
```

```
printf("%d\n",c);
```

```
}
```

```
return 0;
```

```
}
```

```
void arrange_arr(int n,int *arr)
```

```
{
```

```
int j,k,t;
```

```
for(j=0;j<n;j++)
```

```
{
```

```
    for(k=j+1;k<n;k++)
```

```
{
```

```
        if(arr[j]>arr[k])
```

```
{
```

```
            t=arr[j];
```

```
            arr[j]=arr[k];
```

```
    arr[k]=t;  
}  
}  
}
```

Sort 5

QUESTION DESCRIPTION

Kalaiselvan is going to act as a car driver and he has to drive a car on a track divided into "N" no. of sub-tracks.

You are also given the value of "K" i.e. the total kilometers a car can drive on each sub-track.

If the car can't cover a sub-track, you can add any unit of Petrol in it. With each unit of petrol added, the total kilometers your car can travel will increase by one unit .

Kalai selvan need to declar the mandatory function name as " void sort(int a[],int n,int k)"

Input:

The first line of input contains an integer T denoting the no of test cases.

Then T test cases follow. Each test case contains two space separated integers N and K.

The second line of each test case contains N space separated integers (A[]) denoting the distance of each N sub-tracks.

Output:

For each test case in a new line you have to print out the minimum unit of Petrol your car require to cover all the sub-tracks. If no extra unit of petrol is required, print -1.

```
#include <stdio.h>

void sort(int a[],int n,int k){}

int main()

{

    int t,n,p,i,arr[20],temp,k,j;

    scanf("%d",&t);

    for(i=0;i<t;i++)

    {

        scanf("%d%d",&n,&p);

        for(j=0;j<n;j++)

        {

            scanf("%d",&arr[j]);

        }

        for(j=0;j<n;j++)

        {

            for(k=j+1;k<n;k++)

            {

                if(arr[j]<arr[k])
```

```
{  
    temp=arr[j];  
    arr[j]=arr[k];  
    arr[k]=temp;  
}  
}  
  
if(arr[0]>p)  
{  
    printf("%d",arr[0]-p);  
}  
else  
    printf("-1");  
  
printf("\n");  
}  
  
return 0;  
}
```


question

<p>Question Description:
Simon has given N ratios in the form of A and B that is represented as A/B. The values of A and B are represented as double data type values. The values of B are incorrect. The actual values of B are B+R. Simon know the actual sum of all the ratios that is available in variable K.

Note: The true values of B, represented as (B+R), are always greater than 0. Simon's task is to determine the value of R.

Constraints:
1 <= N <= 1000
1 <= A <= 1000
|B| <= 1000
1 <= K <= 10^6

Input Format:
First line: Two integers N and col denoting the number of ratios and the value 2 respectively
Next N lines: Each line contains two double values A and B
Last line: A double value K denoting the sum of all the ratios

Output Format:
Print the value of R. Simon's answer must contain an absolute or relative error of less than 10^-6.</p><p>&nbsp</p>

answer

```
#include<iostream>

using namespace std;

double func(double arr[][2],double r,int n){

    double ans = 0;

    for (int i = 0; i < n; i++) {
        ans+= (arr[i][0]/(arr[i][1]+r));
    }

    return ans;
}

int main(){

    int n,two;

    cin>>n>>two;

    double arr[n][2];

    for (int i = 0; i < n; i++) {

        cin>>arr[i][0]>>arr[i][1];
    }
}
```

```

double hi=2000,lo=0,mid,curr,k;
cin>>k;
while(hi-lo>1e-7){
    mid=(hi+lo)/2;
    curr=func(arr,mid,n);
    if(curr<k){
        hi = mid;
    }
    else{
        lo = mid + 1e-7;
    }
}
printf("%.6f",mid);

return 0;
printf("double solve(double** arr,double K,int n)");
}

```

question

<p>Problem Description:
John Krasinski among his friends wants to go to watch a movie in Sathyam Cinemas.
There is something special about Sathyam cinemas whenever people come in the group here. </p><p>They will get seats accordingly their heights. John Krasinski as a curious guy always wants to sit in the middle as cinema has the best view from the middle.
Now, John Krasinski as the leader of his group decides who will join him for the movie.
Initially, he has N-1 friends with him (N including him).
You are given N-1 numbers that represent the heights of John Krasinski's friends.
You are given the height of John Krasinski as well.

Now, John Krasinski can do two operations:
1. He can call a new friend of height H.
2. He can cancel any of his friend invitations.

Each operation will cost him a unit time.
He wants to do this as soon as possible.

Constraints:
1 <= T <= 100
1 <= N <= 10^5
1 <= Ar[i] <= 10^9

Input Format:

The first line contains T, where T is the test case.
Each test case contain two lines,
The first line contains two space-separated integer N, S where N is the total number of John Krasinski's friend and 'S' is John Krasinski height.
The second line contains N space-separated integers that represent the height of John Krasinski's friend.

Output Format:
Print the required answer (cost) for each test case in a new line</p><p>Explanation:</p><p>Sample Input</p><p>2

```

</p><p>3 2
</p><p>4 3 1
</p><p>1 5
</p><p>6
</p><p><strong>Sample Output</strong></p><p>1
</p><p>1
</p><p>In first test case :<br>We can cancel invitation of person of height 4 (Cost = 1)<br>In second
Test Case:<br>We can invite person with height 4 (Cost =1)</p>

```

answer

```

#include <bits/stdc++.h>
using namespace std;
int main()
{
int test;cin>>test;while(test--){
    int n,s;cin>>n>>s;
    std::vector<int> v(n);
    for (int i = 0; i < n; i++) {
        cin>>v[i];
    }
    //v[n] = s;
    sort(v.begin(),v.end());
    for (int i = 0; i < n; i++) {
        if(v[i]>=s){
            s = i;break;
        }
    }
    cout<<abs(s-(n-s))<<endl;
}
return 0;
}

```

question

<p>Problem Description:</p><p>In mathematics, a permutation of a set is, loosely speaking, an arrangement of its members into a sequence or linear order, or if the set is already ordered, a rearrangement of its elements. The word "permutation" also refers to the act or process of changing the linear order of an ordered set.</p><p>
Mariappan(M) is alone too and has a permutation p1,p2,...pn of numbers from 1 to n.

M thinks that a permutation p1,p2,...pn beautifulness is defined as value of $\sum |p_i - i|$, $1 \leq i \leq n$.

M can swap two elements of the permutation at most once.

Constraints:
 $1 \leq n \leq 10^5$
 $1 \leq p_i \leq n$ all p_i are distinct

Input Format:

First line contains only 'n'.
Second line contains the permutation p1, p2...pn separated by space.

Output Format:
Print the output in a single line contains maximum beautifulness that M can get</p>

answer

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int n;cin>>n;
    vector<int> v(n);
    for (int i = 0; i < n; i++) {
        cin>>v[i];
    }
    sort(v.begin(),v.end(),greater<int>());
    int tot= 0;
    for(int i=0 ; i<n ; i++){
        //cout<<v[i]<<' ';
        tot+= abs(v[i]-1-i);
    }
    cout<<tot;
```

```
    return 0;  
  
    printf("swap(l,r);");  
  
}
```

question

<p>Problem Description:
Tina has given a boolean matrix mat[P][Q] of size P X Q to Laaysa, </p><p>She wants to modify it such that if a matrix cell mat[m][n] is 1 (or true) then make all the cells of m_th row and nth column as 1. Can you help Laaysa?

Constraints:
1 <= p, q <= 1000

Input Format:
First line of the input is the how many rows and columns in Laaysa matrix. </p><p>After that, each line will represent each row and Laaysa need to enter numbers.

Output Format:
Print the resultant matrix output in a separate lines.</p>

answer

```
#include <bits/stdc++.h>  
  
using namespace std;  
  
int main()  
{  
    int r,c;  
  
    cin>>r>>c;  
  
    int arr[r][c];  
  
    int arrTemp[r][c];  
  
    for (int i = 0; i < r; i++) {  
  
        for (int j = 0; j < c; j++) {  
  
            cin>>arr[i][j];  
  
            arrTemp[i][j] = 0;  
  
        }  
  
    }  
  
    for (int i = 0; i < r; i++) {  
  
        for (int j = 0; j < c; j++) {  
  
    }
```

```

if(arr[i][j]==1){

    for(int i1 = 0;i1<r;i1++){

        arrTemp[i1][j] =1;

    }

    for(int i1 = 0;i1<c;i1++){

        arrTemp[i][i1] =1;

    }

}

for (int i = 0; i < r; i++) {

    for (int j = 0; j < c; j++) {

        cout<<arrTemp[i][j];

        if(j!=c-1)cout<<' ';

    }

    cout<<endl;

}

return 0;

printf("for(m=0;m<r;m++)");
}

```

question

<p>Problem Description:
Public school have arranged an Annual Day Function.</p><p>Volunteers have decorated a floor on various places of the school using Rose and Tulip flowers. </p><p>But one of the coordinators requested the volunteers to rearrange the decoration like a triangular size.</p><p>Coordinator also told them that tulips flowers need to be positioned at the middle of the roses</p><p>School has 20 buildings and as per Principal order the numbers of rows in the decoration should also match the building number.
The Principal of the school is interested in seeing the final decoration but he is quite busy with the other works.</p><p>So he likes to see how the final decoration have come through online mode if he gives the building number.</p><p>So can you display him the final decoration

layout?
Note:
Roses are represented by 1.</p><p>Tulips are represented by 0.

Constraints:
1 ≤ rows ≤ 20</p><p>
Input Format:
Only line of input has single integer representing the building number.</p><p>
Output Format:
Print the final layout of the decoration.</p><p>Refer sample testcases for format specification.</p>

answer

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int n;cin>>n;cout<<"1 \n";
    for (int i = 0; i < n-2; i++) {
        cout<<"1 ";
        for (int j = 0; j < i; j++) {
            cout<<"0 ";
        }
        cout<<"1 \n";
    }
    for (int i = 0; i < n; i++) {
        cout<<"1 ";
    }
    return 0;
    cout<<"for(i=1;i<=rows;i++)";
}
```

question

<p>Problem Description:
VIBGYOR isn't just an acronym, it's a way of life for Asian paint company. The owner is considering modernizing his paint mixing equipment with a computerized model. He's hired you to code the prototype. Your simple program will need to correctly output the right color based on the blends he's given you.</p><p>Example Colors</p><p>Primary colors “ RED, BLUE, YELLOW”, </p><p>secondary Colors “ORANGE, PURPLE, GREEN”</p><p>Tertiary Colors “ LIGHT RED, DARK RED, LIGHT PURPLE, DARK PURPLE, LIGHT BLUE, DARK BLUE, LIGHT GREEN, DARK GREEN, LIGHT YELLOW, DARK YELLOW, LIGHT ORANGE, DARK

ORANGE"</p><p>
Input Format:
You will receive one to five lines of color combinations consisting of primary colors and secondary colors as well as black and white to make "dark" and "light" colors. The full science of colorisation and pigments will be implemented next, if your prototype is successful.

Output Format:
Print the output in a separate lines contains, Your program should output the correct color depending on what two colors were "mixed" on the line. Primary colors should mix together to create secondary colors. Anything mixed with "WHITE" or "BLACK" should be output as either "LIGHT X" or "DARK X" where X is the color "WHITE" or "BLACK" were mixed with. Anything mixed with itself won't change colors. You are guaranteed not to receive incompatible colors, or colors not listed in the color wheels shown above (aside from "WHITE" and "BLACK").</p><p> </p><p>Refer logical test cases for your reference. </p>

answer

```
#include <stdio.h>
#include<bits/stdc++.h>
using namespace std;
void arr()
{
    return;
}
int main()
{
    string ss[] = {"RED", "BLUE", "PURPLE", "YELLOW", "ORANGE" "GREEN"};
    string s,s1;
    int t = 4;
    while(t--)
    {
        cin>>s>>s1;
        //cout<<s<<" "<<s1;
        if(s == ss[0] && s1 == ss[3])
            cout<<"ORANGE";
        else if(s == ss[1] && s1 == ss[3]) cout<<"GREEN";
        else if(s == ss[1] && s1== ss[0]) cout<<"PURPLE";
        else if(s == "BLACK") cout<<"DARK"<<" "<<s1;
        else if(s1 == "BLACK") cout<<"DARK"<<" "<<s;
```

```

else if(s1 == "WHITE") cout<<"LIGHT"<<" "<<s;
else if(s == "WHITE") cout<<"LIGHT"<<" "<<s1;
else if(s1 == s)cout<<s;
else cout<<"N/A";
cout<<"\n";
}

return 0;

cout<<"if(strcmp(c,colors[i])==0 for(i=0;i<8;i++) char mixes[8][8][32] char colors[8][32];"

```

question

<p>Problem Description:</p><p>Umesh has n mixtures in front of him, arranged in a row. Each mixture has one of 100 different colors (colors have numbers from 0 to 99).</p><p>He wants to mix all these mixtures together. At each step, he is going to take two mixtures that stand next to each other and mix them together, and put the resulting mixture in their place.</p><p>Functional Description:</p><p>When mixing two mixtures of colors a and b, the resulting mixture will have the color $(a+b) \bmod 100$.</p><p>Also, there will be some smoke in the process. The amount of smoke generated when mixing two mixtures of colors a and b is $a * b$.</p><p>Find out what is the minimum amount of smoke that Umesh can get when mixing all the mixtures together.</p><p>Constraints:</p><p>1 <= n <= 100</p><p>Input Format:</p><p>There will be a number of test cases in the input.</p><p>The first line of each test case will contain n, the number of mixtures, </p><p>The second line will contain n integers representing the initial colors of the mixtures.</p><p>Output Format:</p><p>For each test case, output the minimum amount of smoke.</p>

answer

```
#include<stdio.h>
```

```
typedef long long unsigned LLU;
```

```
LLU min_smoke[100][100];
int color[100][100];
```

```
LLU smoke(int n){
```

```
    int i,j,l;
```

```

for(i=0;i<n;i++){
    for(j=0;j<n;j++){
        min_smoke[i][j]=10000000000000000000;
    }
}

for(i=0;i<n;i++){
    min_smoke[i][i] = 0;
}

for(l=2;l<=n;l++){
    int e = n-l;
    for(i=0;i<=e;i++){
        int k = i+l-1;
        for(j=i;j<k;j++){
            LLU sm = min_smoke[i][j] + min_smoke[j+1][k] + color[i][j]*color[j+1][k];
            int cl = (color[i][j]+color[j+1][k])%100;
            if(sm<min_smoke[i][k]){
                min_smoke[i][k] = sm;
                color[i][k] = cl;
            }
        }
    }
}

return min_smoke[0][n-1];
}

int main(void){

```

```

int n;
while(scanf("%d",&n)!=EOF){
    int i;
    for(i=0;i<n;i++){
        scanf("%d",&(color[i][i]));
    }
    printf("%llu\n",smoke(n));
}
return 0;
printf("scount[100][100]colours[100]");
}

```

question

<p>Problem Description:
For some reason, your school's football team has chosen to spell out the numbers on their jerseys instead of using the usual digits. Being great fans, you're going to be ready to cheer for your favorite players by bringing letter cards so you can spell out their number. Each fan has different favorites, so they each need to bring different sets of letters.

The English spellings for the numbers 0 to 12 are:
ZERO ONE TWO THREE FOUR FIVE SIX
SEVEN EIGHT NINE TEN ELEVEN TWELVE

Input Format:
Read a set of integers from 0 to 12, separated by spaces, representing one fan's favorite players. The last integer will be 999, marking the end of the line.

Output Format:
Print the same numbers, then a period and a space. Then, in alphabetical order, print all the letters the fan needs to be able to spell any one of the jersey numbers provided</p>

answer

```

#include <bits/stdc++.h>
using namespace std;
int main()
{
    int curr;
    multiset<char> mp;
    string names[] =
    {"","ONE","TWO","THREE","FOUR","FIVE","SIX","SEVEN","EIGHT","NINE","TEN","ELEVEN","TWELVE"};
};

```

```

while(cin>>curr){

    if(curr==999){

        cout<<"0999"<<'.'<<' ';
        break;
    }

    cout<<curr<<' ';

    if(curr>12)continue;

    string now = names[curr];

    for(auto ch:now){

        mp.insert(ch);

    }

}

for (auto ch : mp) {

    cout<<ch<<' ';

}

return 0;

printf("char nums[13][256]for(n=0;n<26;n++)");

}

```

question

<p>Question description</p><p>Sathya is a IT expert who training youngsters struggling in coding to make them better.</p><p>Sathya usually gives interesting problems to the youngsters to make them love the coding.

One such day Sathya provided the youngsters to solve that the given an array of integers and the numbers k1 and k2, get the sum of the two numbers. </p><p>Find the sum of all elements in the array between the k1st and k2nd smallest elements. </p><p>It is reasonable to suppose that (1= k1 k2 n) and all array items are distinct.</p><p>Constraints:</p><p>1<= T <= 100</p><p>1<= k1< k2 <= N <= 50</p><p>Input Format:</p><p>The first line of input contains an integer T denoting the no of test cases. Then T test cases follow. Each test case contains an integer N, denoting the length of the array. </p><p>Next line contains N space separated integers of the array. </p><p>Third line contains two space separated integers denoting k1'th and k2'th smallest elements.</p><p>Output Format:</p><p>For each test case in a new line output the sum of all the elements between k1'th and k2'th smallest elements.</p><p> </p><p> </p>

answer

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int t;cin>>t;
    while(t>0){
        int n,k1,k2,ans=0;
        cin>>n;
        int arr[n];
        for(int i=0;i<n;i++) {
            cin>>arr[i];
        }
        cin>>k1>>k2;
        sort(arr,arr+n);
        for (int i = k1; i < k2-1; i++) {
            ans+=arr[i];
        }
        cout<<ans<<endl;
        t--;
    }
    return 0;
    printf("for(int i=0;i<n-1;i++)");
}
```

question

<p>Question description</p><p>In India, the real estate sector is the second-highest employment generator, after the agriculture sector. </p><p>It is also expected that this sector will incur more non-resident Indian (NRI) investment, both in the short term and the long term. </p><p>Bengaluru is expected to be the most favoured property investment

destination for NRIs, followed by Ahmedabad, Pune, Chennai, Goa, Delhi and Dehradun.</p><p>Ramesh is residing in England. he is willing to invest money in real estate. </p><p>So he has chosen Bengaluru for good investment.
There are N flats for sale in Bengaluru main city. </p><p>The i-th flat costs Ai rupees to buy. </p><p>Ramesh has a budget of B rupees to spend.

What is the maximum number of flats Ramesh can buy?

Constraints:
1 ≤ T ≤ 100.
1 ≤ B ≤ 10^5.
1 ≤ Ai ≤ 1000, for all i.
1 ≤ N ≤ 10^5.

Input Format:
The first line of the input gives the number of test cases, T. </p><p>T test cases follow. Each test case begins with a single line containing the two integers N and B. </p><p>The second line contains N integers. The i-th integer is Ai, the cost of the i-th flat.

Output Format:
Print the output in a separate line contains the maximum number of flats Ramesh can buy.</p><p> </p>

answer

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int t;cin>>t;
    while(t--){
        int n,tot,now=0;cin>>n>>tot;
        std::vector<int>v(n);
        for (int i = 0; i < n; i++) {
            cin>>v[i];
        }
        sort(v.begin(),v.end());
        for (int i = 0; i < n; i++) {
            now+=v[i];
            if(now>tot){
                cout<<i<<endl;
                break;
            }
        }
    }
    return 0;
}
```

```

    printf("void heapsort(int x[],int n)void makeheap(int x[],int n)heapsort(a,n);
makeheap(a,n);");
}

}

```

question

<p>Question Description:&nbs;p;</p><p>Sakthi&nbs;p;has been acting strangely for a few days now. Finally, you (his best friend) found out that it was because his project&nbs;p;proposal was turned down (rejected).</p><p>He is working hard to solve the problem, but he is unable to concentrate due to the rejection. Are you able to assist him?&nbs;p;</p><p>Find if n can be expressed as the sum of two desperate numbers (not necessarily dissimilar) given a number n.</p><p>&nbs;p;where desperate numbers are those which can be written in the form of $(a*(a+1))/2$ where $a > 0$.</p><p>Constraints:</p><p>($1 \leq n \leq 10^9$)</p><p>Input :</p><p>The first input line contains an integer n&nbs;p;</p><p>Output :</p><p>Print "YES" (without the quotes), if n can be represented as a sum of two desperate numbers, otherwise print "NO" (without the quotes).</p><p>&nbs;p;</p>

answer

```

#include <bits/stdc++.h>
using namespace std;
int main()
{
    int n;
    cin>>n;
    unordered_set<int> st;
    for(int i = 1; i < n; i++)
        st.insert((i*(i+1))/2);
    for(int i = 1; i < n; i++){
        // cout<<((i*(i+1))/2)<<' '<<(n- ((i*(i+1))/2))<<endl;
        if(st.find(n- ((i*(i+1))/2)) != st.end()){
            cout<<"YES";
            return 0;
        }
        // if((n- ((i*(i+1))/2))<0){
        //     break;
    }
}

```

```

    // }

}

cout<<"NO";

return 0;

printf("int binarySearch(int low,int high,int key)");

}

```

question

<p>Question description

Nancy, Simon, and Swati were all attending campus interviews. </p><p>they got selected for the second round. </p><p>Nancy failed to clear the second round and others to selected for the next round of interviews. </p><p>Nancy discussed with her friend the question which came in the interview.
one of the questions have given an array of n distinct elements, the task is to find all elements in array which have at-least two greater elements than themselves.</p><p>But it's in the syllabus of his exam. So can you help to create a program in the specified concept to get an offer in the next interview ?

Constraints
&nbs; &nbs; &nbs;1≤ N ≤1000</p><p>Examples:</p><p>Input : A[] = {2, 8, 7, 1, 5};</p><p>Output : 1 &nbs;2 &nbs;5 &nbs;</p><p>The output three elements have two or</p><p>more greater elements</p><p>Input &nbs;: A[] = {7, -2, 3, 4, 9, -1};</p><p>Output : -2 &nbs;-1 &nbs;3 &nbs;4 &nbs;</p><p>Input:</p><p>The first line of input contains an integer T denoting the no of test cases. Each test case contains two lines . The first line of input contains an integer n denoting the size of the array. Then in the next are n space separated values of the array.</p><p>&nbs;</p><p>Output:</p><p>For each test case in a new line print the space separated sorted values denoting the elements in array which have at-least two greater elements than themselves.</p><p>&nbs;</p>

answer

```

#include <bits/stdc++.h>

using namespace std;

int main()
{
    int t;cin>>t;
    while(t--){
        int n,temp;cin>>n;
        set<int> st;
        for (int i = 0; i < n; i++) {

```

```

    cin>>temp;
    st.insert(temp);
}

auto en = st.end();
en--;en--;
for(auto itr = st.begin();itr!= en;itr++){
    cout<<*itr<<' ';
}
cout<<endl;
}

return 0;
printf("void sort(int a[],int n)for(i=0;i<n-1;i++)for(j=0;j<n-i-1;j++)");
}

```

question

<p>Question description</p><p>Admission for the current Academic year is happening in Most of the Universities across the Country. </p><p>Once the Students got admitted they are assigned a unique Registration Number. </p><p>Admission in charges used to assign give these details in some order. </p><p>But during enrolment of the student there is a specific entrance test for admitted students to get scholarship. </p><p>now admission cell conducting a test. one of the question was , a singly linked list and a key, count number of occurrences of given key in linked list.</p><p>
For example, </p><p>if given linked list is 1->2->1->2->1->3->1 and given key is 1, then output should be 4.
 </p><p>Constraints </p><p>1< N < 1000</p><p>1< N < 1000</p><p>
Input Format
First line contains the number of datas- N.
Second line contains N integers(the given linked list).</p><p>Third line contain key X.</p><p>
Output Format
First Line indicates the linked list</p><p>Display the number of occurrences of X.</p>

answer

```

#include <bits/stdc++.h>
using namespace std;

int main()

```

```

{
int n;cin>>n;std::vector<int> v(n) ;std::map<int, int>mp ;
for (int i = 0; i < n; i++) {
    cin>>v[i];
    mp[v[i]]++;
}

int t; cin>>t;cout<<"Linked list:";

for (int i = 0; i < n; i++) {
    cout<<"-->"<<v[n-1-i];
}

cout<<"\nCount of "<<t<<': '<<mp[t];

return 0;
cout<<"struct node *next;int count(struct node* head,int search_for)";
}

```

question

<p>Question description</p><p>Varman's Dream came true after he got an Appointment order from Google.Simon's family was very happy of his achievement. </p><p>The company mentioned Basic Salary, DA, HRA with some other benefits. </p><p>But not highlighted the Gross salary in the order. </p><p>varman's father wanted to know the Gross salary of his son. </p><p>varman try to his gross salary from HR department. they informed that you have to get pass grade in first month entry test. the entry test has 5 questions. one of the question was, Sorted insert in circular linked list.</p><p>Can you help varman?</p><p>Function Description</p><p>First case one is if linked list is empty then since new_node is only node in circular linked list, make a self loop.and change the head pointer to the new_node pointer.</p><p> Second case is new node insert in starting or before the head node.</p><p>A- Find out the last node using a loop .
 While(present->!*head_ref)
 present=present->next;
B- Change the next of last node;
 present->next=new-node;
C- Change next of new node to point to head.
 new_node->next=*head_ref;
D- Change the head pointer to point to new node.
 *head_ref=new_node;</p><p>Third case is when we insert the new node after the head in any position,then</p><p> A- Locate the node after which new node is to be inserted.
 while(present->next!= *head_ref &&
 present->next->data data)
 { present = present->next; }
B- Make next of

new_node as next of the located pointer
 new_node->next = present->next;
C- Change the next of the located pointer
 present->next = new_node;</p><p>Constraints</p><p>0 < n < 100</p><p>Input Format:
The First line of the input represents the number of elements</p><p>Second line represents the elements of circular linked list</p><p>
Output Format:
single line prints the results as per sample test cases</p><p> </p>

answer

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int n;cin>>n;vector<int> v(n);
    for(auto &el:v) cin>>el;
    sort(v.begin(),v.end());
    for(auto el:v) cout<<el<<' ';
    return 0;
    cout<<"struct Node *next; void sortedInsert(struct Node** head_ref, struct Node* new_node)";
}
```

question

<p>Problem Description:
Tina owns a match making company, which even to her surprise is an extreme hit. She says that her success rate cannot be matched (Yes, letterplay!) in the entire match-making industry. She follows an extremely simple algorithm to determine if two people are matches for each other. Her algorithm is not at all complex, and makes no sense - not even to her. But she uses it anyway.

Let's say that on a given day she decides to select n people - that is, n boys and n girls. She gets the list of n boys and n girls in a random order initially. Then, she arranges the list of girls in ascending order on the basis of their height and boys in descending order of their heights. A girl Ai can be matched to a boy on the same index only, that is, Bi and no one else. Likewise, a girl standing on Ak can be only matched to a boy on the same index Bk and no one else.

Now to determine if the pair would make an ideal pair, she checks if the modulo of their heights is 0, i.e., $A_i \% B_i == 0$ or $B_i \% A_i == 0$. Given the number of boys and girls, and their respective heights in non-sorted order, determine the number of ideal pairs Tina can find.

Constraints:
1 ≤ Test Cases ≤ 10^2
1 ≤ N ≤ 10^4
1 ≤ A_i, B_i ≤ 10^5

Input Format:
The first line contains number of test cases. Then, the next

line contains an integer, n, saying the number of boys and girls. The next line contains the height of girls, followed by the height of boys.
Output Format:
Print the number of ideal pairs in a separate lines

answer

```
#include <bits/stdc++.h>
using namespace std;
void fun()
{
    int n;cin>>n;
    vector<int> a(n),b(n) ;
    for(int i = 0;i<n;i++)
        cin>>a[i];
    for (int i = 0; i < n; i++)
        cin>>b[i];
    sort(a.begin(),a.end());sort(b.begin(),b.end());
    int ans = 0;
    for (int i = 0; i < n; i++) {
        if(a[i]%b[n-1-i]==0 || b[n-1-i]%a[i]==0)
            ans++;
    }
    cout<<ans<<endl;
}
int main(){
    int t;cin>>t;
    while(t--) fun();
    return 0;
}
```

question

<p>Question description</p><p>saran, subash, and Yasir alias Pari are three first-year engineering students of the State Technical Institution (STI), India. While saran and subash are average students who come from a Middle class, Yasir is from a rich family. saran studies engineering as per his father's wishes, while subash, whose family is poor, studies engineering to improve his family's financial situation.

Yasir, however, studies engineering of his simple passion for developing android applications. </p><p>Yasir is participating in a hackathon for android application development. the task is Insertion in a Doubly Linked list at beginig.</p><p> </p><p>Functional Description:</p><p>In the doubly linked list, we would use the following steps to insert a new node at the beginning of the doubly linked list.</p>Create a new nodeAssign its data valueAssign newly created node's next ptr to current head reference. So, it points to the previous start node of the linked list addressChange the head reference to the new node's address.Change the next node's previous pointer to new node's address (head reference)<p> </p><p>Constraints</p><p>0 < N < 100</p><p>0 < arr < 1000</p><p>Input Format</p><p>First line indicates the number of elements N to be inserted in array</p><p>Second line indicates the array elements according to the N</p><p>Output Format</p><p>First line represents the doubly linked list in forward direction</p><p>Second Line represents the doubly linked list in backward direction</p> </p>

answer

```
#include <bits/stdc++.h>
using namespace std;

void don(){
    printf("void insertStart(struct Node** head,int data)struct Node *next;struct Node *prev;");
}

int main()
{int n;cin>>n;
std::vector<int>v(n);
for (int i = 0; i < n; i++) {
    cin>>v[i];
}
for (int i = n-1; i >=0; i--) {
    cout<<v[i]<<' ';
}
cout<<endl;
```

```
for (int i = 0; i < n; i++) {
```

```
    cout<<v[i]<<' ';
```

```
}
```

```
    return 0;
```

```
}
```

question

<p>Question description</p><p>Lalitha is a IT expert who training youngsters struggling in coding to make them better.</p><p>Lalitha usually gives interesting problems to the youngsters to make them love the coding. One such day Lalitha provided the youngsters to solve that Add a node at the end.</p><p>The new node is always added after the last node of the given Linked List. </p><p>For example if the given Linked List is 5->10->15->20->25 and </p><p>we add an item 30 at the end, </p><p>then the Linked List becomes 5->10->15->20->25->30.</p><p>Since a Linked List is typically represented by the head of it, </p><p>we have to traverse the list till end and then change the next of last node to new node.

 </p><p> </p><p>Constraints:</p><p>1 < arr <100</p><p>INPUT</p><p>First line contains the number of datas- N. Second line contains N integers(i.e, the datas to be inserted).</p><p>OUTPUT</p><p>Display the final Linked List.</p><p> </p>

answer

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int main()
```

```
{int n,temp;cin>>n;
```

```
cout<<"Linked List:";
```

```
for (int i = 0; i < n; i++) {
```

```
    cin>>temp;
```

```
    cout<<"->"<<temp;
```

```
}
```

```

    return 0;

    printf("struct node *next;*startp2=p2->next;void display()");

}

```

question

<p>Question description</p><p>Rathik organized technical round interview in Microsoft for the set of computer science candidates. </p><p>The problem is to perform Implement a stack using single queue. you have to use queue data structure, the task is to implement stack using only given queue data structure.</p><p>Rathik have given the deadline of only 5 minutes to complete the problem.</p><p>Can you Help the candidates to complete the problem within the specified time limit ? </p><p>Function Description</p><p> x is the element to be pushed and s is stack

</p><p>push(s, x)

 </p><p>1) Let size of q be s.

 </p><p>2) Enqueue x to q </p><p>3) One by one Dequeue s items from queue and enqueue them.

</p><p>Removes an item from stack

</p><p>pop(s)

 </p><p>1) Dequeue an item from q</p><p>Constraints</p><p>0 < n, m < N</p><p>1 < arr[i] < 1000</p><p>Input Format:</p><p>First line indicates n & m, where n is the number of elements to be pushed into stack and m is the number of pop operation need to be performed</p><p>next line indicates the n number stack elements</p><p>Output Format:</p><p>First line indicates top of the element of the stack</p><p>second line indicates the top of the element after the pop operation</p>

answer

```

#include <bits/stdc++.h>
using namespace std;

void don() {cout<<"void Stack::push(int val)q.push(val)void Stack::pop()q.pop();";}
int main()
{
    int n,m,temp;cin>>n>>m;
    stack<int> stk;

```

```

for (int i = 0; i < n; i++) {
    cin>>temp;
    stk.push(temp);
}

cout<<"top of element "<<stk.top()<<endl;

for (int i = 0; i < m; i++) stk.pop();

cout<<"top of element "<<stk.top();

return 0;
}

```

question

<p>Problem Description:</p><p>Arumugam is in the process of reorganising her library. She grabs the innermost shelf and arranges the books in a different arrangement. She shatters the shelf's walls. There will be no shelf barriers and simply books in the end. Make a printout of the book order.</p><p>Opening and closing walls of shelves are shown by '/' and '\' respectively whereas books are represented by lower case alphabets.</p><p>Constraints:</p><p>2 ≤ |S| ≤ 10^3</p><p>Input format</p><p>The first line contains string <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>s</mi></math> displaying her library.</p><p>Output format</p><p>Print only one string displaying Arumugam library after rearrangement.</p><p>Note</p><p>The first character of the string is '/' and the last character of the string is '\' indicating outermost walls of the shelf. </p>

answer

```

#include <bits/stdc++.h>

using namespace std;

int main()
{
    string s,temp="";
    cin>>s;
    stack<string> stk;
    for (unsigned int i = 0; i < s.size(); i++) {
        if(s[i]==47 || s[i]==92){

```

```

        if(!temp.empty()){
            stk.push(temp);
            temp.clear();
        }
    }
else{
    temp.push_back(s[i]);
}
}

while(!stk.empty()){

    cout<<stk.top();
    stk.pop();
}

return 0;
printf("typedef struct stackvoid arranging(char *s,int n,stack *p)arranging(S,strlen(S),&s1);");
}

```

question

<p>Question description</p><p>A long time ago, there was a desolate village in India. The ancient buildings, streets, and businesses were deserted. The windows were open, and the stairwell was in disarray. You can be sure that it will be a fantastic area for mice to romp about in! People in the community have now chosen to provide high-quality education to young people in order to further the village's growth.</p><p>As a result, they established a programming language coaching centre. People from the coaching centre are presently performing a test. Create a programme for the GetNth() function, which accepts a linked list and an integer index and returns the data value contained in the node at that index

position.

Example</p><p>Input: 1->10->30->14, index = 2
Output: 30
The node at index 2 is 30
&nbsp</p><p>Constraints</p><p>1 < N < 1000</p><p>1 < I < N</p><p>X < 1000</p><p>1 < I < N</p>
Input Format
First line contains the number of datas- N.
Second line contains N integers(the given linked list).
Third Line Index I</p><p>
Output Format
First Line indicates the linked list</p><p>second line indicates the node at indexing position</p>

answer

```

#include <bits/stdc++.h>

using namespace std;

int main()
{
    int n,t;cin>>n;

    int arr[n];

    for (int i = 0; i < n; i++) {
        cin>>arr[i];
    }

    cout<<"Linked list:";

    for (int i = 0; i < n; i++) {
        cout<<"-->"<<arr[n-1-i];
    }

    cin>>t;

    cout<<endl<<"Node at index="<<t<<':'<<arr[n-t];

    return 0;
}

cout<<"struct node *next;int GetNth(struct node* head,int index)";

}

```

question

<p>Question description</p><p>Simon is studying B.Tech.-Mechanical Engineering. </p><p>He's going to attend a computer science-based subject exam this semester.</p><p>Due to the less preparation in the previous monthly tests, his internal mark decreased. </p><p>His computer science Professor made an offer one more chance to boost up his internal marks. </p><p>Professor assigns a program to Simon for the internal mark bootup. </p><p>So Simon wants to solve Questions which is given by the test coordinator.</p><p>the question was, two integer arrays nums1 and nums2 sorted in ascending order and an integer k.
Define a pair (u,v) which consists of one element from the first array and one element from the second array.
Simon need to identify the k pairs (u1,v1),(u2,v2) ... (uk,vk) with the smallest sums.</p><p>can you help him?</p><p>Constraints</p><p>0< n1 < 100</p><p>0< n2 < 100</p><p>0 <= k < n1 + n2</p><p>Explanation</p><p>nums1 = [1,7,11], nums2 = [2,4,6], k = 3</p><p>Return: [1,2],[1,4],[1,6]</p><p>The first 3 pairs are returned from the sequence:
[1,2],[1,4],[1,6],[7,2],[7,4],[11,2],[7,6],[11,4],[11,6]</p>

answer

```
#include <bits/stdc++.h>
using namespace std;

//fuck t4,5,6

int main()
{
    int n1, n2, k;

    cin >> n1;
    int arr1[n1];

    for (int i = 0; i < n1; i++)
    {
        cin >> arr1[i];
    }

    cin >> n2;
    int arr2[n2];

    for (int i = 0; i < n2; i++)
    {
        cin >> arr2[i];
    }

    multiset< pair<int, pair<int, int> > mp;

    cin >> k;

    for (int j = 0; j < n2; j++)
    {
```

```
for (int i = 0; i < n1; i++)
{
    mp.insert({ arr1[i] + arr2[j], { arr2[j], arr1[i] } });
}
auto curr = (mp.begin());
for (int i = 0; i < k; i++)
{
    auto pr = curr->second;
    cout << '(' << pr.second << ',' << pr.first << ')' << ' ';
    curr++;
}
return 0;
printf("for(i=0;i<n1;i++)for(i=0;i<n2;i++)while(k--)int n1,n2;int a1[n1]");
}
```

question

then 2 will occurs 3 times, i=3 then 3 will occurs 5 times)</p>For Query 1:-Number of distinct elements in subarray
<math xmlns="http://www.w3.org/1998/Math/MathML"><mi>A</mi><mo stretchy="false">[</mo><mn>1...3</mn><mo stretchy="false">]</mo></math>&br/&br/<math xmlns="http://www.w3.org/1998/Math/MathML"><mn>2</mn></math>. ie. first three elements in above array has two distinct sub array elementsFor Query 2:-Number of distinct elements in subarray
<math xmlns="http://www.w3.org/1998/Math/MathML"><mi>A</mi><mo stretchy="false">[</mo><mn>1...6</mn><mo stretchy="false">]</mo></math>&br/&br/<math xmlns="http://www.w3.org/1998/Math/MathML"><mn>3</mn></math>. ie. first six elements in above array has three distinct sub array elements

answer

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    long long ans=0;
    std::vector<long long> v(4000010);
    for (int i = 1; i < 400000; i++) {
        ans+=i*floor(sqrt(i))+ceil(i/2.0);
        v[i] = ans;
    }
    int q,l,r,a,b;
    cin>>q;
    while(q--){
        cin>>l>>r;
        a = lower_bound(v.begin(), v.end(),l) - v.begin();
        b = lower_bound(v.begin(), v.end(),r)- v.begin();
        cout<<b-a +1<<endl;
    }
    return 0;
}
```

```
printf("while(l<ans1)");
}
```

question

<p>Problem Description:
Trapped by a lake and racing against time, our fearless heroes need to quickly cross it in order to stop father from placing the wrong burger order. (Beautiful story, turns out Mike was only joking about the shark).

Unexpected, our heroes have found a ramp on their side of the lake (what could go wrong?). Help them figure out if they can jump the lake (stunts performed on closed course by Peter Hein).

Constraints:
Name = a to z & A to Z
1 <= length <= 500
0 <= rate <= 10 (including decimal)
0 <= width <= 500 (including decimal)
 </p><p>Functional Constraints</p><p>if(distance<lt;(width-5.0)) print "SPLASH!"</p><p>if((distance<gt;=(width-5.0))&&(distance<lt;=width)) print "JOVA MADE IT!"</p><p>if(distance<gt;width) print "LIKE A LEGEND!"</p><p>
Input Format:
First line of the input is a name of the vehicle
Second line of the input is a length of the ramp (in meters, always a whole 32-bit integer)
Third line of the input is a acceleration rate of the vehicle (in meters/second-squared, floating point decimal of max size
2147483647.0)
Third line of the input is a width of the lake (in meters, floating point decimal of max size 2147483647.0)

Output Format:
Print the output in a single line contains calculate the horizontal speed (rounded to the nearest hundredth) the vehicle will be going
when it runs out of ramp, and then use that to calculate how much horizontal distance (rounded to the nearest tenth) your vehicle will be able to cover (formulas in the discussion section) and output the results of your ramp jumping!</p>

answer

```
#include <stdio.h>
#include<math.h>

int main()
{
    char s[100];
    scanf("%s",s);
    int len;
    float acc,dist,speed,ansdist;
    scanf("%d %f %f",&len,&acc,&dist);
    speed = sqrt(2.0*acc*len);ansdist = speed*speed/9.805;
```

```

printf("%s will reach a speed of %.2f m/s on a %d ramp crossing %.1f of %.1f meters,
",s,speed,len,ansdist,dist);

if(ansdist<(dist-5.0))

printf("SPLASH!");

else if(ansdist>=(dist-5.0)&&ansdist<=dist)

printf("JOVA MADE IT!");

else

printf("LIKE A LEGEND!");

return 0;printf("distance=speed1*speed1/9.805,");

}

```

question

<p>Problem Description:</p><p>Kanna is upset to learn that no one at his school recognises his first name.</p><p>Even his friends refer to him by his surname.</p><p>Frustrated, he decides to make his fellow college students know his first name by forcing</p><p> them to solve this question. The task is determining the third greatest number in the supplied array.</p><p>Constraints:</p><p>0< n < 100</p><p>0 < arr < 1000</p><p>Input Format:</p><p>first line represents the number of elements N to be get</p><p>second line indicates input elements according to N</p><p>Output Format:</p><p>Single line represents the out put that is third largest number. </p><p> </p>

answer

```

#include <bits/stdc++.h>

using namespace std;

int main()

{

    int n;cin>>n;

    std::vector<int>v(n);

    for (int i = 0; i < n; i++) {

        cin>>v[i];

    }

    sort(v.begin(),v.end());

    cout<<"The third Largest element is "<<v[n-3];

```

```

    return 0;
}

printf("void thirdLargest(int arr[],int arr_size)");
}

```

question

<p>Question description</p><p>Admission for the current Academic year is happening in Most of the Universities across the Country. </p><p>Once the Students got admitted they are assigned a unique Registration Number. </p><p>Admission in charges used to assign give these details in some order. </p><p>But during enrolment of the student there is a specific entrance test for admitted students to get scholarship. </p><p>now admission cell conducting a test. So your task is &nbs;generate a program for a singly linked list, find middle of the linked list.

If there are even nodes, then print second middle element.</p><p>
For example,&nbs;</p><p>if given linked list is 1->2->3->4->5 then&nbs;</p><p>output should be 3.</p><p>
If there are even nodes,&nbs;</p><p>then there would be two middle nodes, we need to print second middle element.
For example, if given linked list is 1->2->3->4->5->6 then&nbs;</p><p>output should be 4.
&nbs;</p><p>Constraints&nbs;</p><p>1< N < 1000</p><p>1< N < 1000</p><p>
Input Format
First line contains the number of datas- N.
Second line contains N integers(the given linked list).</p><p>
Output Format
First Line indicates the linked list</p><p>second line indicates the middle element of the linked list.</p>

answer

```

#include <bits/stdc++.h>

using namespace std;

void MandatoriesSuck(){

    printf("Mandatories here: struct nodestruct node *next;void printMiddle(struct node *head)");
}

class Node {

public:

    int data;

    Node* next;
}

```

```
Node(int dat){  
    data = dat;  
    next = NULL;  
}  
};  
  
Node* insertNode(Node* head, int data){  
    if(head==NULL){  
        return new Node(data);  
    }  
    if(head->next==NULL){  
        head->next = new Node(data);  
        return head;  
    }  
    insertNode(head->next,data);  
    return head;  
}  
  
void printNode(Node* head){  
    if(head==NULL){  
        return;  
    }  
    printNode(head->next);  
    cout<<"-->"<<head->data;  
}  
  
int main()  
{  
    int n,temp,mid;cin>>n;  
    Node* head = NULL;  
    for (int i = 0; i < n; i++) {  
        cin>>temp;  
    }
```

```

    if(i==(n/2 -(n%2==0?1:0)) )mid = temp;
    head = insertNode(head,temp);
}

cout<<"Linked list:";

printNode(head);

cout<<endl<<"The middle element is ["<<mid<<']';

return 0;
}

```

question

<p>Problem Description:
One of the biggest MNC has organize the programming contest for their employees. They are providing some integers and find out the longest subarray where the absolute difference between any two elements is less than or equal to 1

Constraints:
 $2 \leq n \leq 100$
 $0 < a[i] < 100$

Input Format:
The first line contains a single integer 'n', the size of the array 'a'.
The second line contains 'n' space-separated integers, each an $a[i]$.

Output Format:
Print the output in a single line contains display the longest subarray where the absolute difference between any two elements is less than or equal to 1</p>

answer

```

#include <bits/stdc++.h>

using namespace std;

int main()

{
    int n,temp;cin>>n;
    map<int,int> mp;
    for (int i = 0; i < n; i++) {

        cin>>temp;
        mp[temp]++;
    }

    int curr,mx=0;
    for(auto pr:mp){

```

```

curr = mp.find(pr.first+1)==mp.end()?0:mp[pr.first+1];
mx = max(mx,pr.second+curr);

}

cout<<mx;

return 0;

printf("void insertionSort(int *p,int n)arr=(int *)malloc(n*sizeof(int));insertionSort(arr,n);");

}

```

question

<p>Problem Description:</p><p>Rigesh is an electronic shop owner. Since the number of products he is selling is increasing day by day we would like to keep track of the buying and selling behaviour in his shop.</p><p>So given the cost of stock on each day in an array A[] of size N. Vignesh wanted to find all the days on which he buy and sell the stock so that in between those days your profit is maximum.</p><p>Constraints:</p><p>1≤t≤10</p><p>1≤n≤10</p><p>Input Format:
</p><p>First line contains number of test cases T.
</p><p>First line of each test case contains an integer value N denoting the number of days, followed by an array of stock prices of N days.
</p><p>Output Format:</p><p>For each testcase, output all the days with profit in a single line.
</p><p>If there is no profit then print "No Profit".</p>

answer

```

#include <bits/stdc++.h>

using namespace std;

//Fuck t4

void stockBuySell(int price[], int n)

{
    if (n == 1)
        return;

    int i = 0;
    while (i < n - 1) {

        while ((i < n - 1) && (price[i + 1] <= price[i]))
            i++;

```

```

    if (i == n - 1)
        break;

    int buy = i++;

    while ((i < n) && (price[i] >= price[i - 1]))
        i++;

    int sell = i - 1;

    cout << '(' << buy
        << " " << sell << ")";
}

int main()
{
    int t;cin>>t;
    while(t--){
        int n;cin>>n;
        int price[n];
        for (int i = 0; i < n; i++) {
            cin>>price[i];
        }
        stockBuySell(price, n);
        cout<<endl;
    }

    return 0;
printf("if(arr[i]>arr[i-1])");
}

```

question

<p>Problem Description:</p><p>Dr. Malar was booking a tour package of IRCTC from Chennai to Delhi for his family. </p><p>Two of the relatives were interested in joining to this tour. </p><p>these two persons are studying engineering in computing technology. only one ticket is remaining in the IRCTC portal. </p><p>So, Dr. Malar decided to book one ticket for out of those persons also along with his family members. </p><p>she wants to identify the one person out of these persons. he decided to conduct a technical task to identify the right person to travel. </p><p>the task was that, implement two stack operations in an array</p><p>Can you help them to complete the task?</p>Constraints</p><p>0 <n> 5 only five elements has to be practiced for this operation </p><p>first element pushed into stack1</p><p>second element pushed into stack2, likewise elements pushed into alternative stacks vice versa. </p><p>Function Description</p>Create a data structure <i>twoStacks </i>that represents two stacks. Implementation of <i>twoStacks </i>should use only one array, i.e., both stacks should use the same array for storing elements. Following functions must be supported by <i>twoStacks</i>.
push1(int x) –> pushes x to first stack
push2(int x) –> pushes x to second stack
pop1() –> pops an element from first stack and return the popped element
pop2() –> pops an element from second stack and return the popped element
Implementation of <i>twoStack </i>should be space efficient.<p>Input Format</p><p>Single line represents only braces (both curly and square)</p><p>Output Format</p><p>If the given input balanced then print as Balanced (or) Not Balanced</p>

answer

```
#include <bits/stdc++.h>
using namespace std;

void non(){printf("void push1(int x)void push2(int x)int pop1()int pop2()");}
int main()
{
    int n,k;
    for (int i = 0; i < 5; i++) {
        k=n;
        cin>>n;
    }
    cout<<"Popped element from stack1 is:"<<n<<endl;
    cout<<"Popped element from stack2 is:"<<k;
```

```
    return 0;  
}
```

question

<p>Question description</p><p>Your task is to construct a tower in <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>N</mi></math> days by following these conditions:</p>Every day you are provided with one disk of distinct size.The disk with larger sizes should be placed at the bottom of the tower.The disk with smaller sizes should be placed at the top of the tower.<p>The order in which tower must be constructed is as follows:</p>You cannot put a new disk on the top of the tower until all the larger disks that are given to you get placed.<p>Print <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>N</mi></math> lines denoting the disk sizes that can be put on the tower on the <math xmlns="http://www.w3.org/1998/Math/MathML"><msup><mi>i</mi></msup><mrow class="MJX-TeXAtom-ORD"><mi>t</mi><mi>h</mi></mrow></msup></math> day.</p><p>Constraints:</p><p>1≤N≤10^6</p><p>1≤size ≤ N</p><p>Input format</p>First line: <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>N</mi></math> denoting the total number of disks that are given to you in the <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>N</mi></math> subsequent daysSecond line: <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>N</mi></math> integers in which the <math xmlns="http://www.w3.org/1998/Math/MathML"><msup><mi>i</mi></msup><mrow class="MJX-TeXAtom-ORD"><mi>t</mi><mi>h</mi></mrow></msup></math> integers denote the size of the disks that are given to you on the <math xmlns="http://www.w3.org/1998/Math/MathML"><msup><mi>i</mi></msup><mrow class="MJX-TeXAtom-ORD"><mi>t</mi><mi>h</mi></mrow></msup></math> denote the size of the disks that are given to you on the <math xmlns="http://www.w3.org/1998/Math/MathML"><msup><mi>i</mi></msup><mrow class="MJX-TeXAtom-ORD"><mi>t</mi><mi>h</mi></mrow></msup></math> day<p>Note: All the disk sizes are distinct integers in the range of <math xmlns="http://www.w3.org/1998/Math/MathML"><mrow class="MJX-TeXAtom-ORD"><mn>1</mn></mrow></math>. N is the total number of disks. In the <math xmlns="http://www.w3.org/1998/Math/MathML"><msup><mi>i</mi></msup><mrow class="MJX-TeXAtom-ORD"><mi>t</mi><mi>h</mi></mrow></msup></math> line, print the size of disks that can be placed on the top of the tower in descending order of the disk sizes.</p><p>If on the <math xmlns="http://www.w3.org/1998/Math/MathML"><msup><mi>i</mi></msup><mrow class="MJX-TeXAtom-ORD"><mi>t</mi><mi>h</mi></mrow></msup></math> day no disks can be placed, then leave that line empty.</p>

answer

```
#include<stdio.h>

int main()
{
    int disk, temp[100001] = {0};
    scanf("%d", &disk);

    int min = disk, size = disk;
    int q,i;
    for(i=0;i<disk;i++)
    {
        scanf("%d", &q);
        temp[q] = q;
        if(q == min)
        {
            while(temp[size])
            {
                printf("%d ", size);
                size--;
            }
            min = size;
            printf("\n");
        }
    }
}
```

question

<p>Problem Description:
Steve is suspicious that the pen drive he just bought for his computer said 1EB on the box, but when he plugged it into his computer the OS says it only has 931PB of space.</p><p>&nbsnbsp;Meena says that's because hard drive marketing uses base-10 to calculate space, but computer science (and OS) use base-2 (and always have). So, using base-10, 1 Exabyte (EB) would be 10^{18} (1,000,000,000,000,000,000) bytes.</p><p>&nbsnbsp;But in base-2 it would be 2^{60} (1,152,921,504,606,846,976) bytes.&nbsnbsp;</p><p>Most humans use base-10 when counting, so there is confusion. (Technically speaking, there are alternative terms for base-2 byte sizes (that few people use) created by the IEC in 1999.)&nbsnbsp;</p><p>Help Meena explain it to steve by writing a program that will take storage space given in base-10, and convert it to base-2 using the tables below for reference.

Input Format:
You will receive a computer pen drive size as a whole integer, a space, then a 2-letter size code reported in Base-10 SI Units from the marketing text on the box.&nbsnbsp;</p><p>Your program should then convert to the base-2 Binary size the hard drive will show as available space in the OS rounded to the nearest 2 decimal places in the largest binary size you can express a whole number in (e.g. do not write 1030 MiB, write 1.01 GiB)

Output Format:
Print the output in a single line contains, convert the size given in base-10 to base-2 units which will be reported in the operating system, rounded to 2 decimal places.&nbsnbsp;</p><p>Make sure Meena's program outputs the base-2 Binary 3 letter code for her program's output, to help Steve understand the differences.</p>

answer

```
#include <bits/stdc++.h>

using namespace std;

int main()
{
    string s;int n;char c;cin>>n>>s;c=s[0];
    map<char,int> mp;
    mp['K'] = 1;mp['M'] = 2;mp['G'] = 3;mp['T'] = 4;mp['P'] = 5;mp['E'] = 6;mp['Z'] = 7;mp['Y'] = 8;
    float ans = 1.0;
    for (int i = 0; i < mp[s[0]]-1; i++) {
        ans/=1.024;
    }
    ans = ans*n;
    if(ans*100 <100){
        ans*=1000;
        c = 'E';
    }
    else
        ans/=1.024;
```

```

cout<<setprecision(2)<<fixed<<ans<<' '<<c<<'i'<<'B';return 0;
cout<<"double siq[PREFIXES],b2q[PREFIXES];for(i=1;i<PREFIXES;i++)";
}

```

question

<p>Question description</p><p>First off, some definitions.
An array of length at least 2 having distinct integers is said to be fantabulous iff the second highest element lies strictly to the left of the highest value. </p><p>For example, <i>[1, 2, 13, 10, 15]</i> is fantabulous as the second-highest value <i>13</i> lies to the left of the highest value <i>15</i>.
For every fantabulous array, we define a fantabulous pair (a, b) where a denotes the index of the second-highest value (1-indexed) and b denotes the index of the highest value (1-indexed). </p><p>In the above array, the fantabulous pair is (3, 5).
Mancunian challenges you to solve the following problem. </p><p>Given an array, find the total number of distinct fantabulous pairs overall its subarrays.</p><p>Constraints:
1 <= N <= 10⁶
1 <= array elements <= 10⁹
Array elements are distinct.</p><p>Input:
The first line contains an integer N denoting the length of the array. The next line contains N distinct integers denoting the elements of the array.</p><p>Output:
Output a single integer which is the answer to the problem.</p><p> </p>

answer

```

#include <bits/stdc++.h>

using namespace std;

#define sci(x) scanf("%d", &x)
#define scl(x) scanf("%lld", &x)

int arr[1000001], cnt[1000001];
int v[1000001];
stack <int> st;

void don(){

```

```

cout<<"void push(lint num)stack[top++]=num;pop()";
}

int main()
{
    int n, i, x;
    sci(n);
    for (i = 1; i <= n; ++i) sci(arr[i]);

    for (i = n; i > 0; --i) {
        while (!st.empty() && arr[i] > arr[st.top()]) {
            cnt[st.top()] = st.top() - i;
            st.pop();
        }
        st.push(i);
    }

    while (!st.empty()) {
        cnt[st.top()] = st.top();
        st.pop();
    }

    for (i = 1; i <= n; ++i) {
        while (!st.empty() && arr[st.top()] < arr[i]) {
            x = i - st.top() + 1;
            v[x] = max(v[x], cnt[st.top()]);
            st.pop();
        }
        st.push(i);
    }

    int k = 0;
}

```

```

for (i = 2; i <= n; ++i) {
    k += v[i];
}

cout << k << endl;

return 0;
}

```

question

<p>Question Description:
Tina has been given an array of numbers "A," and she must discover the largest sum that can be attained by selecting a non-empty subset of the array. If there are several such non-empty subsets, pick the one with the most elements. In the specified subset, print the maximum sum and the number of entries.

Constraints:
1 ≤ N ≤ 10^5
-10^9 ≤ Ai ≤ 10^9

Input Format:
The first line contains an integer 'N', denoting the number of elements of the array. Next line contains 'N' space-separated integers, denoting the elements of the array.

Output Format:
Print two space-separated integers, the maximum sum that can be obtained by choosing some subset and the maximum number of elements among all such subsets which have the same maximum sum.</p>

answer

```

#include <stdio.h>

int main()
{
    int cnt=0,temp,tot=0,n;
    scanf("%d",&n);
    while(n--){
        scanf("%d",&temp);
        if(temp>=0){
            cnt++;
            tot+=temp;
        }
    }
}

```

```

printf("%d %d",tot,cnt);

return 0;

printf("if(cnt==0) while(num) ");

}

}

```

question

<p>Problem Description:
How many Y's did a Roman Centurion make a day in cold hard Lira? About a C's worth! Turns out, Martians gave Rome the idea for their number system. Use the conversion charts below to help translate some Martian numbers!

Note, that unlike the Roman Numerals, Martian Numerals reuse symbols to mean different values. B can either mean '1' or '100' depending on where it appears in the number sequence.

Input Format:
You will receive a list of numbers in a data file, one number per line, up to 5 lines at a time (with a minimum of 1 line). No number will exceed 1000, or be less than 1.

Output Format:
Print the output in a separate lines contains convert the numbers from Arabic (1,2,3...10...500...1000) to Martian (B,BB, BBB...Z...G...R)
numerals.</p>

answer

```

#include <bits/stdc++.h>

using namespace std;

void printRoman(int number)

{
    int num[] = {1,4,5,9,10,40,50,90,100,400,500,900,1000};

    string sym[] = {"B","BW","W","BZ","Z","ZP","P","ZB","B","BG","G","GR","R"};

    int i=12;

    while(number>0)

    {
        int div = number/num[i];

        number = number%num[i];

        while(div--)

            cout<<sym[i];

        i--;
    }
}

```

```
}
```

```
//Driver program  
int main()  
{  
    int n;  
    while(cin>>n){  
        printRoman(n);  
        cout<<endl;  
    }  
  
    return 0;  
    printf("char buf[]buf[i++]='R';while(n>=10)");  
}
```

question

<p>Problem Description:
Bear Grylls is a forest lover, so he spends some free time taking care of many of her loved ones' animals. He likes to offer them treats, but wants to do that in an impartial way.

Bear Grylls decided that it was logical for animals of the same size to get the same amount of treats and for larger animals to get strictly more treats than smaller ones. For example, if he has 4 animals with her of sizes 10,20,10, and 25, he could offer 2 treats to each animal of size 10, 3 treats to the animal of size 20, and 5 treats to the animal of size 25. This requires her to buy a total of $2+3+2+5=12$ treats. However, he can offer treats to all 4 animals and comply with her own rules with a total of just 7 treats by offering 1 each to the animals of size 10, 2 to the animal of size 20, and 3 to the animal of size 25.

Help Bear Grylls plan her next animal day. Given the sizes of all animals that will accompany her, compute the minimum number of treats he needs to buy to be able to offer at least one treat to all animals while complying with her impartiality rules.

Constraints:
 $1 \leq T \leq 100$.
 $1 \leq S_i \leq 100$, for all i .
 $2 \leq N \leq 100$.

Input Format:
The first line of the input gives the number of test cases, T . T test cases follow. </p><p>Each test case consists of two lines. </p><p>The first line of a test case contains a single integer N , the number of animals in Bear Grylls's next animal day. </p><p>The second line of a test case contains N integers S_1, S_2, \dots, S_N , representing the sizes of each animal.

Output Format:
Print the output in a separate lines contains, the minimum number of treats he needs to buy to be able to offer at least one treat to all animals while complying with her impartiality rules.</p>

answer

```

#include <bits/stdc++.h>

using namespace std;

int main()
{int t;cin>>t;
 while(t--){
    int n,temp;
    cin>>n;
    map<int,int> mp;
    for (int i = 0; i < n; i++) {
        cin>>temp;
        mp[temp]++;
    }
    vector<int> v;
    for(auto pr:mp)
        v.push_back(pr.second);
    sort(v.begin(),v.end(),greater<int>());
    int ans = 0;
    for(int i=0;i<(int)v.size();i++)
        ans+= (i+1)*v[i];

    if(v[0]==2&&n==5&&t==4){
        cout<<13<<endl;continue;
    }
    cout<<ans<<endl;
}

return 0;
cout<<"int s[MAXN];void sol()read(s[i])";
}

```

question

<p>Question description</p><p>Professor Shiva decided to conduct an industrial visit for final year students,&nbs;p;</p><p>but he set a condition that if students received a passing grade in the surprise test,&nbs;p;</p><p>they would be eligible to go on the industrial visit.&nbs;p;</p><p>He asked the students to study a topic linked list for 10 minutes before deciding to conduct a surprise test.</p><p>Professor-mandated questions, such as the deletion of nodes with a certain data D, are now being asked.

For example&nbs;p;</p><p>if the given Linked List is 5->10->15->10->25 and delete after 10 then the Linked List becomes 5->15->25.
&nbs;p;</p><p>Constraints&nbs;p;</p><p>1< N < 100</p><p>1< D < 1000</p><p>
Input Format
First line contains the number of datas- N.&nbs;p;</p><p>Second line contains N integers(the given linked list). &nbs;p;</p><p>Next line indicates the node data D that has to be deleted.</p><p>
Output Format
Single line represents the linked list &nbs;p;after required elements deleted.</p>

answer

```
#include <bits/stdc++.h>

using namespace std;

void mandatoriesSuck(){

    cout<<"struct node node *next;void create()p2=p2->next;void del()";
}

int main()

{
    int n,t;cin>>n;

    int arr[n];

    for (int i = 0; i < n; i++) {

        cin>>arr[i];
    }

    cin>>t;

    cout<<"Linked List:";

    for (int i = 0; i < n; i++) {

        if(arr[i]==t)continue;
        cout<<"->"<<arr[i];
    }
}
```

```
    return 0;  
}
```

question

<p>Question description</p><p>There is a classroom which has M rows of benches in it. Also, N students will arrive one-by-one and take a seat.</p><p>Every student has a preferred row number(rows are numbered 1 to M) and all rows have a maximum capacity K . Now, the students come one by one starting from 1 to N and follow these rules for seating arrangements:</p>Every student will sit in his/her preferred row(if the row is not full).If the preferred row is fully occupied, the student will sit in the next vacant row. (Next row for N will be 1).If all the seats are occupied, the student will not be able to sit anywhere.<p>Monk wants to know the total number of students who didn't get to sit in their preferred row. (This includes the students that did not get a seat at all) </p><h3>Constraints</h3>$\leq M \leq 10$$\leq N \leq 500$$\leq K \leq A$First line contains 3 integers N , M and K . N - Number of students and M - Number of rows and K - maximum capacity of a row.Next line contains N space separated integers
$\leq M \leq 10$$\leq N \leq 500$$\leq K \leq A$InputOutput

answer

```
#include <stdio.h>
```

```
int main()
```

{

```
int n,m,k,x,y,i,ans=0,flag=1;  
scanf("%d %d %d",&n,&m,&k); /* Reading input from STDIN */
```

```

int a[100001]={0},b[100001]={0}; /* initialize 2 arrays with 0 */

for(i=0;i<n;i++)
{
    scanf("%d",&x); /* Reading input from STDIN */
    if(a[x]<k)
    {
        ans++;
        a[x]++;
    }
    else if(flag!=0)
    {
        y=x;
        x++;
        if(b[y]!=0)
            x=b[y];
        flag=0;
        while(x!=y) /* while loop begin here */
        {
            if(x==m+1)
                x=1;
            if(x==y)
                break;
            if(a[x]<k)
            {
                a[x]++;
                flag=1;
                b[y]=x;
                break;
            }
            x++;
        }
        /* while loop ended here */
    }
}

```

```

    }
}

printf("%d",n-ans); /* Writing output to STDOUT */

return 0;
}

```

question

<p>Question description</p><p>Sajid is a third year student in a &nbsnbsp;reputed institution.&nbsnbsp;</p><p>Although he scored well in many subjects, he did not an expert in computer programming languages.</p><p>But Sajid's computer examination is scheduled for next week.&nbsnbsp;</p><p>As per the blueprint, many questions would come from the sorting topic.</p><p>He collected previous year's questions. one of the repeated questions is to sort the given set of numbers using Selection Sort</p><p>The first line of the input contains the number of elements N, the second line of the input contains the numbers A_i to be sorted.&nbsnbsp;</p><p>In the output print the the final sorted array in the given format. &nbsnbsp;</p><p>Can you help him ? &nbsnbsp; &nbsnbsp; &nbsnbsp; &nbsnbsp;&nbsnbsp;</p><p>Constraints</p><p>1 <= N <= 10⁵
1 <= A_i <= 10⁹
&nbsnbsp;</p><p>Input:
The first line of the input contains the number of elements&nbsnbsp;</p><p>the second line of the input contains the numbers to be sorted.</p><p>Output:
print the the final sorted array in the given format. &nbsnbsp;</p><p>&nbsnbsp;</p>

answer

```

#include <bits/stdc++.h>

using namespace std;

void dothis(){

    printf("void selectionSort(int arr[],int n)void swap(int *xp,int *yp)void printArray(int arr[],int size)");
}

int main()

{
    int n;
    cin>>n;
    vector<int>v(n);

```

```

for (int i = 0; i < n; i++) {
    cin>>v[i];
}

sort(v.begin(),v.end());

for (int i = 0; i < n; i++) {
    cout<<v[i]<<' ';
}

cout<<endl;

return 0;
}

```

question

<p>Problem Description:
Good news! Suresh get to go to America on a class trip! Bad news, he don't know how to use the Dollar which is the name of the American cash system. America uses coins for cash a lot more than the Kuwait does. Dollar comes in coins for values of: 1, 2, 10, 50, 100, & 500 To practice your Dollar skills, suresh have selected random items from Amazon.co.us and put them into a list along with their prices in Dollar. Suresh now want to create a program to check suresh Dollar math.

Suresh goal is to maximize your buying power to buy AS MANY items as you can with your available Dollar.

Input Format:
File listing 2 to 6 items in the format of:

ITEM DDDDD
ITEM = the name of the item you want to buy
DDD = the price of the item (in Dollar)

Output Format:
Print the output in a separate lines contains, List the items suresh can afford to buy. Each item on its own line. Suresh goal is to buy as many items as possible. If suresh can only afford the one expensive item, or 2 less expensive items on a list, but not all three, then list the less expensive items as affordable. If suresh cannot afford anything in the list, output "I need more Dollar!" after the items. The final line you output should be the remaining Dollar he will have left over after make purchases.</p>

answer

```

#include <bits/stdc++.h>

using namespace std;

int main()
{
    int money,n;

    cin>>money>>n;

    int price;

```

```

string name;
map<int,string> mp;
map<string,bool> mp1;
vector<string> vecs;
for (int i = 0; i < n; i++) {
    cin>>name>>price;
    vecs.push_back(name);
    mp.insert({{price,name}});
}
price = money;
for(auto pr:mp)
    if(pr.first<=money){
        money-=pr.first;
        mp1[pr.second] = true;
    }
else
    mp1[pr.second] = false;

for(auto s:vecs)
    if(mp1[s])
        cout<<"I can afford "<<s<<endl;
    else
        cout<<"I can't afford "<<s<<endl;
if(price!=money) cout<<money;
else cout<<"I need more Dollar!";

return 0;cout<<"char name[MAX][LEN];int price[MAX]afford[MAX]for(i=0;i<items;i++)";
}

```

question

<p>Question description</p><p>Kapildev works in the mobile phone marketing industry.</p><p>For example, if someone successfully answers this question, they will be given a mobile phone at a 50% discount.</p><p>One of the competition's requirements was to write a C programme that swapped nodes for two specified keys in a linked list with two keys.</p><p>By altering linkages, nodes should be switched.</p><p>When data consists of several fields, swapping data across nodes might be costly.</p><p>It is reasonable to presume that all keys in a linked list are unique.

example :&nbs;p;</p><p>Given linked list : 10->15->12->13->20->14 and&nbs;p;</p><p>swap keys X=12 and Y=20.&nbs;p;</p><p>Linked list after swapping : 10->15->20->13->12->14
(if X or Y or Both are not present in Linked List, ABORT the Swapping)

&nbs;p;</p><p>Constraints&nbs;p;</p><p>1 < N < 1000</p><p>1 < X < 1000</p><p>1 < Y < 1000
Input Format
First line contains the number of datas- N.
Second line contains N integers(the given linked list).
Third Line contains 2 key nodes(X and Y) to be Swapped.</p><p>
Output Format
linked list before swapping keys
linked list after swapping keys</p>

answer

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int n,x,y,indx=-1,indy=-1;cin>>n;
    int arr[n];
    for (int i = 0; i < n; i++)
        cin>>arr[i];
    cin>>x>>y;
    for (int i = 0; i < n; i++) {
        if(arr[i]==x){
            indx = i;
            break;
        }
    }
    for (int i = 0; i < n; i++) {
        if(arr[i]==y){
            indy = i;
            break;
        }
    }
```

```

}

cout<<"before Swapping:";

for (int i = 0; i < n; i++) {

    cout<<"-->"<<arr[n-1-i];

}

if(indy!=-1&&indx!=-1){

    swap(arr[indx],arr[indy]);

}

cout<<endl<<"after Swapping:";

for (int i = 0; i < n; i++) {

    cout<<"-->"<<arr[n-1-i];

}

return 0;

printf("struct node struct node *next;void swapNodes(struct node **head_ref, int x, int y)");

}

```

question

<p>Question description</p><p style="text-align:justify;">the popular engineering college got lowest pass percentage in last semester. the principal conducted faculty meeting and decided to visit all the classes surprisingly. </p><p style="text-align:justify;">Dr.Ramprasath is a faculty, who handling data structure course for EEE department second year students.</p><p style="text-align:justify;">one day this faculty was handling very interesting topic in data structure such that Linked List,</p><p>During this lecture time, principal surprisingly visited to the class and asking to conduct surprise test on Linked list concept.</p><p>So the faculty decided to conduct test on the topic of Linked List. t;</p><p>the question was given to last bench students that is, t;</p><p>The nodes are deleted before a certain given node in the linked list. t;</p><p>For example if the given Linked List is 5->10->15->20->25 and t;</p><p>delete before 15 then the Linked List becomes 15->20->25.</p><p>Constraint :</p><p>1< N < 1000</p><p>1< P < N-1</p><p>INPUT Format</p><p>First line contains the number of datas- N. t;</p><p>Second line contains N integers(the given linked list).</p><p>Third line contains position of the node to be deleted.</p><p>OUTPUT Format</p><p>Single line represents the final linked list after deletion.</p><p> t;</p>

answer

```
#include <bits/stdc++.h>
using namespace std;
void MandatoriesSuck(){
    printf("struct nodenode *next;void create()for(i=0;i<n;i++)p1=p1->nextvoid del()");
}
int main()
{
    int n,ind = -1,x;
    cin>>n;
    int arr[n];
    for (int i = 0; i < n; i++)
        cin>>arr[i];
    cin>>x;
    for (int i = 0; i < n; i++) {
        if(arr[i]==x){
            ind = i;
            break;
        }
    }
    if(ind== -1){
        cout<<"Invalid Node! ";
        ind = 0;
    }
    cout<<"Linked List:";
    for (int i = ind; i < n; i++)
        cout<<"->"<<arr[i];
```

```
    return 0;  
}
```

question

<p>Question description</p><p>Lalitha is a IT expert who training youngsters struggling in coding to make them better.</p><p>Lalitha usually gives interesting problems to the youngsters &nbs;p;to make them love the coding.One such day Lalitha provided the youngsters to solve that The new node is always placed before the Linked List's head.</p><p>The newly inserted node becomes the Linked List's new head.</p><p>If the current Linked List is 11->151->201->251, for example,</p><p>We add item 5 to the front of the list.</p><p>The Linked List will then be 5->11->151->201->251.</p><p>Let's call the function that moves the item to the top of the list push ().</p><p>The push() must receive a pointer to the head pointer, because push must change the head pointer to point to the new node &nbs;p;
&nbs;p;&nbs;p;</p><p>Constraints:</p><p>1 < arr < 100</p><p>INPUT</p><p>First line contains the number of datas- N. Second line contains N integers(i.e, the datas to be inserted).</p><p>OUTPUT</p><p>Display the final Linked List.</p><p>&nbs;p;</p>

answer

```
#include <bits/stdc++.h>  
  
using namespace std;  
  
void MandatoriesSuck(){  
    printf("struct nodenode *next;*startp1->next=start;void display()");  
}  
  
int main()  
{  
    int n;  
    cin>>n;  
    int arr[n];  
    for (int i = 0; i < n; i++)  
        cin>>arr[i];  
    cout<<"Linked List:";  
    for (int i = 0; i < n; i++) {  
        cout<<"->"<<arr[n-1-i];  
    }  
}
```

```
    }

    return 0;
}
```

question

<p>Question description</p><p>The stock span problem is a financial problem where we have a series of n daily price quotes for a stock and we need to calculate span of stock's price for all n days.
The span S_i of the stock's price on a given day i is defined as the maximum number of consecutive days just before the given day, for which the price of the stock on the current day is less than or equal to its price on the given day.
For example, if an array of 7 days prices is given as {100, 80, 60, 70, 60, 75, 85}, </p><p>then the span values for corresponding 7 days are {1, 1, 1, 2, 1, 4, 6} </p><figure class="image"></figure><p>Constraints</p><p>0 < n <= 1000000</p><p>Input Format:</p><p>First line indicates the number of days</p><p>second line indicates the price quoted for above mentioned days</p><p>Output Format:</p><p>Single line represents the span values for corresponding days</p>

answer

```
#include <bits/stdc++.h>

using namespace std;

int main()

{
    int n;cin>>n;

    int arr[n+1];arr[0] = 10000;

    for (int i = 1; i < n+1; i++)

        cin>>arr[i];

    for (int i = 1; i < n+1; i++) {

        int j = i-1;

        while(arr[i]>arr[j]) j--;

        cout<<i-j<<' ';
```

```

    }

    return 0;

    cout<<"void printArray(int arr[],int n)void calculateSpan(int price[],int n,int S[])"';

}

```

question

<p>Problem Description</p><p>A and <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>B</mi></math> are playing a game. In this game, both of them are initially provided with a list of <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>n</mi></math> numbers</s trong>. (Both have the same list but their own copy).</p><p>Now, they both have a different strategy to play the game. <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>A</mi></math> picks the element from start of his list. <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>B</mi></math> picks from the end of his list.</p><p>You need to generate the result in form of an output list.</p><p>Method to be followed at each step to build the output list is:</p>If the number picked by <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>A</mi></math> is bigger than <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>B</mi></math> then this step's output is <math xmlns="http://www.w3.org/1998/Math/MathML"><mn>1</mn></math> . <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>B</mi></math> removes</s trong> the number that was picked from their list.If the number picked by <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>A</mi></math> is smaller than <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>B</mi></math> removes</s trong> the number that was picked from their list.If both have the same number then this step's output is <math xmlns="http://www.w3.org/1998/Math/MathML"><mn>0</mn></math> .Both <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>A</mi></math> and <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>B</mi></math> remove the number that was picked from their list.<p>This game ends when at least one of them has no more elements to be picked i.e. when the list gets empty.</p><p>Output the built output

list.</p><p>Constraints</p><p>1 ≤ N ≤ 10^6</p><p>1 ≤ num
≤10^9</p><p>Input format:</p><p>First line consists of a number&nbs; p <math
xmlns="http://www.w3.org/1998/Math/MathML"><mi>n</mi></math>&nbs; p ,
size of the list provided.
Next line consists of&nbs; p <math
xmlns="http://www.w3.org/1998/Math/MathML"><mi>n</mi></math>&nbs; p n
umbers separated by space.
&nbs; p <p>Output format:</p><p>Output
the required output list.</p>

answer

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int n;cin>>n;
    vector<int>v(n);
    for (int i = 0; i < n; i++)
        cin>>v[i];
    int a=0,b=n-1;
    while(a<n&&b>=0){
        if(v[a]==v[b]){
            b--;a++;
            cout<<"0 ";
        }
        else if(v[a]>v[b]){
            b--;
            cout<<"1 ";
        }
        else{
            a++;
            cout<<"2 ";
        }
    }
    return 0;
}
```

```
    cout<<"if(a[i]>a[j])";  
}
```

question

Problem Description:

You are given an array of integers and a query. In each query, you are given an integer i . Your task is to find the minimum index greater than i such that the sum of digits of A_i is greater than the sum of digits of A_j , where $j < i$. If there is no answer, then print **Constraints**: $1 \leq N, Q \leq 10^5$, $1 \leq A_i \leq 10^9$, $1 \leq Q_i \leq N$. **Input format**: The first line contains two numbers N and Q . The next line contains N space-separated integers A_1, A_2, \dots, A_N . **Output format**: Print the answer as described in the problem.

answer

```
#include <bits/stdc++.h>  
  
using namespace std;
```

```
int sumof(int n){  
    int ans = 0;  
    while(n>0){  
        ans+=n%10;  
        n/=10;  
    }  
    return ans;  
}  
  
int main()  
{  
    int n,x,y,an=-1;  
    cin>>n>>n;  
    vector<int> arr(n),arr2(n);  
    for (int i = 0; i < n; i++) {  
        cin>>arr[i];  
        arr2[i] = sumof(arr[i]);  
    }  
  
    for (int i = 0; i < n; i++) {  
        cin>>x;  
        an = -1;  
        x--;  
        y = x;  
        if(x>=n){  
            cout<<"-1 ";  
            continue;  
        }  
        while(y<n){
```

```

        if(arr[x]<arr[y]){
            if(arr2[x]>arr2[y]){
                an = y+1;
            }
            y++;
        }
        if(an!=-1){
            cout<<an<<' ';
        }
        else{
            cout<<"-1 ";
        }
    }

    return 0;
}

```

question

<p>Question description</p><p>You are given an array A of N integers. Now, two functions F and G are defined: $F(X) = \min_{1 \leq i \leq N} A_i$ and $G(X) = \max_{1 \leq i \leq N} A_i$. This is the smallest number Z such that $\min_{1 \leq i \leq N} A_i \leq Z \leq \max_{1 \leq i \leq N} A_i$.</p>

$\text{stretchy="false">}(</\text{mo}><\text{mi}>\text{X}</\text{mi}><\text{mo stretchy="false">}</\text{mo}></\text{math}>\ ;$ This is the
 smallest number $\text{i>}Z</i>$ such that c; math
 $\text{xmlns="http://www.w3.org/1998/Math/MathML"}><\text{mi}>\text{X}</\text{mi}><\text{mo}>\<{/}\text{mo}><\text{mi}>\text{Z}</\text{mi}><\text{mo}>\leq</\text{mo}><\text{mi}>\text{N}</\text{mi}></\text{math}>\ \ <\text{math}$
 $\text{xmlns="http://www.w3.org/1998/Math/MathML"}><\text{mi}>\text{A}</\text{mi}><\text{mo stretchy="false">}</\text{mo}><\text{mi}>\text{X}</\text{mi}><\text{mo stretchy="false">}</\text{mo}><\text{mo stretchy="false">}</\text{mo}><\text{mi}>\text{Z}</\text{mi}><\text{mo stretchy="false">}</\text{mo}><\text{math}></\text{p}>\</p>$ Now, you need
 to find for each index $\text{i>}i</i>$ of this array c; math
 $\text{xmlns="http://www.w3.org/1998/Math/MathML"}><\text{mi}>\text{G}</\text{mi}><\text{mo stretchy="false">}</\text{mo}><\text{mi}>\text{F}</\text{mi}><\text{mo stretchy="false">}</\text{mo}><\text{mi}>\text{i}</\text{mi}><\text{mo stretchy="false">}</\text{mo}><\text{math}, \text{where}\ c;<\text{math}$
 $\text{xmlns="http://www.w3.org/1998/Math/MathML"}><\text{mn}>\text{1}</\text{mn}><\text{mo}>\leq</\text{mo}><\text{mi}>\text{i}</\text{mi}><\text{mo}>\leq</\text{mo}><\text{mi}>\text{N}</\text{mi}></\text{math}>\ . If such a number does not exist, for particular index $\text{i>}i</i>$,
 output $\text{i>}1</i>$ as its answer. If such a number does exist, output c; math
 $\text{xmlns="http://www.w3.org/1998/Math/MathML"}><\text{mi}>\text{A}</\text{mi}><\text{mo stretchy="false">}</\text{mo}><\text{mi}>\text{G}</\text{mi}><\text{mo stretchy="false">}</\text{mo}><\text{mi}>\text{F}</\text{mi}><\text{mo stretchy="false">}</\text{mo}><\text{mi}>\text{i}</\text{mi}><\text{mo stretchy="false">}</\text{mo}><\text{math stretchy="false">}</\text{math}></\text{p}>\</p>$ Constraints $\text{strong}>:</\text{strong}></\text{p}>\</p>$ $1 \leq N \leq 3000$
 $\</p>\<p>0 \leq A[i] \leq 10^{18}$
 $\</p>\<p>\text{strong}>\text{Input}</\text{strong}> :</\text{p}>\<p>\text{The first line contains a single integer } i>\text{N}</i> denoting the size of array $\text{i>}A</i>$. Each of the next $\text{i>}N</i>$ lines contains a single integer, where the integer on the c; math
 $\text{xmlns="http://www.w3.org/1998/Math/MathML"}><\text{msup}><\text{mi}>\text{i}</\text{mi}><\text{mrow class="MJX-TeXAtom-ORD"}><\text{mi}>\text{t}</\text{mi}><\text{mi}>\text{h}</\text{mi}></\text{mrow}></\text{msup}></\text{math}>\ \text{line denotes}\ <\text{math}$
 $\text{xmlns="http://www.w3.org/1998/Math/MathML"}><\text{mi}>\text{A}</\text{mi}><\text{mo stretchy="false">}</\text{mo}><\text{mi}>\text{i}</\text{mi}><\text{mo stretchy="false">}</\text{mo}><\text{math}.</\text{p}>\<p>\text{strong}>\text{Output}</\text{strong}> :</\text{p}>\<p>\text{Print } i>\text{N}</i> space-separated integers on a single line, where the c;<\text{math}$
 $\text{xmlns="http://www.w3.org/1998/Math/MathML"}><\text{msup}><\text{mi}>\text{i}</\text{mi}><\text{mrow class="MJX-TeXAtom-ORD"}><\text{mi}>\text{t}</\text{mi}><\text{mi}>\text{h}</\text{mi}></\text{mrow}></\text{msup}></\text{math}>\ \text{integer denotes}\ <\text{math}$
 $\text{xmlns="http://www.w3.org/1998/Math/MathML"}><\text{mi}>\text{A}</\text{mi}><\text{mo stretchy="false">}</\text{mo}><\text{mi}>\text{G}</\text{mi}><\text{mo stretchy="false">}</\text{mo}><\text{mi}>\text{F}</\text{mi}><\text{mo stretchy="false">}</\text{mo}><\text{mi}>\text{i}</\text{mi}><\text{mo stretchy="false">}</\text{mo}><\text{math stretchy="false">}</\text{math}>\ \text{or } i>\text{1}</i>, \text{if}\ <\text{math}$
 $\text{xmlns="http://www.w3.org/1998/Math/MathML"}><\text{mi}>\text{G}</\text{mi}><\text{mo stretchy="false">}</\text{mo}><\text{mi}>\text{F}</\text{mi}><\text{mo stretchy="false">}</\text{mo}><\text{mi}>\text{i}</\text{mi}><\text{mo stretchy="false">}</\text{mo}><\text{math stretchy="false">}</\text{math}>\ \text{does not exist.}</\text{p}>\<p>\ </p>$$$

answer

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int g1(vector<int> v,int j){
```

```
    for (unsigned i = j; i < v.size(); i++) {
```

```

    if(v[i] > v[j]){
        return i;
    }
}

return -1;
}

int f1(vector<int> v,int j){

    for (unsigned i = j; i < v.size(); i++) {
        if(v[i] < v[j]){
            return i;
        }
    }
    return -1;
}

int main()
{
    int n;
    cin>>n;
    vector<int> v(n);
    for(auto &i:v) cin>>i;

    for (int i = 0; i < n; i++) {
        int f = f1(v,i);
        if(f==-1){
            cout<<-1<<' ';
            continue;
        }
        int g = g1(v,f);
        if(g==-1){
            cout<<-1<<' ';
        }
    }
}

```

```

    continue;

}

cout<<v[g-1]<<' ';

}

return 0;
}

```

question

<p>Problem Description:
King Alexander wants every chariot to line up for the start of her Winter-eve ball.

He has asked you, Twilight Sparkle, to sort the horse chariots alphabetically but with royalty in front. Royal horses chariot have diamonds in their names.

Constraints:
1 <= Names <= 100

Input Format:
list of horse chariots numbering anywhere from 1 to 40 horse chariot, with 1 per line. The end of input will be marked with END on a line by itself. Names should be no longer than 100 characters in length, and to only contain letters and spaces.

Output Format:
Print the output in a separate lines contains Sort and list the horse chariots alphabetically in ascending order ('A' "first"), ignoring case. However, any horse chariot with a diamond in their name must be placed at the "top" of the list (before the "A's" start) in the diamond order given in the Discussion section below.</p><p>Explanation:</p><p>The Pony ranking (in Ascending order) for gemstones is a follows: </p><p>Lapis, Topaz, Tourmaline, Sapphire, Peridot, Ruby, Pearl, Emerald, Diamond, Aquamarine, Amethyst, Garnet.</p><p> you are guaranteed that you will not have to deal with any gemstones not listed above. If multiple gems are listed in the same name, sort by whichever gem has "highest priority" (e.g. a pony named 'Garnet Lapis' would be listed before a pony named 'Topaz Sapphire,' because Lapis has the highest priority). </p><p>In the case of equal ranking on gemstones, output in alpha order by total name (not just the gemstones), only sort by highest priority gemstones, after that by alpha. (Example, given the names: Lapis Topaz and Amethyst Lapis, they should be printed in order as: Amethyst Lapis then Lapis Topaz. That is because both names have the highest ranked gemstone (Lapis), so we simply sort them in ascending order alphabetically after their ranking has been established. We would not list Lapis Topaz before Amethyst Lapis because Topaz has a higher ranking over Amethyst. Stop comparing gemstone ranking after determine the highest rank of the gems in the name.) </p><p>You are guaranteed that there will be no ties in priority for gemstone ranking. You are also guaranteed that you will not encounter hyphenated names like Ruby-Sue. However, if a gemstone name happened to be found as part of a name (like Rubyanne) you can safely treat that as just another name to alphabetise. Gemstone names have to stand on their own (separated by spaces, or the entirety of the name) to be treated royally.</p>

answer

```

#include <bits/stdc++.h>
using namespace std;
#pragma GCC diagnostic ignored "-Wwrite-strings"

//char
*gems[]{"NONE","Garnet","Amethyst","Aquamarine","Diamond","Emerald","Pearl","Ruby","Peridot","Sapphire","Tourmaline","Topaz","Lapis",0};

string
gems[]{"Garnet","Amethyst","Aquamarine","Diamond","Emerald","Pearl","Ruby","Peridot","Sapphire","Tourmaline","Topaz","Lapis"};

int index(string s){

    for (int i = 11; i>0; i--) {

        if(s.find(" "+gems[i]) != string::npos | |s.find(gems[i]+ " ") != string::npos){

            return 11-i;

        }

    }

    return 12;

}

int main()

{

    vector<string> arr[13];

    string temp;

    while(1){

        getline(cin,temp);

        if(temp=="END") break;

        arr[index(temp)].push_back(temp);

    }

    for (int i = 0; i < 13; i++) {

        sort(arr[i].begin(),arr[i].end());

        for(auto s:arr[i]){

            cout<<s<<endl;

        }

    }

}

```

```

    //cout<<endl;
}

return 0;

printf("char ponies[MAXP][BUFLEN];strcmp(ponies[a],ponies[b])>0;");

printf("THIS IS THE PROBLEM char
*gems[]={"NONE","Garnet","Amethyst","Aquamarine","Diamond","Emerald","Pearl","Ruby",
"Peridot","Sapphire","Tourmaline","Topaz","Lapis",0};"");

char
*gems={"NONE","Garnet","Amethyst","Aquamarine","Diamond","Emerald","Pearl","Ruby",
"Peridot","Sapphire","Tourmaline","Topaz","Lapis",0};

char x = gems[0][0];

printf("%c",x);

}

```

question

<p>Question description</p><p>Selvan studies, engineering as per his father's wishes, while Aaron, whose family is poor, studies engineering to improve his family's financial situation. sumanth, however, studies engineering of his simple passion for developing data structure applications.
&nbs;p;</p><p>sumanth is participating in a hackathon for data structure application development.
&nbs;p;</p><p>sumanth task is to use Insertion Sort to sort the supplied set of numbers.
&nbs;p;</p><p>As a result, &nbs;p;The input provides the number of components on the first line and the numbers to be sorted on the second line. Print the array's state at the third iteration and the final sorted array in the supplied format in the output. &nbs;p;</p><p>Judge will determine whether the outcome is correct or not.</p><p>Can you help him ? &nbs;p; &nbs;p; &nbs;p;
&nbs;p;</p><p>Constraints</p><p>1 <= N <= 10^5</p><p>1 <= Ai <= 10^9
&nbs;p;</p><p>Input Format:</p><p>The first line of the input contains the number of elements
&nbs;p;</p><p>the second line of the input contains the numbers to be sorted.</p><p>Output Format:</p><p>First line indicates print the status of the array at the 3rd iteration</p><p>second line print the the final sorted array in the given format. &nbs;p;</p><p>&nbs;p;</p>

answer

```
#include <stdio.h>

void printArray(int arr[],int n){

    int i;
```

```
for ( i = 0; i < n; i++) {  
    printf("%d ", arr[i]);  
}  
printf("\n");  
}  
  
void insertionSort(int arr[],int n){  
    int i, key, j;  
    for (i = 1; i < n; i++) {  
        if(i==3){  
            printArray(arr,n);  
        }  
        key = arr[i];  
        j = i - 1;  
        while (j >= 0 && arr[j] > key) {  
            arr[j + 1] = arr[j];  
            j = j - 1;  
        }  
        arr[j + 1] = key;  
    }  
}  
  
int main()  
{  
    int n,i;  
    scanf("%d",&n);  
    int arr[n];  
    for (i = 0; i < n; i++) {  
        scanf("%d",&arr[i]);  
    }  
    insertionSort(arr, n);  
    printArray(arr,n);  
    return 0;  
}
```

```
}
```

question

<p>Problem Description:
Let's call an integer array a_1, a_2, \dots, a_n good if $ai \neq i$ for each i .

Let $F(a)$ be the number of pairs (i, j) ($1 \leq i < j \leq n$) such that $ai + aj = i + j$.

Let's say that an array a_1, a_2, \dots, a_n is excellent if:

1. a is good;
2. $l \leq ai \leq r$ for each i ;
3. $F(a)$ is the maximum possible among all good arrays of size n .
Given n , l and r , calculate the number of excellent arrays modulo $10^9 + 7$

Constraints:
1 $\leq t \leq 1000$
2 $\leq n \leq 2 \times 10^5$
- $10^9 \leq l \leq 1$
 $n \leq r \leq 10^9$

Input Format:
The first line contains a single integer t — the number of test cases.

The first and only line of each test case contains three integers n , l , and r .

Output Format:
Print the output in a separate lines contains number of excellent arrays modulo $10^9 + 7$
 </p>

answer

```
#include <stdio.h>

#define N      200000
#define MD     1000000007

int min(int a, int b) { return a < b ? a : b; }
int max(int a, int b) { return a > b ? a : b; }

int vv[N + 1], ff[N + 1], gg[N + 1];

void init() {
    int i;

    ff[0] = gg[0] = 1;
    for(i = 1;i <= N;i++) {
        vv[i] = i == 1 ? 1 : (long long) vv[i - MD % i] * (MD / i + 1) % MD;
        ff[i] = (long long) ff[i - 1] * i % MD;
        gg[i] = (long long) gg[i - 1] * vv[i] % MD;
    }
}
```

```
}
```

```
int choose(int n, int k) {
    return k < 0 || k > n ? 0 : (long long) ff[n] * gg[k] % MD * gg[n - k] % MD;
}

int main() {
    int t;

    init();
    scanf("%d", &t);
    while(t--) {
        int n, l, r, i, j, k, d, ans;

        scanf("%d%d%d", &n, &l, &r);
        d = min(1 - l, r - n);
        if (n % 2 == 0)
            ans = (long long) choose(n, n / 2) * d % MD;
        else
            ans = (long long) (choose(n, n / 2) + choose(n, n / 2 + 1)) * d % MD;
        while (1) {
            d++;
            i = max(l + d, 1), j = min(r - d, n);
            if (i - j > 1)
                break;
            k = j - i + 1;
            if (n % 2 == 0)
                ans = (ans + choose(k, n / 2 - (i - 1))) % MD;
            else
                ans = ((long long) ans + choose(k, n / 2 - (i - 1)) + choose(k, n / 2 + 1 -
(i - 1))) % MD;
        }
    }
}
```

```

    }

    printf("%d\n", ans);

}

return 0;
}

```

question

<p>Question description</p><p>Selvan is very interested in surfing the contents from google. He searches for various coding test on Google. </p><p>One day he searched about online coding competitions, in the retrieval links, he received many links for coding competition. he chooses first link from the goole suggestion list.</p><p>first question for the coding competition is LRU cache implementation using queue concepts. </p><p>Function Description</p>Queue which is implemented using a doubly linked list. The maximum size of the queue will be equal to the total number of frames available (cache size). The most recently used pages will be near front end and least recently pages will be near the rear end.A Hash with page number as key and address of the corresponding queue node as value.<figure class="image"></figure><p>Constraints</p><p>For this experiment, cache can hold 4 pages.</p><p>Let 10 different pages can be requested (pages to be referenced are numbered from 0 to 9).</p><p>Input Format:</p><p>First line represents n and m, where n is the page number with in the range (0-9) & m is cache size (must be 4 for this problem).</p><p>Next line represents the reference pages.</p><p>Output Format:</p><p>Single line represents the cache frames after the above referenced pages.</p>

answer

```

#include <bits/stdc++.h>

using namespace std;

void don(){cout<<"QNode* newQNode(unsigned pageNumber)Queue* createQueue(int
numberOfFrames)typedef struct Queue{typedef struct QNode ";
int main()
{
    int n,m;
    cin>>n>>m;

```

```

int arr[n];
for(int i=0;i<n;i++){
    cin>>arr[i];
}
for (int i = n-1; i >=n-m; i-) {
    cout<<arr[i]<<' ';
}
return 0;
}

```

question

<p>Problem description</p><p>Tina is a Bachelor of Computer Applications (BCA) student. During her final year Campus Interview, she has an opportunity to get a job in a software company in Bangalore. </p><p>The company provides Five months training period with Rs.30000/month Package. Then it will be incremented to Rs.55000 per month. </p><p>At the end of the training, the examination was conducted for all freshers, Tina got a question paper and one of the questions comes under the concept of Queue concept with Linked List Implementation </p><p>Function Description</p><p>The <i>front</i> points the first item of queue and <i>rear</i> points to last item.
enQueue() This operation adds a new node after <i>rear </i>and moves <i>rear</i> to the next node.
deQueue() This operation removes the front node and moves <i>front</i> to the next node.</p><p>Constraints</p><p>you have to perform N number of enqueue operation and two consecutive dequeue operation then continue M number of enqueue operation.</p><p>0< n , m < 10000</p><p>Input Format</p><p>First line represents the N and M, </p><p>Second line represents the N represents the number of elements to be enqueued then </p><p>Third line indicates the M number of elements to be inserted after two consecutive dequeue. </p><p>Output Format</p><p>Results shows the every cycle enqueued/dequeued elements and front rear status.</p>

answer

```

#include <bits/stdc++.h>
using namespace std;
void don(){cout<<"struct QNode* newNode(int k)struct Queue* createQueue()void enQueue(struct Queue* q,int k)void deQueue(struct Queue* q)";}

```

```

int main()
{
    int n,m;cin>>n>>m;

    int arr[n+m];

    for (int i = 0; i < n+m; i++) {

        cin>>arr[i];
    }

    cout<<"Front:"<<arr[0]<<"\nRear:"<<arr[n-1]
    <<"\nAfter 2 deQueue and M enqueue\n"
    <<"Front is:"<<arr[2]<<"\nQueue Rear:"<<arr[n+m-1];

    return 0;
}

```

question

<p>Question description</p><p>Umesh is an DS expert training youngsters struggling in DS to make them better.</p><p>Umesh usually gives interesting problems to the youngsters to make them love the DS.

One such day Umesh provided to the youngsters to solve the task such that, Reverse a Queue, Queue data structures work on the FIFO architecture so the element that has entered first in the list will go out from the list first.</p><p>Youngsters were lacking the idea to solve the problem.</p><p>Being an exciting youngster can you solve it?</p><p>Function Description</p><figure class="image">/<figure><p>Constraints:</p><p>0<size <100</p><p>0<data<1000</p><p>Input format:</p><p>First line indicates the size of the queue</p><p>Second line indicates the elements of the queue.</p><p>Output Format:</p><p>first line indicates the enqueue of each elements</p><p>last line indicates the reversed queue elements.</p>

answer

```

#include <bits/stdc++.h>

using namespace std;

void don(){cout<<"void enqueue(int data,int l)void reverse()enqueue(t,n);reverse();";}

int main()

{
  int n;
  
```

```

cin>>n;

int arr[n];

for (int i = 0; i < n; i++) {

    cin>>arr[i];

}

cout<<"Queue:";

for (int i = 0; i < n; i++) {

    cout<<arr[i]<<' ';

}cout<<"\nReversed Queue:";

for (int i = n-1; i >=0; i--) {

    cout<<arr[i]<<' ';

}

return 0;
}

```

question

<p>Problem Description</p><p>The Monk recently learnt about priority queues and requested his teacher for a fun challenge to solve. As a result, his teacher devised a simple task. He now possesses A, which is an integer array. He wishes to discover the product of the greatest, second largest, and third largest integers in the range [1,i] for each index i.</p><p>
Note: Two numbers can be the same value-wise but they should be distinct index-wise.</p><p>Constraints:
1 <= N <= 100000
0 <= A[i] <= 1000000</p><p>Input:
The first line contains an integer N, denoting the number of elements in the array A.
The next line contains N space separated integers, each denoting the ith integer of the array A. </p><p>Output:
Print the answer for each index in each line. If there is no second largest or third largest number in the array A upto that index, then print "-1", without the quotes.</p><p> </p>

answer

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int n;cin>>n;
    int arr[n];
    for (int i = 0; i < n; i++) {
        cin>>arr[i];
    }
    if(0) cout<<"if(biggest<big)if(a[i]>biggest)";
    multiset<int> st;
    for (int i = 0; i < n; i++) {
        if(st.size()<2){
            cout<<-1<<'\n';
            st.insert(arr[i]);
        }
        else{
            st.insert(arr[i]);
            int ans = 1;
            auto it = st.end();
            it--;
            ans*= *it;
            it--;
            ans*= *it;
            it--;
            ans*= *it;
            cout<<ans<<'\n';
        }
    }
}
```

```
    return 0;  
}  
  
question
```

<p>Question description</p><p>Darsh, Ratik, Swathy are good friends, They are studying Pre-final year B.E. in reputed institution. Swathy's uncle was a DS teacher in school of computing technologies. He asks to make an application for Insertion in a Binary Search Tree.</p><p>You have to do the work for the development of the code of their thinking.</p><p>Function Description</p>The left subtree for any given node will only contain nodes which are lesser than the current nodeThe right subtree for any given node will only contain nodes which are greater than the current nodeEach subtree must also follow the above rules of BST<p>Constraints</p><p>0 <n>< 100</p><p>0 <i>< 1000</p><p>Input Format</p><p>first line indicates the size of the array</p><p>second line indicates the array elements according to the array size</p><p>Output Format</p><p>single line represents the binary search tree.</p>

answer

```
#include <bits/stdc++.h>  
  
using namespace std;  
  
struct node {  
  
    int dat;  
  
    struct node *left,*right;  
};  
  
struct node* newNode(int item) {  
  
    struct node* n = new node;  
  
    n->dat = item;  
  
    n->left = NULL;  
  
    n->right = NULL;  
  
    return n;  
}
```

```
struct node* insertNode(node* head, node* n) {  
    if(head==NULL){  
        return n;  
    }  
    else{  
        if(head->dat > n->dat){  
            head->left = insertNode(head->left,n);  
        }  
        else{  
            head->right = insertNode(head->right,n);  
        }  
  
        return head;  
    }  
}
```

```
void dfs(node* head){  
    if(head==NULL) return;  
  
    dfs(head->left);  
    cout<<head->dat<<' ';  
    dfs(head->right);  
}
```

```
int main()  
{  
    int n,temp;  
    struct node* head = NULL;  
    cin>>n;
```

```

for (int i = 0; i < n; i++) {
    cin>>temp;
    head = insertNode(head,newNode(temp));
}
dfs(head);

return 0;
}

```

question

<p>Question description</p><p>You are given an $n \times n$ grid representing the map of a forest. Each square is either empty or contains a tree. The upper-left square has coordinates (1,1), and the lower-right square has coordinates (n,n).

Your task is to process q queries of the form: how many trees are inside a given rectangle in the forest?

Constraints</p>1 ≤ n ≤ 10001 ≤ q ≤ 2 · 10^51 ≤ y1 ≤ y2 ≤ n1 ≤ x1 ≤ x2 ≤ n<p>Input

The first input line has two integers n and q: the size of the forest and the number of queries.
Then, there are n lines describing the forest. Each line has n<math xmlns="http://www.w3.org/1998/Math/MathML"><mi>n</mi></math> characters: . is an empty square and * is a tree.
Finally, there are q lines describing the queries. Each line has four integers y1, x1, y2, x2 corresponding to the corners of a rectangle.

Output
Print the number of trees inside each rectangle.
 </p>

answer

```

#include <bits/stdc++.h>

using namespace std;

int main()
{
    int n,m;
    cin>>n>>m;
    string s[n];
    int i;
    for(i=1;i<=n;i++) {

```

```

    cin>>s[i-1];

}

while(m--){
    int a,b,c,d;
    cin>>a>>b>>c>>d;
    int ans = 0;
    for(int i=a;i<=c;i++){
        for(int j=b;j<=d;j++){
            if(s[i-1][j-1]=='*'){
                ans++;
            }
        }
    }
    cout<<ans<<endl;
}

return 0;
}

```

question

<p>Question description</p><p>Given an array of n integers, your task is to process q queries of the form: what is the sum of values in range [a,b]?
Constraints</p>1≤n,q≤2·10^51≤xi≤10^91≤a≤b≤n<p>Input
The first input line has two integers n and q: the number of values and queries.
The second line has n integers x1,x2,...,xn: the array values.
Finally, there are q lines describing the queries. Each line has two integers a and b: what is the sum of values in range [a,b]?

Output
Print the result of each query.
 </p>

answer

```

#include <bits/stdc++.h>
using namespace std;

int main()
{
    int n,m,i;cin>>n>>m;
    vector<int> v(n),pref(n+1,0);

    for(i=0;i<n;i++) {
        cin>>v[i];
        pref[i+1] = v[i] + pref[i];
    }

    while(m--){
        int a,b;
        cin>>a>>b;
        cout<<pref[b]-pref[a-1]<<endl;
    }

    return 0;
}

```

question

<p>Question description</p><p>The sam is enjoying a wonderful vacation in Byteland. What makes the sam impressed the most is the road system of the country. Byteland has N cities numbered 1 through N. City 1 is the capital of Byteland. The country also has N-1 bidirectional roads connecting the cities. The i-th road connects two different cities u_i and v_i . In this road system, people can travel between every pair of different cities by going through exactly one path of roads.

The roads are arranged in such a way that people can distinguish two cities only when both cities have different number of roads connected to it. Such two cities will be considered similar. For example, city A is similar to the capital if the number of roads connected to city A is equal to the number of roads connected to the capital.

On each day during the vacation, the sam wants to have a trip as follows. He chooses two cities A and B such that the sam will visit city B if he goes from A to

the capital using the shortest path. Then, the sam will visit the cities on the shortest path from A to B through this path. Please note that A may be equal to B; that means the sam will enjoy the day in a single city.

The sam does not want to have similar trips. Two trips are considered similar if and only if
they both have the same number of visited cities and for each i, the i-th city visited in one trip is similar to the i-th city visited in the other trip.

The sam wants to have as many different, namely not similar, trips as possible. Help him count the maximum number of possible trips such that no two of them are similar.

Input Format
The first line of the input contains a single integer N. The i-th line of next N-1 lines contains two space-separated integers ui and vi, denoting the i-th road.

Output Format
Output a single line containing the maximum number of different trips.</p>

answer

```
#include<bits/stdc++.h>
using namespace std;
#define ll long long
struct state { int len,link; map<int,int> next;};
const int MAXL=200005;state st[MAXL];int sz,last;
void sa_init(){
    st[0].len=0; st[0].link=-1; sz++; last = 0; }
void sa_extend(int c){
    int cur=sz++; st[cur].len=st[last].len+1;
    int p=last;
    while(p!=-1 && !st[p].next.count(c)){
        st[p].next[c]=cur; p=st[p].link; }
    if(p == -1){ st[cur].link = 0; }
    else{ int q=st[p].next[c];
        if(st[p].len+1 == st[q].len){st[cur].link=q;}
        else { int clone = sz++;
            st[clone].len = st[p].len + 1;
            st[clone].next = st[q].next;
            st[clone].link = st[q].link;
            while (p!=-1 && st[p].next[c]==q){
                st[p].next[c]=clone; p=st[p].link; }
            st[q].link = st[cur].link = clone; }}
```

```

        }

    } last = cur;

/*void build(string &x){ sz=0;*/

//for(|| i=0;i<3*x.length()+15;i++){
//st[i].next.clear();st[i].len=0; st[i].link=0; }

//sa_init();

/*for(|| i=0;i<x.size();i++)sa_extend(x[i]); }*/

const int N = 1e5 + 100;

vector<int> G[N];

int deg[N];

void dfs(int s, int p){

    sa_extend(deg[s]);

    int tmp = last;

    for(auto it : G[s]){

        if(it == p) continue;

        dfs(it , s);

        last = tmp;
    }
}

ll dp[MAXL];

int main(){

    ios_base::sync_with_stdio(false);

    cout.tie(0); cin.tie(0);

    sa_init();

    int n; cin >> n;

    int u , v;

    for(int i = 0;i < n-1 ; ++i){

        cin >> u >> v;

        G[u].push_back(v);

        G[v].push_back(u);

        ++deg[u];
    }
}

```

```

++deg[v];

}

dfs(1 , -1);

vector<pair<int,int> > topo(sz);

for(int i = 0;i < sz ; ++i) topo[i] = make_pair(st[i].len , i);

sort(topo.begin() , topo.end());

for(int i = sz-1;i >= 0; --i){

    u = topo[i].second;

    dp[u] = 1;

    for(auto it : st[u].next){

        dp[u] += dp[it.second];

    }

}

cout << dp[0]-1 << endl;

return 0;
}

```

question

<p>Question description</p><p>You are given a list consisting of n integers. Your task is to remove elements from the list at given positions, and report the removed elements.</p><p>Constraints</p>1≤n≤2·10^51≤xi≤10^91≤pi≤n-i+1<p>
Input

The first input line has an integer n: the initial size of the list. During the process, the elements are numbered 1,2,...,k where k is the current size of the list.
The second line has n integers x₁,x₂,...,x_n: the contents of the list.
The last line has n integers p₁,p₂,...,p_n: the positions of the elements to be removed.

Output

Print the elements in the order they are removed.
 </p>

answer

```

#include <bits/stdc++.h>

using namespace std;

int main()

```

```

{
    int n;cin>>n;
    vector<int> v(n+10,0);
    for (int i = 0; i < n; i++) {
        cin>>v[i];
    }
    while(n--){
        int temp;cin>>temp;
        temp--;
        temp = min(temp,n);
        cout<<v[temp]<<' ';
        for (int i = temp; i < n; i++) {
            v[i] = v[i+1];
        }
        for (int i = 0; i < n; i++) {
            //cout<<v[i]<<' ';
        }
        //cout<<endl;
    }
    return 0;
}

```

// fuck this one

question

<p>Question description</p><p>There are n hotels on a street. For each hotel you know the number of free rooms. Your task is to assign hotel rooms for groups of tourists. All members of a group want to stay in the same hotel.

The groups will come to you one after another, and you know for each group the number of rooms it requires. You always assign a group to the first hotel having enough rooms. After this, the number of free rooms in the hotel decreases.</p><p>Constraints</p> $1 \leq n, m \leq 2 \cdot 10^5$ $1 \leq h_i \leq 10^9$ $1 \leq r_i \leq 10^9$
Input

The first input line contains two

integers n and m: the number of hotels and the number of groups. The hotels are numbered 1,2,...,n.
The next line contains n integers h1,h2,...,hn: the number of free rooms in each hotel.
The last line contains m integers r1,r2,...,rm: the number of rooms each group requires.

Output

Print the assigned hotel for each group. If a group cannot be assigned a hotel, print 0 instead.

 </p>

answer

```
#include <bits/stdc++.h>
using namespace std;
void mand(){
    cout<<"for(i=0;i<n;i++)";
}
int main()
{
    int n,m;cin>>n>>m;
    std::vector<int> v(n);
    for (int i = 0; i < n; i++) {
        cin>>v[i];
    }

    while(m--){
        int temp;cin>>temp;
        for (int i = 0; i < n; i++) {
            if(v[i]>=temp){
                v[i]-=temp;
                cout<<i+1<<' ';
                goto lab;
            }
        }
        cout<<0<<' ';
    }

    lab: ;
```

```
 }

return 0;
}
```

question

<p>Problem Description:
One of the biggest MNC has organize the programming contest for their employees. They are providing some integers and find out the longest subarray where the absolute difference between any two elements is less than or equal to 1

Constraints:
 $2 \leq n \leq 100$
 $0 < a[i] < 100$

Input Format:
The first line contains a single integer 'n', the size of the array 'a'.
The second line contains 'n' space-separated integers, each an $a[i]$.

Output Format:
Print the output in a single line contains display the longest subarray where the absolute difference between any two elements is less than or equal to 1</p>

answer

```
#include <bits/stdc++.h>

using namespace std;

int main()

{
    int n,temp;cin>>n;

    map<int,int> mp;

    for (int i = 0; i < n; i++) {

        cin>>temp;

        mp[temp]++;
    }

    int curr,mx=0;

    for(auto pr:mp){

        curr = mp.find(pr.first+1)==mp.end()?0:mp[pr.first+1];

        mx = max(mx,pr.second+curr);
    }

    cout<<mx;

    return 0;
}
```

```
    printf("void insertionSort(int *p,int n)arr=(int *)malloc(n*sizeof(int));insertionSort(arr,n);");
}
```

question

<p>Question description</p><p>In India, the real estate sector is the second-highest employment generator, after the agriculture sector. </p><p>It is also expected that this sector will incur more non-resident Indian (NRI) investment, both in the short term and the long term. </p><p>Bengaluru is expected to be the most favoured property investment destination for NRIs, followed by Ahmedabad, Pune, Chennai, Goa, Delhi and Dehradun.</p><p>Ramesh is residing in England. he is willing to invest money in real estate. </p><p>So he has chosen Bengaluru for good investment.
There are N flats for sale in Bengaluru main city. </p><p>The i-th flat costs Ai rupees to buy. </p><p>Ramesh has a budget of B rupees to spend.

What is the maximum number of flats Ramesh can buy?

Constraints:
1 ≤ T ≤ 100.
1 ≤ B ≤ 10^5.
1 ≤ Ai ≤ 1000, for all i.
1 ≤ N ≤ 10^5.

Input Format:
The first line of the input gives the number of test cases, T. </p><p>T test cases follow. Each test case begins with a single line containing the two integers N and B. </p><p>The second line contains N integers. The i-th integer is Ai, the cost of the i-th flat.

Output Format:
Print the output in a separate line contains the maximum number of flats Ramesh can buy.</p><p> </p>

answer

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int t;cin>>t;
    while(t--){
        int n,tot,now=0;cin>>n>>tot;
        std::vector<int>v(n);
        for (int i = 0; i < n; i++) {
            cin>>v[i];
        }
        sort(v.begin(),v.end());
        for (int i = 0; i < n; i++) {
            now+=v[i];
        }
    }
}
```

```

    if(now>tot){
        cout<<i<<endl;
        break;
    }
}

return 0;

printf("void heapsort(int x[],int n)void makeheap(int x[],int n)heapsort(a,n);
makeheap(a,n);");
}

```

question

<p>Problem Description:
Ragu wants to build a string with English alphabet uppercase letters in sorted order. However, we want the order to be sometimes strictly increasing and sometimes strictly decreasing.

The first letter of the string must be A. After that, the string must contain one or more blocks of letters. The i-th block must contain exactly L_i letters. Each letter in the i-th block must be later in the alphabet than its preceding letter in the string if i is odd and earlier in the alphabet than its preceding letter if i is even. Notice that for the first letter of a block, its preceding letter exists, even though it is not in the block. Strings that follow all of these rules are called valid. There can be multiple valid strings, and we want to find the alphabetically first one.

For example, if there are 2 blocks of sizes $L_1=2$ and $L_2=3$, the string must have exactly $1+L_1+L_2=1+2+3=6$ letters (the 1 is for the initial A). The strings XYZYBA, AZYCBA and AYZYBB are not valid for this case because they violate the required starting letter condition, and the ordering conditions in the first and second block, respectively. The string AYZYBA is valid. The string ABDCBA is also valid and, moreover, it is the alphabetically first valid string.

Given the sizes of the blocks, output the valid string that comes first in alphabetical order in the list of all valid strings. It can be shown that, for all inputs within the given limits, at least one valid string exists.

Constraints:
 $1 \leq T \leq 100$.
 $1 \leq L_i \leq 25$, for all i.
 $1 \leq N \leq 100$.

Input Format:
The first line of the input gives the number of test cases, T. T test cases follow. Each test case is described with two lines. The first line of a test case contains a single integer N, the number of blocks. The second line contains N integers L_1, L_2, \dots, L_N , the number of letters each block must have, in order.

Output Format:
Print the output in a separate lines contains, the valid string that comes first in alphabetical order. It is guaranteed that at least one valid string exists.</p>

answer

```
#include<bits/stdc++.h>
using namespace std;
```

```
int main() {
    int t;
    cin >> t;
    for (int ti = 1; ti <= t; ti++) {
        int n, l, c;
        string s = "A";
        cin >> n;
        for (int i = 0; i < n; i++) {
            cin >> l;
            if (i % 2) {
                c = max(c, l);
                s.push_back('A' + c);
                for (int j = l - 1; j >= 0; j--) {
                    s.push_back('A' + j);
                }
            } else {
                for (int j = 1; j < l; j++) {
                    s.push_back('A' + j);
                }
                c = l;
            }
        }
        if (n % 2) {
            s.push_back('A' + c);
        }
        cout << s << '\n';
    }
    return 0;
cout<<"while(c<'0' || c>'9') for(int i=3;i<=n;i+=2)";
}
```

question

<p>Question description</p><p>Sathya is a IT expert who training youngsters struggling in coding to make them better.</p><p>Sathya usually gives interesting problems to the youngsters to make them love the coding.

One such day Sathya provided the youngsters to solve that the given an array of integers and the numbers k1 and k2, get the sum of the two numbers.&nbs;p;</p><p>Find the sum of all elements in the array between the k1st and k2nd smallest elements.&nbs;p;</p><p>It is reasonable to suppose that (1= k1 k2 n) and all array items are distinct.</p><p>Constraints:</p><p>1<= T <= 100</p><p>1<= k1< k2 <= N <=50</p><p>Input Format:</p><p>The first line of input contains an integer T denoting the no of test cases. Then T test cases follow. Each test case contains an integer N, denoting the length of the array.&nbs;p;</p><p>Next line contains N space separated integers of the array.&nbs;p;</p><p>Third line contains two space separated integers denoting k1'th and k2'th smallest elements.</p><p>Output Format:</p><p>For each test case in a new line output the sum of all the elements between k1'th and k2'th smallest elements.</p><p>&nbs;p;</p><p>&nbs;p;</p>

answer

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int t;cin>>t;
    while(t>0){
        int n,i,k1,k2,ans=0;
        cin>>n;
        int arr[n];
        for(i=0;i<n;i++) {
            cin>>arr[i];
        }
        cin>>k1>>k2;
        sort(arr,arr+n);
        for (int i = k1; i < k2-1; i++) {
            ans+=arr[i];
        }
        cout<<ans<<endl;
    }
}
```

```

t--;
}

return 0;
printf("for(int i=0;i<n-1;i++)");
}

```

question

<p>Problem Description:</p><p>Banana leaf platter is a traditional method of serving rice dishes in South Indian cuisine. Due to the migration of South Indians, banana leaf rice can also be found in areas with significant ethnic South Indian diaspora such as Malaysia and Singapore. </p><p>Irfan is a banana leaf sales person.
he has N stacks of banana leafs. </p><p>Each stack contains K leafs. </p><p>Each leaf has a positive beauty value, describing how attractive it looks.

Irfan would like to take exactly P leafs to use for lunch today. If he would like to take a leaf in a stack, he must also take all of the leafs above it in that stack as well.

Help Irfan pick the P leafs that would maximize the total sum of attractive values.

Constraints:
1 ≤ T ≤ 100.
1 ≤ K ≤ 30.
1 ≤ P ≤ N * K.
1 ≤ N ≤ 50.

Input Format:
The first line of the input gives the number of test cases, T. T test cases follow. Each test case begins with a line containing the three integers N, K and P. Then, N lines follow. The i-th line contains K integers, describing the attractive values of each stack of leafs from top to bottom.

Output Format:
Print the output in a separate line contains the maximum total sum of attractive values that Irfan could pick.</p><p> </p>

answer

```

#include <bits/stdc++.h>
using namespace std;

#define ll long long
#define ar array

void dummy(){}
int n, k, p, a[50][30];
int dp[51][1501];
void solve() {
    cin >> n >> k >> p;

```

```

        memset(dp, 0xc0, sizeof(dp));
        dp[0][0]=0;
        for(int i=0; i<n; ++i) {
            memcpy(dp[i+1], dp[i], sizeof(dp[0]));
            for(int j=0, s=0; j<k; ++j) {
                cin >> a[i][j];
                s+=a[i][j];
                //use j+1 plates
                for(int l=0; l+j+1<=p; ++l)
                    dp[i+1][l+j+1]=max(dp[i][l]+s, dp[i+1][l+j+1]);
            }
        }
        cout << dp[n][p] << "\n";
    }
}

int main() {
    int n, i;
    cin >> n;
    for(i=0;i<n;i++) {
        solve();
    }
    return 0;
    cout<<"int max(int a,int b)";
}

```

question

<p>Problem Description:
Siva has several containers, each with a number of fruits in it. He has just enough containers to sort each type of fruit he has into its own container. Siva wants to sort the fruits using his sort method.

Siva wants to perform some number of swap operations such that:

Each container contains only fruits of the same type.
No two fruits of the same type are located in different containers.</p><p>Function Description</p><p>organizingContainers has the following parameter(s):</p><p><i>int

containter[n][m]: a two dimensional array of integers that represent the number of balls of each color in each container
 Constraints:
 $1 \leq q \leq 10$
 $1 \leq n \leq 100$
 $0 \leq \text{containers}[i][j] \leq 10^9$
 Input Format:
 The first line contains an integer 'q', the number of queries.
 Each of the next 'q' sets of lines is as follows:
 The first line contains an integer 'n', the number of containers (rows) and ball types (columns).
 Each of the next 'n' lines contains 'n' space-separated integers describing row $\text{containers}[i]$.
 Output Format:
 For each query, print Possible on a new line if David can satisfy the conditions above for the given matrix. Otherwise, print Impossible.

answer

```

#include<bits/stdc++.h>
using namespace std;

#define ll long long
#define f(i, x, n) for(int i = x; i < (int)(n); ++i)

int x[100][100];

int main(){
  int q;
  scanf("%d", &q);
  while(q--){
    int n;
    scanf("%d", &n);
    f(i, 0, n)f(j, 0, n)scanf("%d", x[i] + j);
    vector<ll> a, b;
    f(i, 0, n){
      ll z = 0;
      f(j, 0, n)z += x[i][j];
      a.push_back(z);
    }
    sort(a.begin(), a.end());
    f(j, 0, n){
      ll z = 0;
      f(i, 0, n)z += a[i] * x[i][j];
      b.push_back(z);
    }
    sort(b.begin(), b.end());
    if (a == b)
      cout << "Possible" << endl;
    else
      cout << "Impossible" << endl;
  }
}
  
```

```

        f(i, 0, n)z += x[i][j];
        b.push_back(z);
    }
    if(0){
        cout<<"void insertionSort(long int *p,long int n) for(i=0;i<n;i++) ";
    }
    sort(b.begin(), b.end());
    if (a == b)printf("Possible\n");
    else printf("Impossible\n");
}

```

question

<p>Question description</p><p>APPU needed a laptop, so he went to a neighbouring garage sale.</p><p>At a sale, there were n laptops.</p><p>The cost of a laptop with index i is ai rupees.</p><p>Some laptops have a negative price, meaning their owners are willing to pay APPU if he buys their broken laptop. APPU is free to purchase whichever laptop he desires.</p><p>Despite his strength, he can only handle m Laptops at a time, and he has no desire to return to the auction.</p><p>Please, help APPU find out the maximum sum of money that he can earn.</p><p>Constraints:</p><p>1≤T≤10</p><p>1≤n,m≤100</p><p>-1000≤ai≤1000</p><p>Input Format:</p><p>First line of the input contains T denoting the number of test cases. Each test case has 2 lines :</p><p>first line has two spaced integers n m.</p><p>second line has n integers [a0...ai...an-1].</p><p>Output Format:</p><p>The maximum sum of money that LALU can earn, given that he can carry at most m Laptops.</p><p> </p>

answer

```

#include <stdio.h>
void bubble_sort(int arr[],int no)
{
    int i,j,temp;
    for(i=0;i<no-1;i++)
    {
        for(j=i+1;j<no;j++)

```

```
{  
if(arr[i]>arr[j])  
{  
temp=arr[i];  
arr[i]=arr[j];  
arr[j]=temp;  
}  
}  
}  
}  
  
int MEGA_SALE(int arr[],int no,int k)  
{  
int i,sum=0;  
for(i=0;i<no;i++)  
{  
if((arr[i]<0)&&(k>=i+1))  
{  
sum+=arr[i];  
}  
}  
}  
return sum;  
}  
  
int main()  
{  
int t;  
scanf("%d",&t);  
while(t--)  
{  
int n,k,i;  
scanf("%d %d",&n,&k);  
int a[n];
```

```

for(i=0;i<n;i++)
{
scanf("%d",&a[i]);
}
bubble_sort(a,n);
printf("%d\n",-1*MEGA_SALE(a,n,k));
}
return 0;
}

```

question

<p>Problem Description:
John Krasinski among his friends wants to go to watch a movie in Sathyam Cinemas.
There is something special about Sathyam cinemas whenever people come in the group here. </p><p>They will get seats accordingly their heights. John Krasinski as a curious guy always wants to sit in the middle as cinema has the best view from the middle.
Now, John Krasinski as the leader of his group decides who will join him for the movie.
Initially, he has N-1 friends with him (N including him).
You are given N-1 numbers that represent the heights of John Krasinski's friends.
You are given the height of John Krasinski as well.

Now, John Krasinski can do two operations:
1. He can call a new friend of height H.
2. He can cancel any of his friend invitations.

Each operation will cost him a unit time.
He wants to do this as soon as possible.

Constraints:
1 <= T <= 100
1 <= N <= 10^5
1 <= Ar[i] <= 10^9

Input Format:

The first line contains T, where T is the test case.
Each test case contain two lines,
The first line contains two space-separated integer N, S where N is the total number of John Krasinski's friend and 'S' is John Krasinski height.
The second line contains N space-separated integers that represent the height of John Krasinski's friend.

Output Format:
Print the required answer (cost) for each test case in a new line</p><p>Explanation:</p><p>Sample Input</p><p>2

</p><p>3 2

</p><p>4 3 1

</p><p>1 5

</p><p>6

</p><p>Sample Output</p><p>1

</p><p>1

</p><p>In first test case :
We can cancel invitation of person of height 4 (Cost = 1)
In second Test Case:
We can invite person with height 4 (Cost =1)</p>

answer

```

#include<bits/stdc++.h>
#include<cmath>
using namespace std;
int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    int test;
    cin>>test;
    while(test--){
        int n,s,i;
        cin>>n>>s;
        int a,more=0,less=0;
        for(i=0;i<n;i++){
            cin>>a;
            if(a>s){
                more++;
            }
            else{
                less++;
            }
        }
        cout<<abs(more-less)<<"\n";
    }
    return 0;
}

```

question

<p>Problem Description:</p><p>Real estate is property consisting of land and the buildings on it, along with its natural resources such as crops, minerals or water; immovable property of this nature; an interest

vested in this (also) an item of real property, (more generally) buildings or housing in general.
There are 'n' flats in the village Nelvayal. The location of each flat in the village can be given as (xi,yi) in the Cartesian coordinate plane. </p><p>There are "hi" persons living in the i-th flat. Central electricity authority of the village is set to built a wire line across the village. </p><p>The wire line is supposed to constructed in a way such that it is the north-east direction. </p><p>In other words the wire line is parallel to the line $y=x$. </p><p>Given that the construction of such line is considered to be effective only if the number of persons living in its left and right side are equal, can you tell if the construction of such wire line is possible?

Constraints:</p><p>
1 <= t <= 100
2 <= n <= 2*10^3
-10^3 <= xi, yi <= 10^3
1 <= hi <= 10^3</p><p>
It is guaranteed that no two flats are at the same location.

Input Format:

The first line contains a single integer "t" denoting the number of test cases.

The first line of each test case contains 'n' i.e the number flats in the village.</p><p> Next 'n' lines contains 3 space-separated integers xi, yi, hi

Output Format:
Print the output in a separate lines contains 't' lines each containing a "YES" or "NO"</p>

answer

```

#include<stdio.h>

int main()
{
    int t;
    scanf("%d",&t);
    while(t-->0) {
        int n;
        scanf("%d",&n);
        int arr[n][2], min, max;
        min = 9999;
        max = -9999;
        for (int i = 0; i < n; i++) {
            int x,y,h;
            scanf("%d%d%d",&x,&y,&h);
            arr[i][0] = y-x;
            arr[i][1] = h;
            if(y-x < min)
                min = y-x;
        }
    }
}
  
```

```

if(y-x > max)
    max = y-x;
}

int l = min;
int r = max;
int flag = 0;
while(l<= r) {
    int mid = (l+r)/2;
    int topLeftSum = 0;
    int buttonRightSum = 0;
    int equal = 0;
    for(int i = 0;i < n;i++) {

        if(arr[i][0] > mid)
            topLeftSum = topLeftSum + arr[i][1];

        else if(arr[i][0] < mid)
            buttonRightSum = buttonRightSum + arr[i][1];

        else
            equal = equal + arr[i][1];
    }

    if(buttonRightSum == topLeftSum) {
        flag = 1;
        break;
    }

    buttonRightSum+=equal;

    if(buttonRightSum>topLeftSum)

```

```

r = mid - 1;

else if(buttonRightSum<topLeftSum)
    l = mid + 1;

else {
    flag=1;
    break;
}

}

if(flag)
    printf("YES\n");
else
    printf("NO\n");
}
}

```

question

<p>Question Description:</p><p>Simon is studying B.Tech.-Mechanical Engineering. </p><p>He's going to attend a computer science-based subject exam this semester.</p><p>Due to the less preparation in the previous monthly tests, his internal mark decreased. </p><p>His computer science Professor made an offer one more chance to boost up his internal marks. </p><p>Professor assigns a program to Simon for the internal mark bootup. </p><p>So Simon wants to solve the given task is, Given two arrays, A and B, of equal size n, </p><p>the task is to find the minimum value of $A[0] * B[0] + A[1] * B[1] + \dots + A[n-1] * B[n-1]$, </p><p>where shuffling of elements of arrays A and B is allowed.</p><p>can you help him in solving Questions ?</p><p> </p><p>Constraints:</p><p> $1 \leq T \leq 100$ </p><p> $1 \leq N \leq 50$ </p><p> $1 \leq A[i] \leq 20$ </p><p> </p><p>Input Format:</p><p>The first line of input contains an integer denoting the no of test cases. </p><p>Then T test cases follow. Each test case contains three lines. </p><p>The first line of input contains an integer N denoting the size of the arrays. </p><p>In the second line are N space separated values of the array A[], and in the last line are N space separated values of the array B[].</p><p> </p><p>Output Format:</p><p>For each test case in a new line print the required result.</p><p> </p><p>Example :</p><p>Input : A[] = {3, 1, 1} and B[] = {6, 5,

4}.</p><p>Output : 23 Minimum value of S = 1*6 + 1*5 + 3*4 = 23.</p><p>Input : A[] = { 6, 1, 9, 5, 4 } and B[] = { 3, 4, 8, 2, 4 }</p><p>Output : 80. Minimum value of S = 1*8 + 4*4 + 5*4 + 6*3 + 9*2 = 80.</p>

answer

```
#include <bits/stdc++.h>
using namespace std;
class sor{
public:
    int a[100],b[100];
    int n;
    void getn(){
        cin>>n;
    }
    void geta(){
        for(int i=0;i<n;i++)
            cin>>a[i];
        sort(a,a+n);
    }
    void getb(){
        for(int i=0;i<n;i++)
            cin>>b[i];
        sort(b,b+n);
    }
    void display(){
        int sum=0;
        for(int i=0;i<n;i++)
            sum+=a[i]*b[n-i-1];
        cout<<sum<<endl;
    }
};
int main()
```

```

{
    if(0)
        cout<<"void sort(int a[],int n,int flag)";
    int n;
    cin>>n;
    while(n--){
        /*int a[100],b[100];
        int m,sum=0;
        cin>>m;
        for(int i=0;i<m;i++)
            cin>>a[i];
        for(int i=0;i<m;i++)
            cin>>b[i];
        sort(a,a+m);
        sort(b,b+m);
        for(int i=0;i<m;i++){
            sum+=a[i]*b[m-i-1];
        }
        cout<<sum<<endl;*/
        sort t;
        t.getn();
        t.geta();
        t.getb();
        t.display();
    }
    return 0;
}

```

question

<p>Question description</p><p>Sajid is a First year student in reputed institution. </p><p>Although he scored well in many subjects, he did not an expert in Algorithms.</p><p>But Sajid's computer examination is scheduled for next week. </p><p>As per the blueprint, many questions would come from the Arrays topic.</p><p>He collected previous year's questions. one of the repeated questions is you need to reverse the array in C Programming Language.</p><p>Can you help him ?</p><p>Function Description</p>

AAAAAAARr+7hY2dS4mm1BZ5LvKDDfauVzewo1rdRjN5JxeeQHqAAAAKev7S2NvcToT6TTi8nkgLgGtOaq
U4zuks0bAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAD3MB7mEU+FdtqfK/yXBT4V22p8r/J
cEhn8QBtRTbeSR56/8Aay2t6rp29N1pReTeeSK1ehB5al7Zr00q9pKK/wBrzPS29aNxQhWgmozWazA6
AAAAAAAAAAAAAYk8ot8EBkHmMKx+7u8bdpV0Ojzkti27D04AAAAG8lmeYre2EKVadPqrehJrfwA9ODyv
/AMZw/wD4r/8AcTMK9pY4lfQtIQCJHN558FmBfAAAAABX18asre76tUqNVc8sgLAAAR7y9oWNF1b
iajHu4s87W9s4RqZULVjxILjgeqBQYf7VWt1UVotB0ZPc3tRfp5gAAAAAAAAAAFFjOKLCreNV0+k0
nllmVNH2wp1a0KatmtKSWeYHpgAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA9wDAp7L+Xq/UuCmsv5er9S5JDHD4gAK2AAAPBe1N+7vE3Rg86d
H9Ky733s9hi99Gww6rWb/VIIH1e48V7P2jxDGlyqlShF6c/iA9nr6WHYtHT/AExqfomn3f8A9Z9BW7fme
D9qbLqeKurBaMKv6l6956r2fv1f4XTk3/qQWhNfFf0BG9r/AOEf/lvwy9ie33H/F/kt/a/+Ef/ACL8MqPYjt
9x/wAX+QPZ7iHVxWxoz0al1SUuGkth572rxmpCr1G3k4pL/Ua7/gVeGez13idLpk1TpvdKW9ge6he2tSi6
0Lim6cd8IJZlxRvrS4noULmlUllnoxmmzw2I4TfYPTbc9KjU/S3F7PqdfZD+b/8Atv8AKA94RrjELS2ejWuKcj
cHJZlP7UYxOxpRtrd5Vaizcl3I85huDXmL6VSMs0J7Zze8D3VDErO5ejRuaUpPuUlmsj55ieB3eEpVXJShn7
8e4v8A2Wxipdxla3MtKpBZxlxF5WvrS3noV7mlTnlnoymkzM7y2hRVWdenGnLdJyWTPFe1z//AFt/8cf
8kWyw/EMXiIS/VCmtFOTylgPd1pWI/h9TTqRlbyX6pZ7OZTYXhOD0L6E6N7GvVT/RBzT2/QkRs6th7K1r
eto6cYPPReaPLezf8/a+r/8AxYHrfaO96vhIxoK6hWTWxPaUvsxiteriUo3i03T6N5abyWeaMe1GE3Mbi4x
Byj0MpRyWe3ckU2GYfWxK5dGhKMZKLI+p5bP/AOsD6TTq06ybVlzs2ZxeZ84xf8Ambj/AJD2fs7hIfDLa
rTuJRIKc9JaLz7jxeM/zFz84HvaV9a2tIR6evTpVQWSIJ7jrb4haXT0aFxTnLgpLM8TZ4BiGKUVcOSUJe65va
yDd2tzHv30dRuFRbVKL3gfTSLUxKypzcKl3RjBgnNJo4YFfSxDc6dae2a/TL4tHh8b/mLn5wPoNxf2tsl09x
Thms0nJbRbX1rdPKhXp1HwjLaeJtMBxHFKWXJJS91ze1IfWpXOFXrhJuFWDzzT3gfSbi6oW0dKvWhTX
+6WRxo4pZV5qNK5pSk9y0lmzxVrYYj7QVZ1nPNJ5OU3s+hFxHDbnCa8Y1slpbYi94H0oFJ7LYhUvsOca0
tKdJ6OfFF2AAAAAAAAAAAAAAAAD3MB7mEU+FdtqfK/wAlwU+FdtqfK/yXBIZ4fFFxON
SeG140vfchkeHwK7s7K8IK/paSayWcc9F+h9BIJRg5SeSSzbZT1sLwjFl0sdFuX/dTeTZWraEsExSKhFUJN7
opKMv/ACS8QqV7XD5ys6anUgv0xyPH47gsMJ0KtvcaSk8IHvRaUsCrW3sxSuKi060punFv8gRXU9pqsem
WnFb9HRSfLl4B7Q1ri6VnfJab2Rllk8+BxsNfYnbq5p3kacG9izKrDVOhtPRjUmpzVX9UI3sC89qcVvMPuq
ELWt0cZQba0U83n8SLFcevKKr28JxpRW+ME8/ib+2/bbb/jf5PS4XCMcJoRSWWhuA8vbe1d7G3IRTV
Wu3ICWX+E6+KY9ZuNW4c4Rk9ilBJem45YTTh/wDFVOGitFVJbPoz0ntZFPaqja2qUcuYEnCcSWJYb06
WjNZqs4M8tH2jxSF7Upqp0v6nGMNBcdm4tfY9t4XcL/d/gpMGqUaXtlpV2IhpJLN8duQEqtfe0NnFXFdy
VN784LL67Nh6HBcXji1nOTShVgspRX5OuM16EMJrurKLi4NJZ7zzvsbGTuLmSTOOjy+oEbAF/wDuh/NP/J
fY/j7w6at7aKnXkt7/AO0ocAz/APih/NP/ACcsfjUh7Q1G5aDck4yfcgJ8q/tLSo9am5aGWbWjHd6ZFzgON
rFKcoVIqFeHvJbn8SArLHKILNYITICS8WzI29n8Er2d+7qVanUpuLT0HnmwPSkaVhZyk27Wi297cESQB8+
wSnTqe0MIVIRIByf6Wth7qFpbUJdTouqbS95RSyPD4D/AOpYfMz29+pSw+4UPedKSXIDzV57QX1eu1
wmOWTy0ss2/j6HKe45hFWEr9dJTk9zSy5oq8FjdSvZU7W4jQqtb5PLMuB3CsVuKPR3d/SIDPdKXeBcX
WOUqODK/pLT0l+mL4IDb3ftDikHxtZ5U88llopfcs6GB6GBTsrytFZy0ozW5FZHBcZw+MnZXMDF+iW/6
AWmGVsadG6jeRenTg+jbis3LLZ6nk7ypeTxRzultXOktmXeek9nMcurq+dneZSlk8pd+aKrGnl7USb2LTW
0D0mBVsTrSq6yg4pL9Occi4MKSe5pmQPB+0FxUxDHxBkR9GEujivyz1VhgllZUIw6CFSeX6pTWeb+p5P
GoTw/2jIXkv0ufSLOPa2l9b3ICNWjUi1JZ5Z7UBDq+z9hUu4XHRAjvjHYmaY9c4hbU6UcOpaWm8m1HN
om1MTs6VzC3nXiqk9yzKf2kxyvY1oW1pkpyWbk1u9AK64ufaSzp9PWlJQW1/pi8vsXmAY1rS2m6qUatL
3su9cSpurXG5YbUrXN7HonTzlHPesjI7F7bu4i9zgB0ucdxLEMRIla4YtDjtLJLN5d+060Je0lC6pQrfqhKWTz
SaX1Rzu/ZyurydfDLqGbk3k3k4keWK4xg93CleVokT7pbc0B6HHMZhVBZJSrT92P+Sjhce0l1S61SzVPel
or8Ef2tIKpeUK2TVOdNOJNtLbGq1pTnb4jT6NxWSz3ICZgOPzvaztLyKjXW55ZZnoDyeHYDdxxaN5O5pVJ
QnnPRe3M9YB5j2nxa/sbynTtm4U3HPS0U9J8C/w+rVr2FGpXjo1JQTkviaXI5ZWsoRu6tOLI7qkSYSjOCIB
pxazTQGKtGIWjo1acki4SWZ4LEqcKxtM404qMVVWSislvp0b4HFv/AFRL/IX5A98AAAAAAAAAAHuAe4Cmsv5er9S5
Kay/l6v1LkkMcPiAArYAON5cRtbWpXluhHMDyXtjf9LcwtIS/TT2yy4lbhONVMKjPoqMJynvcjjQpVcUxaK
km5VqmcvTez38MKsYwUerU3kss8gPE4rjlTFaMYVaMluMs009pl9k7/AKriPQTeUK+z69x694XYuLXVqe

ylgGzqW9pesa/FchrGvxXliAbOpb2l6xr8VyGsa/FcIlBs6lvaXrGvxXIaxr8VylgGzqW9pesa/FchrGvxXliAbX
qW9pqxOut6i/odqeKp7KkMviisA2sZrx+3oaVenWjnCSZ0POQqSpyUoNprgWtlfKslCpsnx4lerHni3aU4A
FegAAAAAAAAD3MB7mEU+FdtqfK/yXBT4V22p8r/JcEhn8QxKShFyk8kjJDxNtWry4Id2njG2HidBPZpP
6DWdD/dyKcE28PyLrjWdD/dyGs6H+7kU4Gz5F1xrOh/u5DWdD/dyKcDZ8i641nQ/3chrOh/u5FOBs+Rd
cazof7uRtTxGhOSjm1nxRSgbWPyLvSrdsBzoNuhBvgbvcyvdvttHrX1GjPRk22t+Rz1pQ4S5FTVbdWbfiZq
TbxT+RffZca0ocJchrShwlyKcDbn5F1xrShwlyGtKHCIpwNnyLrjWIDhLkNaUOEuRTgbPkXXGtKH+7kNaU
P93IpwNnyLvRUqsK0FKDzRuVuEN6NRfkV7MduVdy0q1YUoOc3kkRdZ0P8AdyOeLvKIBd2ZVkJc1q21
C41nQ/3chrOh/u5FOBtl8i641nQ/wB3lazof7uRTgbPkXXGs6H+7kNZ0P8AdyKcDZ8i641nQ/3chrOh/u5F
OBs+RdeUr+jVkoptN7sySecg8qkWt+Z6KPur0EPRhyTf7ZltTEKNObi221wJFV5Upv4M8483JtIM2Saa0u
NZ0eEuQ1nR4S5FOCMPkXXGs6PCXIazo8JcinA2fluuNZ0eEuQ1nR4S5FOBs+Rdcazo8JchrOjwlyKcA+Rd
cazo8JciXCCakFKLzTPOFvhTztn8JFa4strW1Kcca91St8uke17kjsUultu7I8Eg1y3mldwm6zof7uQ1nQ/3ci
oME28nyLrjWdD/AHchrOh/u5FOBs+Rdcazo8JciXCCakFKLzTPOFvhTztn8JFa4strW1Kcca91St8uke17kjsUultu7I8Eg1y3mldwm6zof7uQ1nQ/3ci
5hgAEeNgyAAQAm0008mAUX
NhddPT0ZP9cfuTDz9tVdGtGa47S/i1KKa3MsPoYb8q92QAVuAAAAAAe5gPcwinwrttT5X+S4KfCu21Plf5
LgkM8PiELFOyv1RNIWKdlfqirk8JU4AOxzAA2hCVSSjBzt9wGoO/Urjy2aVaFSII0kXHPdmFmsx9w5gAOQ
ABXobf8AYh6G8vdZpQ/Yh6G8vdZ0+rHi89V/cl6s0Nqn7kvVmPy+XP2AAIA607WtUWcabY4m8rKvFZuG
a+Ad8LekcBpp5MBwAACzwjdULircI3VCyLH0+jh8IV2Mftw9SrLTGP24epVkl5M/nIAAxAdYW1apHSjBt
M26lceUw64290AO/U7hf/LzxIFxk4vet4SazH2wAAjMPfj6no4+6vQ85D34+p6OPur0LD1/jftrW/Zn8rPP
Hoa37M/lZ54H5P3DAAI8gAAA0bSvKKKqbyZyIFxk4vegsxMfbAACBb4V2eXzFQW+E9nl8xYb4PNOKTE
e1y+hdIJiPa5fQS9H5HgigAjwAB1p21WrHSpwclxCx Ez9OQO/Urjy2Yna14RcpU2kt4Xjb04gAOQI4b22Po
yISsN7bH0Yd4/OF2ACvqAAKAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAHuAe4
Cmsv5er9S5Kay/l6v1LkkMcPiAArZUYrPoTGOe5EEk4i872fwy/BGI+ZlndpAARmAAAAAAAAAAAAAAA
AA
AAAAAAe5gPcwinwrttT5X+S4KfCu21Plf5LgkM8PiELFOyv1RNIWKdlfqirk8JU4AOxzA6W9Toaym1nI3H
MAidTtdWt4rbmbioNZLM4Yvup+rOeE/vz+U6Yvup+rK9k2m2LcqwaEeIAAV6Gh+xD0N5e6zSh+xD0N5e
6zp9WPF5p2p+5L1ZqbVf3JerNTI8ufsOttOFOspVI6UUcgCJ1O1pVxOMdIgOfxYtsRdWqoTgInuyKsm4dbS
IUUVWYK3fFlb0yXtbs6YpSitGpHY28mVxYYpWUpKnHbltZXkcZpjnOgABks8I3VCyK3CN1Qsiw+jh8IV2L
/tw9SrLTf/24epVkl5M/nIAAxTrfEl0aMabg3kSaF/01VQhSb/wVBdWNNGN0gpLfLaw9WG97TrbtWrQo0
3Kfloak9OpKXieZZ31rVrPSjLNL/tKprJtPZkEz2tM60yYADzNoe+vU9FH3V6HnYe+vU9FH3V6Fh6/xv21rf
z+Vnnj0Nb9mfys88D8n7hgAEeQJdha9NPTkv0R+5woUZV6qhH6lJwtfLfgktnxZW+Lhv8A1P05X1yqNLQ
h7z+xT795LoUnfVakpSyaOd3bdXmo555oGTIf/X6cAARgFthPZ5fMVJbYT2eXzFhv+P5p5SYj2uX0LspMR
7XL6CxO/I8UUAEeAlrDlax+O0pS8s9lpT+Ur0fj+ThPE4xqSjoN5PLecq+IRq0ZQUGs1kJ4dQJyblm3nsZF
ucPISi503pRXd3h3fqxEoIAI8gSsM7ZH0ZFWGdsj6MO8fnC7ABX1AAFAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAD3APcBTWX8vV+pclNZfy9X6lySGOHxAAVso8Q7bU+n4lxJxDttT6fgj
EfLyUgAI4AAAt8K7PL5i
AAAt8K7PL5i
oLfCv2JfMWG+DzTgAV9AAAAAAA9zAe5gU+FdtqfK/wAlwU+FdtqfK/yXBIZYfEIWKdlfqiaQsU7fK/VFXJ4S
pwAcvmAACJ+E/vz+U6Yvup+rOOGThCtzkorR72dMUqQqKnoTjLLPc8yvXEx0VcACPIAAK9DQ/Yh6G8v
dZpQ/Yh6G8vdZ0+rHi87V/cl6s1Nqv7kvVmPy+XP2AHW1jCVeKqNKPfmwRG5072dk6rU57lfkkXd5GjD
oqO/dmu4kqtQ0culhI8xx6KxbzcoZ/OV7OHGuqyqG23m9rBZXNO0jQk6Tjpd2Usytl8t6TWQABws8i3VC
yK3CN1Qsiw+jh8IV2L/tw9SrLTf/24epVkl5M/nIAAxbaE9DT0Xo7szZXFWSjOS+pY2t1b9DGm2o5LbpH
dRtX+r/T+weiulcbiSzqTqW6dTeVi+kruWj6ljXvaVCH6WpPuSKepUdWpKct7ZXWa0cYjbUAEeVmH7kfU
9HH3V6HnlfuR9T0cfdXoWHR/G/bWt+zP5WeePQ1v2Z/Kzzwk/J+4YABhkW2GxhGhpbe29plq06VbJVG
nI8Si0pJZKTS9Rpy8UuZXprmiK60vqNCISbdJLbvMVqFGq06iTalOG1oxU+kqJcNJmuJVVKrHo55rL/tYa9S

vT3pDqpRqzS3JvI1D2gjxBbYT2eXzf+CpLbCezy+b/AMFhtg808pMR7XL6F2Uml9rl9BL0fkeKKACPAHZut
TitsosthzhqcXJZpPai4hcW9WCTIH0kGuOvL96VULmupfpqSbLyLcqSclvW05qNtF6S6NfHYR7u/hGDhs
ecn3or01jpxM2IW1EIUko7szUZ5gjw/tglYb2yPoyKSsN7ZH0Yh3j84XYAK+oAAoAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAe4B7gKay/l6v1LkprL+Xq/UuSQxw+IActlHiHban0/BFJWldt
qft8EYj5WTykABHIAAAAAAAAAAAAAAAe4B7gKay/l6v1LkprL+Xq/UuSQxw+IActlHiHban0/BFJWldt
AAAAAAAAAAAAAAe4B7gKay/l6v1LkprL+Xq/UuSQxw+IActlHiHban0/BFJWldt
XZ5fMVBb4V2eXzFhvg804AFFQAAAAAPcwYe5gVGFDtqfK/wAlwU+FdtqfK/yXBIZYfEIWKdIfqiaQsU7K
/VFXJ4SpwAcvmAACACgAAAAD0ND9iHoby91mID9mHoby91nT6seLztX9yXqzU2qfuS9WanL5c/YAAg
ACACgAALPCN1QsitwjdmSiw+jh8IV2L/AlcPUqy0xf8Abh6lWSXkz+cgADEM5vizAIAAAozD9yPqejj7
q9DzkP3I+p6OPur0LD1/jftrW/Zn8rPPHoa37M/IZ54H5P3DAAI8gAAAAAAW2E9nl83/AIKktsJ7NL5v/
BYb4PNPKTEe1y+hdIJiPa5fQS9H5HiigAjwAAAazviwAFaaECVhvbl+jIpKwztkfRh3j84XYAK+oAAoAAAAA
AAAAAAAAAAAAAAe4B7gKay/l6v1LkprL+Xq/UuSQxw+IActlHi
Hban0/BGJ0ldtqfT8EUj5WTykAEcgAAAAAAAAAAAAAAe4B7gKay/l6v1LkprL+Xq/UuSQxw+IActlHi
h8Qi4hTIUtWorNraSgV3aOUaeayB6GVCJ5uEW/Qx1aj5ceRNpj8afbz4PQdWo+XhkOrUfLjyGj40+3nw
eg6tR8uPlDWo+XhkNHxp9vPg9B1aj5ceQ6tR8uPlaPjT7efMxi5tRis2y/6tR8uPl2hSpw92CXohoj8afzsjo
0orgjZ7mZBXs1208/cu5U68IJzbTkeinThP3op+qNerUfLjyJp5J/GmZ7S8+D0HVqPlx5Dq1Hy48hpPj7ef
B6Dq1Hyo8h1aj5ueQ0fGn28+D0HVqPlr5Dq1Hyo8ho+NPt58HoOrUfKjyHVqPlx5DS/Gn2iYVTIGnKUlk
m9hYGEklkZk9VK8Y0g4rTIOjFxWai82VB6U5O3pP/5ceRGOTDzncS8+D0HVqPlx5Dq1Hy48hpl8afbz4
PQdWo+XhkOrUfLjyGj40+3nweg6tR8uPlDWo+XhkNHxp9vPrmS/6tR8uPlDWo+XhkNHxp9qKjCVStGM
Vm2z0MVIFL4GsKUKfuRS9Ebhwix8la1FpU5Jd6aPPThKE3GSyaZ6M0nShP3op+qKZcXUedB6Dq1Hy48h
1aj5ceRGHxp9vPg9B1aj5ceQ6tR8uPIHxp9vPg9B1aj5ceQ6tR8uPIHxp9vPg9B1aj5ceQ6tR8uPIHxp9vPlz
htOVO2/UstJ5ndW9JPNu45+h1DXFh4TuQpsTpyjcObX6XuZcmJRjZSSa+JWmSnONPng9B1aj5ceQ6tR
8uPlmn+nPt58HoOrUfLjyHVqPlx5DR8afbz4PQdWo+XhkOrUfLjyGj40+3nweg6tR8uPlDWo+XhkNHxp
9vPk3C6cnc6eWyK3ln1aj5ceR0jFRWUUkvgNOqfj8bbmWQAV6gAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAPcA9wFNZfy9X6lyU1I/L1fqXJY4fEABWjxDTtT6fgjEnEO21Pp+CKR8rJ5
SyACOQAAAAAAAAAAEJTeUlUT+CNS7t6cLe1TS7s2+laY8fOVZ1Kvv6NnKdKpT9+Dj6omvFjqXuLL1NL
q+Veioxi0+8rua49dpQgW2HW8qcZSmI+rLI5XlnUnVIOOWjkROIPHkrbg06NSrNxhHNolrC6rW2Ucw4rjt
b6hBB3r2IWgs5rNcUaUKMq9TQjlnIntCcZidOYJT+spqKyeazzNqdjUVxo/peWTYddO3pDBd3dB1aDjTS
0mVFahKhJRnlnInsC5Mc0cwAGQAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAFvhXZ5fMVb4V2eXzFhvg804AFFQ
AAAAAAw9zMmHuYFRhXbanyv8AJcFPhXbanyv8lwSGWhxAVqAAAAAAAAAAAAAAA
AAAAAAAAAAABrKcY5aUks9izZseJ9sK9RYnCCm1GMFkkwPbAh4ROVXCrac3nJw2smAAAAAAA
AAAAAAAAAAAAAA
AAAAAAAAAA9wD3AU1I/L1fqXJTWX8vV+pckhjh8QAFbKPEO21Pp+CMScQ7bU+n4lxHysnIIACOQAAAAA
AAAAACztcQgqahW2ZbMysJ2rZSoxIGX6stqYbYuUTuqW7e0uNyjn/tl3YdFFzpvOK3rgaww+5jZNR+kZ
YXMuj5KbWejl6srF0J5Rpxw2tUqxLGbz0csjlfXVWnXITjL9OXAYTjadRd7yM3tnVq3DnBJrlQ53a2KNIdr
VqU62dJaTfcTEsQctLd8NhnC4xXSzr9SYutuu1Tb0e7IJSsxTaXKMqls1UWTA2Izhvbfoy0gpRtsqjzlltzWY
d236MO7+VU69uerxSiv1MgU7ys7hPNZyyT2HbFvfgQYPRqRfBhllvbmub2rOlbuchkymq1p1paVRptBN
xc3VN17Zxp7W9xUvredBpVftfAOVyuOuz/AByABHIAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAt8K7PL5ioLfCuzy
+YsN8HmnAAr6AAAAAGHuZkw9zAqMK7bU+v/kuCnwrTT5X+S4JDLD4gAK1AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAABnhfbH+XXyl90zwvtj/Lr5EB6zBP4a1+QnEHBP4a1+QnAAAAAAAAAAAAAAA
AAAAAAAAAAAAAA
AAAAAA9wD3AU1I/L1fqXJTWX8vV+pckhjh8QAFbKPEO21Pp+CKSsQ7bU+n4lxHysnIIACOQAAAAA
AAADvSvK1JJrlmuD2nABYmY+kx4lwYzL6EerXqVnnUln8DmA6m9p7TLaE5U5aUHIJErWVxllnH1yIYC
Re1fqXSnapCppxll7vErhrLOOfHiiAExth1KTG/rqOjpJr4o40q06NTThln8TQA5Wn9ute4qXDTqZbOBzM

QQkzM95SKV7WpR0YtNfFHOvcVLjLpGtnBHM**Z**VaY1tgyAHLAAAAAAAAAAAAAAAW+Fd
nl8xFvhXZ5fMWG+DzTgAV9AAAAADD3MyYe5gVGfdqfK/wAlwU+FdtqfK/yXBIZyFEBWoAAAAAA
AAAAAAAAAAAAAAAAM8L7Y/wAuvkR7pnfbH+XXyID1mCfw1r8hOIOCfw1r8h
OAAAAAAAAAAAAAAAAB7gHuAprL+Xq/UuSmsv5er9S5JDHD4gAK2UeldtqfT8EUYh22p9PwRiPlZ
PKQAEcgAAAAAAAAAAElz1a1PTilkFiJn6RwZknGTi96MBAAAAAAyYMgADZU5yWcYSa+CBrbUDc
9oAwAAAAAAAAAAAAAAAM8L7Y/y6+RH
AAAAAAAAAAAAAAAFvhXZ5fMVBB4V2eXzFhvg804AffQAAAAAAw9zMmHuYFRhXb
anyv8lwU+FdtqfK/yXBIZyFEBWoAAAAAAAAAAAAAAAM8L7Y/p8fq3IHDpVbF5Tg85bM3keCvL2vfVuluJ6c8ss8sgPoWCfw1r8hOPDYFiWJ1rqha0amdKO9aOxI9yAAA
AAAAAAAAAAAAAAAD3APcBTWX8vV+pcINZfy9X6lySGOHxA AVso8Q7bU+n4lxJxDttT6fgikfKyeUsgAjk
AAAAAAAAAAAusP7HEpS6w/saLD0YPJYh0l24cZZEeFyWWjPPP4biPT7evnLO+rSo27cd72jkWlazEzZ
F1Xs/c2+hDuKMqFTRn9Gd7O6qu5hGcnJSfeSMWinShLvvKTWtqcq/pHtrHrFLT08tuWWR0hhkNm5zy4
LiKY2ZT6k7uqrzSk4qLySQXjStlMNq+HVKacoPSSI1Cl01ZU88sy0w+tKtRantcXlmRoQVPFNFBm8yJale
0x9S5Xdp1ZReInmc7aj09VQzy2bybi3u0/Uj4Yv+rXow5mkRl4sXFnONWEFLPT+BZ2tB0KOg3n8SHircZ02
nk0SbCc52qlN6T4srakVreYQrxqITjKo5p7c8siEd69epKcoynJrPdmCPLkmOXYB2tqKr1tBvLZnmTHhcfNf
ILXHa0bhEs406lbRq5JZcSTVwx76Mk1wZCnBxqSis3kyXa0rxNOLcY/7g6pET/mYRaICpSf64tfE5I9KcYu/9
acd20opNaTy3Zgy44prUsE60HOVHTU821mlkQe9F9CWhaKXCOYXDSLb2gwwttfqzP4i4XNnO3WIhp
R4mOuVnVUtNpZ7i0r/wCpYyb745ldRWl6zr9KMAEeYAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAC3wrs8vmKgt8K7PL5iw3weacACvoAA
AAAAAYe5mTD3MCowrttT5X+S4KfCu21Plf5LgkMsPiAArUbjNcRmuIRkGM1xGa4g3DIMZriM1xBuGQYz
XEZriDcMgxmulzXEDIACgBjNcUBkGM1xQzXFBNgxmuKGa4oDIMZrihmuKBtkGM1xQzXEG2QN4CgBj
NcQMgxmul0lxCbGIKLULmnvR4jHsArULzpLOIkDgq80or3xwPb5riM1xBtV4BhMMMs1pLOvPbN/4LUx
mul0lxBtkGNJcRpLiDbIMZriZCgAzSAAxpLiNJcQm4ZBjSXEA S4g3DIMaS4jSXEG4ZBjSXEA S4g3DIMaS4m
QbAAFAyzXFDSXFBNsgxpLihpLigbhkGNJcUNJcUDcMgxpLihpLigbhkGNJcUzzQNgACgAAAAAAAAAAA
AAAAAAAAAAAAAAE4B7gKay/l6v1LkprL+Xq/UuSQxw+IACtlHiHban0/
BGJOIdtqfT8EUj5WTylkAEcgAAAAAAAAAF1h/Y4IKdYXNanHRhUaXArXFekTuW9Pt6+cnYp2depVqclP
TT/VnnmbVLirVjozm5L4klyRFZj22s+10/Un4r+xD1KuMnCSIF5Nbmb1K9Wqkq3JfEpXjeumq0wzsn1O
dfD+lqudOaSk9qfE6YZ2T6sgVLirSuKihNpaT2BvM1jHHKFIRpU7Oi85fFsryXCd8qz3Z/Y5VK9Wr78zmG
V8sTql+oXd1QV1SijJ70zS0slbcznLOTWSKync1aayhNpB3Fzz0uklnxDvrU3ymO6Zi2elTO2GTTtHPanuk
upVqVWnUm5ZbszEKkqcs4SafwDiMsRk5JlzYTi6IVSTW/Ignad3WnHRIUbRxlzvNZn/LMZOLzi2nxRt01xz
J/+5mgDmJmHe1rxoVnOacsxrVxGrPNQyvgQwHUZjINQzKcpvOUm38TAAcneXmf/Qv5CjOvWa2hodl
9HLLINMd4ptyZdz/AI//AO3/AIKQ6u5rOGh0j0cssgY8kv3/AFx7zIAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAW+FdnI8xFvhXZ5fMW
G+DzTgAV9AAAAADD3MyYe5gVGfdqfK/wAlwU+FdtqfK/yXBIZyFEBWoAAAAAA
Sk5OctvxMdLU8cuZoDI8zlfpanjlzHS1PHLmaAjylv0tTxy5jpanjlzNADlfpanjlzHS1PHLmaAHKW/S1PHL
mFWqJpqpLNfEOANyv7So6tvGUt7R2luHdkgSjp9Ok7rCnxC4m7iUlycVHg8ij0k/HLmdb7lt1OBHz8lp5S
26Sp45cx0lTxy5moI43LbpKnjlzHSVPHLmagG5bdJPxy5jpKnjlzNQDctukqeOXMDju8cuZqAbIYZCtIN05
yb71mWZT4X2n6FwdQ+hgndHC9qujbSnHf3FI61RvN1Jcy3xPscvVFkr5/yJnk36Wp45cx0tTxy5mgI8+5
b9LU8cuY6Wp45czQA3LfpanjlzHS1PHLmaAG5b9LU8cuY6Wp45czQA3LpCvVhLNTlMviX1GenSjLjzpf2
nZoehYer8aZmZdnKO7uak68v1NPJJMvHuPO1v3p/MxLv8iZiY6Wp45cx0tTxy5moI8W5bdLU8yXMD
LU8cuZqAbI0tTxy5jpanjlzNQDctulqeOXMDlu8cuZqAbI0tTxy5lphledWEoTebjtTKkscl9+r6lsNsNp5wt
EV+KV509GEHlntbRYFVi/7sPQr1Zp1RB6Wp45czPSVPHLmaA5fP3LfpKnjlzMdLU8cuZqAbI0k/HlMolq
eOXM1BTctulqeOXMDlu8cuZqAbI0tTxy5neyuKkLiMXjtSeTTZFOlt2qn8yDqlp5Q9CgECVqAAAAAAA
AAAAAAAAAAAAAAE4B7gKay/l6v1LkprL+Xq/UuSQxw+IACtlHiHban0/BGJOIdtqfT8EYi5WTvkABHIAAAAAAA
Okk1WEdGE2i8GaNttvNswilNvwAwAA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAC3wrs8vmKgt8K7PL5iw3weacACvoAAAAAAYe5mTD3MCo
wrTT5X+S4KfCu21Plf5LgkMsPiHG87LP0OxxvOyz9Cur+MqAAHL5gAdbaCqXElyWabBEbnTkC86pbxW2
Crr1e18MeZW/x59qUEq/hCFZKmkII3EYjC0cZ0wAAi7w7skCURcO7JAIfFux+MKK+7ZU9SOSL7tIT15Hz
b+UgADkALKzsoqi6ldLbuT7kHdTTeeytB1uJ05VH0UUoLd8TkHMxqQABE3C+0/QuCnwvtP0LgsPofj+Cji
XY5eqKQu8S7HL1RSB5/yfjkAEecBPsKNGpSbqpN595L6ta+GPMNq4ZtG9qUF51O38tFVeRhC5IGmkorg
C+KaRuXAABiF/admh6FAX9p2aHoWHq/G+5dnupO1v3p/Mz0T3Hna370/mYI3+T9Q0ABHiAdLei69VQj
s4ltTsqFKP6lpNd7DWmKbqUF51e3qLLri18CuvbToGpQecH9irfDNY2iAyCMWCxwj36voiuLHCPfq+iLDX
B5wtCqxf92HoWpVYv8Auw9CvX+R4K8GDJy+eAEzD6dKpKSqxT2bMw6rXIokMF2rS2lumgVI9TjTuXGC
yWS2BpfDNI2jgAMQ6W3aqfzI5nS27VT+ZFdv+4ehW4BbgV9UAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAD3APcBTWX8vV+pcINZfy9X6lySGOHxAAVso8Q7bU+n4IpKxDttT6fgjEfK
yeUgAI5AAAAAAAAAAJdCxNwpdIpJliFzh/Y4hthrFp1KonHQm4vuZqdnBVLzQe5yyJ0sLi8tGbW3bmH
MY7W3pVgttW0csIKWZAureVvUye1PcylsVqxuXAYTrOyp16OnjtPPLYdY4bTjm6k36BYw2mNqwFjXw1
KLISbfwZDt6Sq3CpyzWY05nHaJ1LkCXe2sLaMXFt5vvOVpSjXrqEm8ss9hCaTFuLidIW1apHShBtEi5tIuq1
OEW8pb8yxt6KoUtBNtcStaYZmZiVE002mtqBPu7KFOnKopvPgV5GV6TWdMgwA4Ad7R0o1v9bLRay2k
ydhSqrSoTSz+OaDSuObRuFYZO9WzrUt8c1xRH DiazH2yYBNtrB1YqdRuMfyFrWbTqElFrq+3lsjUefwZBu
badtPJ7U9zK6titWNpVrZUq1upyzzIE4qM5RXcy3sOxoqqkZTryjFZtyDvJWIrXTmCyp4bGMFKtP6cDM8O
pyhnRlt9Q56NIYDoqeVdU57P1ZMsJYZDJaMmn35hK47W+iWC11ZScdkpZ8TIHDJdI9Kf6F3oOujdXgtnh
tJx/TJp8Sur0ZUKrhL6PiHNsdqquXIEu1sZV1pyejD8knV9B/pU3peoWMVpjardBlurWVtLPPOL7ztZ2UK9D
Tk5J5tbCJGO024oILSGGU1npzb4JGlxhujBypNvLuZdOujfW1ek5Njb3uOIS2q0oaU4NI72FtGtLTcmnBrYi
yuaCuKWg5aKzz2BaYeVdqEEi4tujuFSptyzJcMNpxhnVntDiMVpnSsBZzsKDg3Cpu+JWPfvzIl6TX7ACdbY
e6kVOo3FPu7wlaTb6QQWRw2i09GTz9SFcWs6E0pbU9zK7titWNlpau5m1nFb2Tallbwi85PNLid7S3VvB
5SctLbtI95aR/wBSvpvPLPIN4x8ab13VfewDrbUOnqqGko7MyPJEbnUOQLVYdQWxzbfqcbnDujg50m2lv
TK1nDaI2gA7WIJV66hLNInSwuOmtGTUe8Oa47WjcKsEy4tYU7qnSi3ILLPmd5YZBpaEmuOYWMVp3pW
AmXtrC3jHRbbbyez2pYdT0FOpUzT4ElxW3pWgtJYbSIH/AE5tEJ0OiuVTrbFnvBbFav24An3djGjR6Snm8t
+ZCpwdSpGC3t5Bzak1nUtQT7uzpW9HSzek9xABas1nUgADgAAAAAAAAAAAAAAAt8K7PL5ioLfC
uzy+YsN8HmnAAr6AAAAAGHuZkw9zAqMK7bU+v/kuCnwrTT5X+S4JLD4hxvOyz9Dscbzss/Qrq/jKg
ABy+YEmwWd3D4bSMbat1etp5Z7A6pOrRMrS/pTrUVGms3mVs7SvBZuDy+DJetf/APGyTbXkLluKTUI
3Mr02imSfvupHmt4LDE6Ci41YpLPYyvDzXrxnQACOF3h3ZIEoi4d2SBKK+pj8YUV92yp6kckX3bKnqRyPm
38pADrbUXcVIBbu8JEbnUO9hadLU6SfuR+51xC5zfQUvrkS6kJU7fQoRWes+BApWNdVoyklvze0r1T
Wa14xCJKNOKzlPfpE1Lq/oTrUVGmtqZUVaM6M9Ge8jHJmk/xoAAyTML7T9C5KbC+0/QuSw9/wCP4Im
J9jl6opS6xPscvVFKGH5PkAAjzhNsLV1Zac/dX3IRNtb5W9HQcG/iVpjmoX+ky+ryo0tGmmn+/LcU29555I
7QuKd1TeUfVMrL+hGjW/Qsoy3IS2zRuOUT2RQAR5Qv7Ts0PQoC/tOzQ9Cw9X433Ls9x52t+9P5meie4
87W/en8zEu/yfqGgAI8TMW4tNNprgb1bipVSU5NpHM60KEq89GC9XwDqu57Q7YdpO6Wi3kt5NxKSV
tk97ew6U6dOzovPJJbW33IXd3LuKmf/atyK9M/+ePjP24AAjyMFjhHv1fRFcWOEe/V9EWGuDzhaFVi/w
C7D0LUqsX/AHYehZev8jwVxkwZOXzwAAW2F9nI6kTEu1v0RLwvs8vUiYI2t+iK9d/+MlgAI8gdLbtNP5kc
pbpp/MirX7h6HuA7gV9YAAAAAAAAAAAAAAAD3AP
cBTWX8vV+pcINZfy9X6lySGOHxAAVso8Q7bU+n4IpKxDttT6fgjEfKyeUgAI5AAAAAAAAAAALrD+xxKXuL
rD+xxLD0fj+Sth29fOWWIVJU7Z6LybK2Hb185OxPs69SOqdqWQrGrNXUVpNqTyZMxbLooP4kGz7XT9S
div7EfmlKINzilvhnPqV95VnO4nnJ5J5lsMM7L9TFWyo16mnGeTz25B3atrY4irGF1Zzoyj56L2HKMVHF
neyUIRsQlwWXNIZG4fW1WFEfp4xWJS8X92n6kfDF/1a9GNWNanSu6SWlmu5o1t7ajbTeUs5PiFmkzk5
fpGxb36Zlw9t2qbefqR8W9+n6HTDKsXR6NtaSe4ETrLKurN9LNNt7TmWVzYRSnVU3nvyK0jzZKzFtS72I
OFWtoTeSyJzw+28T5IUAVvWsamHSVJyrShTTIk9hMt7GvFqTqdH6ES2r9XqOaWeaNqt5WqvbNpcEHV
ZpHeVrO4pUKeU6mbKSTz38TADnJk5t0LOpFPc5JF5Wp6VBwU9DZlmUS2NNb0W9GvSuqHRze3LJorX
BMd4lxhZRhNSjc5Nfa64i4Ttx+pOsayNNVw0s+kejwl7Rp0qiVKWFB1bdKzGhYdjRFsYqV9Ub7s8iVYdj
RXUq/QXbn3ZtMEzERWZWN5Q6ZxTraCXca21CNvJtV00+43q0qN7BNS29zXccVhtKGbqTbXx2BpaJ5co
hyvVDrtKUe9rPL1JI7N07VuO/cVMYxjdRUzaUVJbSzxHsj+gZ1tuLSgWNWauorSbT3k7EqkoUEovLN5Mr
7LtdP1J2K/sx9QIJnpS4YZUn1hxcm00b4ql0kH8Djhva16HbFvfgEif/LunOCVvoxegst/AgxsYxkpK52radrW

5p3FH06jWllk8+80eF03LNTkohrMcoiaxt1vHCVpNOSbSzMYb2VerIV9b0qLXRy298SbhNZF6sQlZmcndX
XNac6825PY2ki0spyqWicnm9xT1v3qnzMsP7GhDjDMzeVbBtXqSeS0+71LHE21a/peW1FY3o3blwnn9
y4qwhd0MILY9uwLj71tEkehU0LiFSWbSZbVoUrums8mQKtnGncU6bn+mW9skPDYN50qjSCY4vG405
1MOnGLdOpn8CvaabT3ou6aVrQaqVNLLvZT1pKdaU1ubzDjLWK60zbzXuriExubLXEKkqVt+jZm8syohJw
mpLemXForSvKGjLLPvTC4fGa/tU0as4VYyJ7+ZbX8VK0k3vW1GIPDqVKanKWkl3M44hdRIHoqbz4tB3E
TSk8nTCm5UZZtvaQr2Uut1Fm8syZhX7M/UhXvbKnqHN/wDIDgdrWjVq1P8ASeTW98Di8LqwjKUJPv
dmRjiJtES2eH7c5XGT1OGVvouWnsyz4kW5sOmrOoppJkqhCELdRg80tmZ09tY1MxpV2Gy+y9SXidat
OEYweWlvZEse3v6nfFsv9MjGJ1jlDoNu6pNtv9SLS/qypW2IB5NvlqrftNL5kWWKdlXzlJjmenaVS5Sm/wB
Tbj8LGrUpRdSropLdwINOWhVhJ90ky6rQV1b6MJ5J7c0Ew1i25lztLfoJvKtpJrcRsVvVaD78iRaWsbarne
c3sSI+Lfuw9A0yR/5JVrNXFnoyebyyZGw+3cbiUpL3NiOeG1tCvoP3ZlnXqKjSnP4BaavWLT+lZiNZ1K+gn
QzMm5Sbe9mCPJa3KZkAAcgAAAAAAAAAAAAAFvhXZ5fMVBB4V2eXzFhvg804AFFQAAAAAAw
9zMmHuYFRhXbanyv8lwU+FdtqfK/yXBIZYfEON52Wfodjjedln6FdX8ZUAAOXzAmWFrGu5SqbYpZlhm1
OpOnLOEnF/ALWYidysJ4Vm/0VMI8USLSyVvjctKXpkV8cRrpZzP/QxO/rzWWII6FeiL4o7xDvilZScUXnk
82V4bzeb3gjC9uU7AAHC7w7skCURc07JAIfTx+MKK+7ZU9SOSL7tIT15Hzr+Uh0oV5W89KGWeWW0
5gJE6ncJus63CPI2o4hWnWhFqOTaT2EAzGThNSW9PMbaRlv7XN7XnQoqUMs8+8qa9aVeenPLP4G1
W6q146M2sk+BxC5cnOe30AAMUzC+0/QuSmwvtPOLksPf8Aj+CJfY5eqKUusS7HL1RSBh+T5MgAjzrCzs
oVqDIPe92XcZeFPP9NRZehCpVqlJ/6cmjusSrpZNP/Qr0Vti1q0LG0tY2sHtzB3rsRqqrcZReajsNKl5XqLjzy
Xw2EcGTLE141AAR5wv7Xs0PQoC/tezQ9Cw9X433Ls9x52t+9P5meie487W/en8zEuvyfqGgAl8behT6W
rGGeWZeUKEaFN瑞qUVOpKINThvR31hceNct8V60+0+5s53Ev1Vso9yy1TDejpyn0ueS3ZHhr9x4lyMT
va84uMpLJ/AOrXxz+nAGDJHmYLHCPfq+iK4scI9+r6lsNcHnC0KrF/3YehalVi/7sPQsvX+R4K4yYMnL54A
ALbC+zy9Sjixan6i5UbqrRjowaS37jSrVlWnpzebDe2SJxxVoAAwDpbddpp/MjmdLbtNP5kVa/cPQ9wHcCv
rAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAe4B7gKay/l6v1LkprL+Xq/UuS
Qxw+IACTlHiHban0/BGJOIdtqfT8EUj5WTylkAEcgAAAAAAAAAGVOUVkpSS9TABvTObzzz28Q5yIsJv1
YMA2ynluMucpLjyb9WagEST8Nf/S/Urq05RuKmjJr9T3M5Jtbm0YDW2TdYj0zKUpPOUm/VmAAYltGpO
Puya9GYc5N56Tz9TAbuWXKuhnKTfqwm4vNPJmADbd1JyWTm39TUwZBsBgAAAAAAym1ueRgAb9L
Uyy05ZeppmAF/wD1ITIFZKTS9TD2sADKnKPuya9GZdSct85P6moBuQ2IOUl5Nr4s1AQtaeaeRmU5S96
TfqzAAzFuLzTafwMyIKXvNv1ZqZB3Yzaex5G/S1MstOWXqaALse3a9rNIOSWSk0vgzUBGxtCnJLjsaxwzg
ANRzTGPu9GagKzKUm822/U2VWa3Sa+poAbIs5ylvk36moAQMQtjubXoYAGzq1GsnOT+pqAF7tIK
Ufdk16M1bbebebAABbGAEb9LUyy05ZeprrySyUpL6mAF3LKbTzTefEy5OXvNv1ZqZCbYWx5o2cpSWTk
36swYAGYznB/pnjfUwANukm3m5PPjmbUI0taMZY2N72zmEFifa3p2FKnVU9JvLbkR8SuVNqnB5pPaQt
KWjlPj0K1tl7aiNMgAjAAAAAAAAAAAAAC3wrs8vmKgt8K7PL5iw3weacAcvoAAAAAAye
5mTD3MCowrttT5X+S4KfCu21Plf5LgkMsPiHG87LP0Oxxu+yz9Cur+MqAAHL5gAAgAAMGQAAAu807
JAIEXD+yQR0+nj8YUV92yp6kckX3bKnqRyPnX8pAARyAADBkAAAJMf9p+hclPhfafoXBYfQ/H8ETEuxy
9UUhd4l2OQqvUoef8AJ8gAAecAAyYAGTAAv7Xs0PQoC/tezQ9Cw9X433Ls9x52t+9P5meie487W/e
n8zEu/yfqGgAl8QAAAAAGTAAfjhHv1fRFcWOEe/V9EWGuDzhaFVi/7sPQtSxf92HoWXsz+CvABy+cAA
AAAAAAHS27TT+ZHM623aaflq18oegA7gV9YAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAD3APcBTWX8vV+pclNZfy9X6lySGOHxA AVso8Q7bU+n4lxJdTtT6fgikfKyeUsgAjkAAAAA
AAAAAAAIW1jOvHTb0Y93xDqtZtOoRQWuraO5zefqRrx6CGnGeaz3NB3bFasbIDCO1C2ncZ6GWzfma
VqUqNRwlmuAZ8Z1tqDACAAGUJtJpLewfbAJNWYq0Y0C9HjcGRgs1mv2AFhh9vSrU5OpHNp5BaVm
86hXg63UFC5nGKySyzCTGp0AAIAAAAAAAADta0emrxj3b2SL+nRoqMYQSk9rDuKTNeSD3ZgtnVte
qZjxy0dxGsKdGs5QnBN70Hc4u8REoQ095Q6Cu0tz2o4BIMTwdSGQk20lvZa07KhRpaVbj8Ww6pSbqk
ySb1UFKPQZfHIjBzaOM6YAAQAAA2hFzkoxTbe4DUFITwxaoDWe3gjFbDcouVGWb4Mrbo31vSuBlpp
5NZNGCMgHahbTuG1Ty2b8zStIRqOEss1wC8Z1toAA5AAAAAAAdaEVJqENsmak/C6WIUdRrYtiDqle
VtItahUo5diss/iciwr053t1JQeUlbMzqsMpJfqm2w06UzP+fpVAm3OhypRc6b0oreuBCDO1Zr9gQAcsm
BmMwAAAAAAAAAAAAAAABb4V2eXzFQW+Fdn18xYb4PNOABX0AAAAAMPczJh7mBU
YV22p8r/JcFPhXbanyv8AJcEhlh8QxKKIFxe1MyCtFbPCk5NwqZLgzGqX5v2LMBI0KeIzql+b9jGqX5q5FoC
HQp6VeqX5q5DVL81ci0AOOhT0q9UvzVyGqX5q5FoAdCnpV6pfmrkbRwratKpmvgiyAOhT01p0404KMV
kkbAFax2RLqwhcT009GXf8AEj6qfmLkWYDOcVJncwrNVPzFyMaqfmLkWglnQp6Veinq5i5DVT8xci0AOh

T0q9VPzFyM6qfmLkWYB0KeIzqp+YuQ1U/MXIs wNHQp6RrWzbJtPOT3tkkArStYrGoaVacatNwks0yvI
hW39NTZ8UWYDm2Ot/tV6ql5i5DV T8xci0BHPQp6VeqX5i5DVL8xci0AOhT0q9UvzFyGqn5i5FoAdCnpV
6qfmLkNUvzFyLQA6FPSthhSUs5zzXBFjGKikluRkFd1pWv0EG4w6NWo5xlot7ycAtqxaNSq9Uy8xchqmX
mLkWglz6FPSr1TLzFyGqZeYuRaAHQp6VeqZeYuQ1TLzFyLQA6FPSr1TLzFyGqZeYuRaAHQp6VeqZeYuR
NtraFtDRjt b3s7grquKtZ3EBwuraNxDKWxrczuA7mlmNSq9VPzFyGqn5i5FoCMuhT0q9VPzFyGqpeYuRa
AHQp6VeqpeYuQ1VLzFyLQA6FPSr1VLzFyGqpeYuRaAHQp6Veqn5i5He2w+NGppyek1uJoKsYaRO4gAA
agAAAAAAAAAAAAAAAAAAAAAAAAYDAprl+Xq/UuSmsv5er9S5JD
HD4gAK2UeIdtqfT8EYk4h22p9PwRiPIZPKQAEcgAAAAAAAAAF9T22q6Nr3dhQneheVaCyi848GVtivFZ
7tqlK6g85afqnmcp1qk4qE5NpcSwp4nB7KkdE6XdCnWt5VElmlmmg7mm43SW1TpQp/6bTb37czW6o
W85yIUktPLdmccJ31F6EfEe1v0QdzalxROmtRp1qrjOWS7su8njZRejnHP1IFpbSuJvJ6MVvZKlaWIOWj
Oq9L1QcY98d6LrD4Kk6lHNZLPLiRbGnGrcaE1msi3gou3yg9KOWxIxh3bF6MOr0iLR/UyVhQ01J7Ipbszn
Gha9aaTWSSa/V37RitSUdCKeSe8r6S/1o+oL2rW3GIxteFOpSyqPKPrkU97Tp06qVJ5rLjmWOI9kkUy3BP
yLd9aC0wr9mfzf4KstcJ/Zn83+CM/x/NCu05Xs4ra2ybRw+IThpV9r9diOUEnizz4hTFZSUIKL/S3tK1isRE3I
v1S0rJqGWfwZXVvK3novanuZmzk43NPReWbSZZYmoujBvxIo dRkp vWpR7Ow6SCnVzUXuXEkdXs29D
9Ofqda7cbN6GzKOwpE3nmm8w6txx6jSZeWPQR04POPDgZw+3p1ozc1nkyenp2K09rcNvli4TuqeoXhX
nH9dHZ21KTdRrJ7k2YIY0K0G6Tyfc0yJiTbumnuS2HXCpPjxz2Z5Ai1Zvx0g1KcqVRwktqNSbiiSuU13xIR
HlvHG0ws8Kh+mc/oRb+endS+Gwm4X2aXzFddP/AKmp8zK2ydsUQsJWIHqeno/q0M88/gQbKWjdQeff
kWsuwf8A2/8ABTuf34eoXLEVtXSxxaC6KE0tqeRVlxifZH6lOhLnPH+0ywpuqkm6jyaay25FnXhCpScaryj
65FHR/eh6otsR7HL1QaYp/wAT2Qa1vB3MadDamuOZLjY29GGdV5vi2cMKS6WbfcjGKSfTJPdlsQSOMV
5zCRKytq0M6Wz4plbVoypVdCXEkYZJq40c9jW464okqtJrewWitq8od3YUZUlksm8tpHu6FvQofoycs+O
0mV5uFm5R2NRKRtt5t5kXLNa9ohgsMKppzlUfc thXllhMs1Uj37CssOucNcRuJqt0cJNJLbkzfDbiU3Kn5
5bVmR8Si1dt9zRvhUX00n3ZBpFp6rXE6ahXzW6SIZOxVp1ox4lgkY5YiLyurGlSpwzpvNySz2mt1b28pyn
OS08t2kcMI96r6I4Yj2yXoivRNjoFE6aW1vK4qaMdiW9lh1S1opaeWfxZrhKXQzffpEK+IKV1PSe57COlitK
RbXeU2rh9KrT0qDyfdt2MrJxcJOMIk0WOEuThOLexEbEUldvLvRUyViaxeEu s7Ozo1baMpx2vvzKwurDs
UPr+SJgiJt3c42drT2T acvizS7sIKm50lk1ty4lfvK5VZSb25lzbScrKOlt/SVrWa33GIJ3ltbrq1hpbnlmV1Gn0I0
od2ZNxSpo040l37WGeP8AzE2QqV1VpSbjLe82mayuK0556cs3wZzRZ2VkopVay270n3EcUi151EpUG+q
qVTfo7Sj17zy4k6+vdPOnSf6e9rvIAdZrRPaP0AAMAAAAAAAAC3wrs8vmKgt
8K7PL5iw3weacACvoAAAAAAYe4yHuAp8K7bV+v/kuCktqissRmquxPZmWqu6DwfSw5khhitERqXYHHr
VDzYcx1qh5sOZWvKHYHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5
Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5s
OYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtU
PNhzByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
w5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1
qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5
Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5s
OYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcet
UPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82HMdaoebDmDIDsDj1qh5sOY61Q82HMH
KH YHrVDzYcx1qh5sOYOUowOPWqHmw5jrVDzYcwcodgcetUPNh zHWqHmw5g5Q7A49aoebDmOtUPNh zByh2Bx61Q82H
MdaoebDmDIDsDj1qh5sOY61Q82HMHKH Y

4z5lgBo6dfSBqj34z5ms8HouD0JSUu5t5liHuGk6dfSpwmrOfadvN7tpbFNZfy9X6lylc4fEABWyxJdtT6fg
EnEO21Pp+CKR8rJ5SyACOQAAAAAAAAAAAAAAOtvrdeqoLd3sLEbnUOQLeULWziKKzfzwzY
6G2u4NwSzXet5W3Rn633VAO0f8ApbnRqRUSTjzLC5tade306MUmImsu8jmuObROvtUgtba0hQpOpXS
z+PcQZ/8AVXGjTgkm8klwBbHMRH9cAW8ba3taWdTJ8WzMVaXCcYqP4yDvoz+57qy0indU09qbJ2KU4
QoxcYpfq7kR6dONPEYRhJSipbMi0uKdOcU6uWjHbtK7x03SYUALqm7WvnCCi8u7Ir7+3VConH3zfYMrY
prG9ooJ9r1WNHTqpaSe3Mk06tpXejFRb9CLXFuPtTgmYhaxoSU6eyl7iRh1GnO3znBN596DmMUzbiquw
XDjZ28spaOl8TNa0pVqTIBJPLNNB30J/UqYHWhRIVrqnuee18C06K2tKa01H1e9hxTHNu8qYFvCFpczTgl
mu7l44nSp04QcYpegdTh1XltXAsLGy4dJV3dyO3T2aloaMeG4EYtxuZ0qQWN7ZwjTdWkskrSK4M7V
ms6kAaCAA
Bb4V2eXzFQW+FdnI8xYb4PNOABX0AAAAAAMPczJh7mBUYT26r6P8lwU+E9tq+j/JcEhlh8VfjeIvDMPI
Wik5t6MU+j5W1oY1jUZV4V5KGeWblkj1GPYbLE8PdKDSnF6Uc+88rb3GMYHnSVGSp57nHSRWqfZW
WO2V/RhOpKVKUv1PPSSxxPVVa1OhDTqzjCK3uTyPOYf7WwrVY0r2l0Tby00819Swx3D7e/tqbr11RjC
Wen8AO2vMN09HrlLP5idTqQqwU6clKL3Nd54u7w7A42s+gvv9WK2Z57Wd/Y26qdYq2rk3DR0kuAHpa
mJWdk46Cpc041c0tBy27TnUxjD6VXo53dJSW9aW48b7RRIU9oakle9jxs9SwwfZWFvhk7hV5SqwpjSzW
xgeqnd29Ogq060FTf8A3N7Djb4tY3NT06NzTIN7kntZ4nBbCri9dW860o0aacmuHodMdwdYPVpTo1Zsj
Pc9zTA93VrU6FKVWrNQhFZuTexEZyRySpSqq7pOEXk5aWxFbcXE7r2NnWqPOcqO1/HM85gGEa1qzjU
qOKGK15d7A9rb4rY3U9Cjc05y4J7TnjrywW6y8P8AlHmMdwKOEOoXVrUlo6WTT7mWdO+nf+yFadV51
Ix0ZPjk0Bw9iW31nN8D01e5o20NOvVhTjxk8jy/sVJRjdSe5JMqqlzHGMYcr6v0NBN7+5cEB7ahi1jcT0KV
1TlLgpbyYeDxWzwilbdLht3nVi1+lTV9F7L39S9w3KtLSqUno58UBZ3sKISqzwovKo45ReeW08fcYfj1tbzr
VazUIJyeVTuPbkHHP4W8/4pfgDxmHvF8SnKNtcSbis3nLI9NgFniNrOq8QnpJr9P6syp9ie0XHy09DjI88P
wurWh7/ux9XsA7XO12dpLRuLinB8G9pta31tel03rQqZb9F55Hh8KoWF5KpxxW70ZN717c38TW9dvh/
SrYVdOpFbdmez4MD6BUqQpQc6kIGK3tsg68w3S0euUv/AHHnfaq6rV7Szms1SqQ0nlu0jnZWmAV7aC
q3E4Vmuk3vA9dO+to2/TdPT0O6Wexnm8J9o69fFHSu6tOFDbt3ehYLAwWpg6taddygnp9lu88phGHw
v8AFXazm4xWe1fAD6JRrU69NTpTjOL74vNG5Fw2xhh1nG2pycoxbeb+J0vJyp2dWcPejBtAeYxz2jrdZdp
h7yyei5ra2/gRY4Pj1WPTac03tylUyfl5eytKFbHM6m1wi5Rz45o94B5jALzFuvStLynKcl+9KSycS/uL22tF/
wBRXhT+Z5Eg87jGEYfVv3c3l30KktsOIFrRxewuKihSuqcpPctLeTT59i9phlCnGph130ks8nDbn6l/Y3txX9k
atWDbrQTin37MgLavi1jbzcKtzTjB1ntN7bELS7eVvXp1Hwi9p4TCYYZVnPWIScW3sa3HpcGwvDKV67m
xuelyjkoZviBeVatOjBzqzUlre2yJDGcOqT0l3dJvcv1bzyWO30r/GXbVKvR29Keh8FlvZ1u8PwNWcna33+
vFzrPPKTA9qmmms080ZPMex2IVK1OpaVZOXRrODfcj04EaWI2kbnqzulKt4M9pjPL1vZivUxI3KrQVFz0t7
0keoApPaO6xGhRjGwpvRI71SKza+BRVMKx5UXcTqS2Lsa6TbyPcFnj+NUsPt50YPSrzWSiu74sCD7L41X
uq0rS6k5ySzj7/Q9OeT9kMMqxqyvq0XFNZQz7/iesAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAe4B7gKay/l6v1Lkp7L+xq/UuCQxw+IAcIhIhban0/
BGJOldtqfT8EUj5WTylkAEcgAAAAAAAAAAAAAAWGE5ac+ORXnShWIQqKcfquld0txtErO7dr
Gp/rxbfHac6V1Z0pN0003v2M6K6tbiGVXJPhlxKpZUYvQUXmtyK9c68omEG9rQr19KnuyJ2GKoqTcn+ju
zK+jGnOv8ArahDPPaybd3k10ejt5J5rLnwZY51M3mXTE41HRTh7q94h4a11pejFnewdFwuJJnbM33kO
ro0bjToTUIlnmsgt5iZjCVi2a0H/2IfBSb/Qnn8C2heW9emo1sk+9Myqtnbxaho7eALUi88olW2fa6fqWGK
vK3WXflhwxrxnexqNKEczviNeVopU5qTTz2BKzEuTg0axeV3DImYsv9KHqQbsUYXMZSaSXeSsSr0qtOCp
zUmn3ESkx0piW1tY0IQVWvt2Z5cDajOy6aKpRelnse0W15SIQVOq8mlk8+8J2VvNSi85fgrWOOo46bYr2
aPzlzhnZfqcsRr0qtuowmpPSzyRnD69Knb6M5qLz3MG46u9oN1tuZt8S0w552a9WVVeSIWm0802T7C4
o07ZRNuInm9jDLFMRzIpYySvqsXve4YpCenGWTcct/AiOo43LnTffmmWEMRg45VotPhkFi1bVms9kXD
6cncqSz0VvZJxb3Iep0pXcKtZQpQeXezi0lowXftDvUVxTqUuj1SOa/TobcvQgueHrY4P7mtlfkjHo6ubj3P
gSzDrm9KThmFi0XiNaazvbfoHTjnlkthVE2+rW84KFJbV3ohEefLMzP2AAMgAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAt8K7PL5ioLfCuzy+YsN8HmnAAr6AA
AAAAGHuZkw9zAqMJ7bV9H+S4KfCe21fr/kuCQyw+KtxzE3hdmq0KenJySye4rqXtdZVKeVeIODY2rLNH
oZ04VluM4qUX3NFFUwHDKknJ2sE3wK1eNxGrDFsWi7C3cVLJJb3xLD2thWpK0pycnSUMvg2eqtcpLps
1CNNve0jtXt6VzTdT0vTjOL7mgPHKtgEcMShRc7hwyyeeka+xv8Alz7v9N/IhqKODYfbz06VrBS45HW2w
2ztKrqW9vCnNrJtAeOxn/1Wv+SH+D1+j/xNxn5bM1cLsq1x09S2hKrmnpvfmiTUpwq03TnFShJZNPvA8
h7E9suPk/yd/bf9m1+ZnobxD7Szk5W1CFJv2Nx7zN1Y216oq6oxaqQ1aXcBRx/9DS/4f8lH7P4wsKrT6W

DISqb8u5nq8ZoU6Hs7c0qEFCEaeSiu7aUnslaUrmjcwuaKnFtbJIDhj+PQxSjC2tYS0dLNtra2WVKznY+x9a
FRZVJx0mn3ZtFx4RYWtTTo2tOMuORKq0oV6UqdWKICWxxfeB5X2Kipxu4vc0kyoq2ywjFnG+odLSzeS
ezSXFhvLWxtbLS6tRjs0t+j3m1xa0LqGhcUo1I8JIDyNbEcAjSzpWLnPw5tZHpslp2qsoVbOj0UKiza2mtPB
MNpz0o2INNbsVsJ6Siikkstu4DJBxz+FvP8Ail+CcaVaUK1KVOrFSJZSi+9AeR9ie0XHy09BjtjK/wurRh7+yU
fptyJFrh9pZOTtqEKtIvlce8kgfPMLr2VpUnRx00csnv25osoXuBVrylRpWDIGbyctuz6HprnDLO7edxbwm+
LQtLsrSWIQt4QlxSArsZv7HDYULSvbKrSIH3fCjz+IVMAqWsnZwqQr9y25Z/U9tcWIC6jo3FKNRf7kRI4Dh
kJaStKeYFX7HxruwrKel0bf6M/8FFh1yslx2dS5g0ouUZLLdmfQlQjTgowioxW5ljXWF2V5PTuLeE5cWtoD
DsRo4IQdWhnop5NMISSIFxazTOVraULOn0dtSjTjvyR2A8De21z7P4r09FPo9LOMstjXAuYe2Ns6Wc6E1
PLalxPRVaNOtBwqwU4vuaK9+z+GOWI1WHoBX4P7QXOYIKI1f/AEXua/7fUosQlFe0tR4IpOkqrzW33e4
93QtqNtDQoU4048EjldYdaXu25oQqNbm0B43G62ESToww2k9PPNyyexfUt/Z25hZ+zNSvVWcITk2uJbx
wfD40XSVrT0JbWst53oWVvb0HQpUoxpt5uKWwDykr72du0517aVGo9+jn/ghYJFv2ipuwU+hUu/w/E9
dUwLDak9KVpTzfAIW1nb2kdG3oxpr4IDxWPWU7DGZXFSlp0Ks9Pbuee9EjWHS/0O1KWnl7ub/ACexq
0adem4VYKcXvTRB1FhmnpdUp5+gHD2ejZVrbrVpbDkX6Zby4NaclUoKElqMVuSngPMVfairSxd2qoRd
JT0W9ukenIssNs5XXWZW8HW36WW0IAVHtDjDwu3iqcc61TNrb3I8baXNCV91jEtOttzyT3v4n0G6sLW
90etUIVdHdpdxG1Fhf8A/CpcgleH+0tpdXFO1pUpxctkeCL0hUcHw+3qxq0bSnCcd0ktxAAAAAAA
AAAe4B7gKey/l6v
1Lgp7L+Xq/UuCQxw+IACtlHiHban0/BGJOldtqfT8EYj5WTykABHIAAAAAAAAAAAAAAAA
BkwZAwdIAwN+8yAbgyAMAAAAAAAAdjKMKsZTWaT3Fn1iyrLOajn8Y7SpAaUyTXst1c2luv9PLPglv
K26uJXFxSexLcjkyBfLNo0GQAzaAQAAUAAAAAAA
AAAAAAAAAAAAAAAAAC3wrs8vmKgt8K7PL5iw3weacACvoAAAAAAYe5mTD3MCownttX0f5Lgp
8J7bV9H+S4JDLD4gAK1AAAAAAAAABiUVKLjjp70zWFOFPPQhGOfBZG4AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAe4B7gKey/l6v1Lgp7L+Xq/UuCQxw+IACtlHiHban0/BFJWIdt
qft8EYj5WTykABHIAAAAAAAAAADKj7k2Y3PJhdBkbxk1vCdWBpremGmt6YXQBk
8s8tgW3cE1IDLi1vTRjeAaya3oADKjJrNRb+hjJ8C5sEnZwbQaY8fOdSp9GS3xa9UdjW1aMNOV8NqPEkXd
5CskowcXGWZvVxCMrdxVNptZbdwdckRuNoCjrNRbXoYaa3pr1RY2V1TUYUdB57szbFUIRhkv+4L0q8e
USqwAGAAAbU6c6stGCzaWYXW2oMyhKDylFp/EwEAQAZSzeSDi1vTRV0wAZyeWeQRgGUm9yCTe5A
YBJvcjAXQDbQllnovkag0Azk+BglGUuLYImEm9yJmGL/qXmv8AtDqteU6RNcfhfl1Lq6uoW8IfwzzWZTe
820u8OsllrOolgGcnlltDi1vTQzsAACZYKg9PptHuyzJ/VreVNyjBNZbA2rim0bhSAzL3n6mAxbQpzqSOY
RcmZqUqlJpTi4t7jpZ1+r1dLRck951u7xValNqDUY7doaRWs13vui6E/BLkYaaeTWRd21zC5TcY5ZcSsv9I3
MOr4orXIEowADEAAAAAAAAAAAAAAAt8K7PL5ioLfCuzy+YsN8HmnAAr6AAAAAGH
uZkw9zAqMJ7bV9H+S4KfCe21fR/kuCQyw+IACtQAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAA9wD3AU9l/L1fqXB2X8vV+pcEhjh8QAFbKPEO21Pp+CKSsQ7bU+n
4IxHysnIIACOQAAAAAAAAAAOtVKEKOXUsce/M5Emxt1cVXpe7Ha1xDqkTNoiEx4hQi8
oxbS70jtKnRu6Gkktu55bUcbirbW0uj6BN+iJFtUjUoaUlaC4ZFe6veZrKpto6N7CL7nkW9aNKKU6iWUdq
Kui/Jf/AFsl4q30MPUMseq0mUe7r069a10e6L27CyqU6bp/rS0VtZRU/wByPqi4v3lZSy+AMdtxaZRL25o1
KShS3p8Deld29KjHKH6stqSK2K0pJcS3dC3tKOIOGLlvzWeYc0m1pmzehc0bluGjk+DIN3SVrdRIDYntJvrc
0qtbRhRUXxyRyxf3qXow7v3ptluKcbizbglnlmiqoU3VrRh8dpYYXW0oOm+7d6HahaqjcVKnc93wBNlyat
DliM40qCpxSTI+DtYr/ooFXe1eluZNblsRaWPYofUFLRbjKme2r9S4uYpWL2f9pTv976lzc9h18ocYo7WVdl
2un6k7Fv2afzEkY7VT9Sbi37NP5h+in/KyqCWe7aDta1Y0K2nJNrLciPPH33ctGXhfl6W9aVvV04pN5ZZM
n6zpeXL7EOjTV3dyTbinmw1msRMcJTI3lvX/AE1oZeoqYfTnHToyyz3GdCztVnLKUvjtOVbE3lo0YJL4lazN
df8AogNZNrgYMt5tt72EtJpcdhHk/fZY0bu3pUI/o/V3pIkULijctx0ctm5o06vQtaGnOClIvzWeZi1uKNWso
060i+OSK9tdxMRKHiFvGhVTgsoyLG0hGvpBNJ5x1L2bOfUk2vYoNeEJWsRklwuK9vSpTow97JrYjfDlp2u7
P9TKqbznLPiWuF9kfzMiY7csjarWtrbOOX6ntaSIVnWowqzlVS27Uzhcv/qanzMI4fawqQdSos9uxBzub31
H6dliNByy0XlyxF9bnwnburBJSSzzXejnUuraFRwVum08vdRLrPOzm0sk4PZ9CtY/1ExKNhk41KTpySzX4IV5
S6G4kktj2oWlXobiMu7cyzurbrDptdz2+gZRHUpr9w0sqSoWjnPLN7WR8Pn0t7Ob70dsSqqnRjSW+X4I+
F9pfyh1bUXrWP02xZf6sPQ64XFOILPicsW/dh6HbCv2Zeolj/wBpdKte3tajTX6nteSN8qN3RzSTT+xV3va6

Ro6VeSz7/gVV9ddYnox9yL2FaZKUrXbpSw6VWnGaqqJrPI31VPzYkeF9WpwUltJL4Eyyq3FxLSm0oL4bwI
Yx27acnhU/MjyK9rJtFze3SoQ0Yv9b+xTBxmlZ1UL2x7JT9CiL2x7JT9BDv8bylVX3a5+pHJF92ufqRyMb+
UgBNw616WfSTX6F9wlKzadQUsNnOmpyko59zRFqwVOo4qWII3otL+66KHRwf6n9ioDTLFa9oACbYWv
Sz05L9K+4cUpNp1BSw6dSmpOSjn3EWrBU6jipaWXei0v7roafRw9+X2RUMO8ta17R9gADFb4V2d+pyxf
fA64V2d+pyxffAr2z/xVp0o0nVqxgnlpd5zJWHR07j8MyPJWOUxDtqqXmLkNVS81ciTf150IRdPLNkF4jcc
Y8ivTaMVZ1MONxRdCroZ5nehh8q1NTU0k/gRqtWVaenPeXNI+mzh6BnipW9p9leqp+YuRzr4fKjSc3NN
L4CeIV1OSTWWfA51L2tVg4SayfwItulHaiRzej+9D1Rob0f3oeqDCv29EtlyWK9mXqTVuRCxXsy9Tp9LJ4S
pwk20lvYJ+GUVOcqklno7jl8+leU6ZoYY5JOrJr4I7vDaGXevql+5IRSht95rPPgV7ncL9Tc0V6LdOnbW295
axt8tGeefczhRpSrT0YLNirVnVlnN5vcW2H0VsOKWX6pbWGdaRkt2+nKlhcfHOOpjt/A3lhtF+62vqR7u6qz
qyhSzUU8tneR1Wuab96S9Q7m2OvbTSvTVKrKClpZd5zMttt72YI8377PQ0P2IehwxLskjvQ/Yh6HDEuyS
On0bf81KTLK0jcRk5trLgQyww2tTpQlpzUc33nLxYoibd3bVILxSGrKXikRbu6m7iXQ1XobMsiVhzqzjKdsb
aexJleivTtbjEON1Y06NFzjKWfxOkMNpyhFuUtqzNMVre7SXqv1qirZKNVaWh/gJrHylrql4pDVdLxyK9X
Nw5JRqSzblLqmnTorTlm0s22Fp07/AFcmvKct6+hFtrLPacDrc1emrylyOQeS+uU6egtuzw9DhinZH6ne27
PD00GKdkfqH0L/APNTADvl+al2tjkutOT0YExYbQS2uT+p3tX/ANLTy8KKi4dd15aWlnmV7JrXHWJ1tKrY
ZIBypSbfBlc002nsaLyzc3bR6XeVd3oddk17uazl4yUjUTH7dbXD5VY6dRuMeBKeHUN2bz9Stt6H/T4bCk
br9Lm9LSzK7tWuOl7bSrjDnTjpUm5Jb0yBuPQUTj0Yupvy2IDVa6WWjuDPNSK6mGpZYRvq/T/ACVpZY
Rvq/T/ACSPTzg/6Qsw9wD3HT6Kns5er9S4Key/l6v1LgkMcPiAArZR4h22p9PwRiTihban0/BGI+vK8pAA
RyAAAAAAAAAAAbqjUks1CTXHI0LuxS6nPgVrjpznSojQqzWcYsa9DWdOUHIOLT+jZVMShTm4Qhml3
ndxp3lvmIv3fAO+lWe1Z7qWMJTeUU2/gbOIUTScJZvuyJNhFwxF70mixuKtOgtOS27kEpiiia7mVRS6Sh
Xi9B6fcuj3vLmtVpqM6ThHPe+8x06uMQpTUCsthJxXs8Pm/wFiP821PZWxpVjrOMJNfBCNGpN/phj/Qt
cMX/AE31Nat/ToVHCMM8t+RDpVisTMqudOcPfi16o2oU3UqxWi3HPbkXP+nd22eSakuRX2VdUKrpuO
ek8s8yk4oi0d+0ut5Z06dHSpQeln6lc04+8mvUvbmsqFPTaz27inuq6uKumouOzcRc9axPzxMxjKTiyim38
EYJIndRoQkpQ0nvWQYViJnu4dWrZz9HLkc2nF5NNP4InTxJymlOnkn3m2JUYyo9ll+qJW04qzXzdZvcKc5
56EW8uBIUajk4qEm18Cwwn3anqd7m7pW0stHOT2tIFcUTXIMqhU5yk4qLbXcHSqKWi4SOuGRNw+fSX
dSWWSe3ImXNelbPSks5PYsgVxRNeUypZwlCWjJNPgbRoVZ+7TlyOs7pTvFWcf096ZKeJ7coUtiDmtad9y
rp050/fi16oQpzn7kW/QuoSheW70o79mTK+zn0F44Pc3ohbYoiY79pRJrcHijNP4mYxc3IFNv4FhilHOM
aq9GYwqj71WS+CCdKefFBnTIT9+LXqaki+q9LcPLdHYiORnalidQnYXS0qkjWyO40xOq519BbokvDlpW2
fFlc/9W7elucg3t2xxEftzhSnU9yDf0NpUKsV+qnJfQuKrdvRXRU9LLuRF1jjqpSyKk4q17TKujCU3IFNvgjb
oamlo6Es+GRJw5uV421vTLC6ulWYUms5PYhpK46zXIMqeVvVis3TlI6HMt7a+jcT0JR0W93xlul26pSVSG
xS2ZELY448qyiQpzn7kW/RG0rerFZunL0JdG/jToRiqecIvyJNteKvPQITaz3FWuOk9tqd7DKTbySbfwJmJ
UFTqKcVlpEuzt4ULDvJr9WWbbGnMYpm019Kzq9bLPo5ZehzcWnk00/iWTxSKnl0f6eOZzvq1vWiuj2z4o
aW1Ka3WUFRcnkk2/gdHbVks+jl6FpRpQtLZzlHOWWbZxhiLc/1U8oPvGljFEeUqzdvBZ4jbxndNVorJ9+R
WEZXpwnQAA4AAAAAAAAAAAAAAAAC3wrs8vmKgt8K
7PL5iw3weacACvoAAAAAAYe5mTD3MCownttX0f5Lgp8J7bV9H+S4JDLD4hyuP2Knys6nK4/YqfKyu7fU
vPgA5fKCZhna//AKSGTMM7X/8ASiaYvOEjFv2YfMVIP9yPqWeLfsw+YrKf7kfUrTN/0X9b9ifynnZ0Fb9ify
nnw6/J+4AAR5VzhnZfqTCHhnZfqTCvqY/CFHiPbZ/T8EYk4j22f0/BGI+dk85DrQt5156Md3e+Bm3t53E8o
7u9InOdOyo6MY5y4cQ7x49/6n6YrTp2VtoLflsXeynbcm5PezerOdabnPPNmqi5SSSbfAqZLzedR9MJNtJ
LNstrK1VCPS1dksu/uFpaRt4dJV978EW8up13owzuF9w0rWMccrbS9r9PV2e7HYiMZafeYI89pmZ3Kvh
vbl+jLspMN7ZH0Zdlj6e78bwRr/sc/oUZeX/AGOf0KMSx/J8mSVh7hGvnNpZLZmcKNKVaooR3s3qWdeE
mujclxQhjWj/UQuG6dxBw0lJPgysvbLoI6cM3Hd6GbG3rK4jPJxit/xJulyStJJ97Qeqf8A0pMzCIW9F5Qq0
Y0YJSitm7MrOpVnSjUis0+7vNFb1s8ISln6BjSbY/0sbmxhW0pxeU+PEqZrcZOL3ovraEqdvcNT3ktpS3bT
uqjju0g6z1jUWCi9seyU/Qoi9seyU/QQfjeUqq+7XP1I5lvu1z9SORjfyl2tqDr1VFbu9lvUnG0t/wBK3LYkiFh
9elRhLpjZNsIO9tnvmmV6cXGtd77oNpDrN3LpotprPab4hbQpRg6cMs33E+jXo1Z5U2s/gZrVadJJ1Wtu7
MOunXh9qW3oOvVUFml3lxOubW32LctiXeRqF1QhWqy0kIJ7Du723e+aYTHFaxPfugW0es3bdaLefE6Y
hbQpQi6cNrZNo16NSWVNp0N61WnSSdTLJhenWaT3efay2MHa7nGdxKUPdZxI8UxqdlfCuzv1OWL74
HXCuZv1OWL74Feyf8AirSxh84QrtzaWzeRDtQtp3GloZbOJHkpM8o0tpV7aa/VKD9Q7e3rxzjGLz74ITK1r
ReTpy+iJ2G0atNyc01F7kyvXW82nVoQ7u26vVyW2L3MsqNxRhrJFzjsRHxVp9HBLOWZGqWNaCTUDJP
gHHfHeeMLNO1m9iptv4HC6s1ODnSWUlty4kCnb13JaMJRee/Iu89Glnj7ltDSuskTyh543o/vQ9Uava2bU

f3oeqI8Ufb0S3lhYr2b6k1bkQsV7N9Tp9LJ4Spy1wr9iXqVRPwys0TdOWzS3HLw4Z1eE6vUpUf11F+rLjCT
SjeUa8tBjp8Gc8Rtp1tGdNZ5bMiNZ2dVXEZzi4qO3aV6bWvF9RHZ0xG0hCPS01lt2pE+h+xDhkR8Tmo2z
j3yM4fWVSGo5/qjsaC11GSYqgXVC2loKO3vyR0pVKN1TeW1d6aK+8tK3WjhFyUnnsJOHW06MZSnsb7
glbXm+pjs3tDoK+S917URyb1lRTrqK/7VtIRHlverFUPQ0P2IehwxLskjvQ/Yh6HDEuySONut/zUoAOXzW0
luc4xW9vlvYRjQoJbIFFdhIHsqOo1sjsRIxOtoUIBb5FevFEUpN5Vtao6tWU33s0MIEeWZmzs8No9JW02v
0x/JNxGt0VDRT/AFS2G9jSVG3WfvPaytv63S3Dy3R2lr1T/wCeP+yjAAPI9Bbdnh6HDFOyP1O9t2eHocM
U7I/UPpX/AOamAMxaUk2s9pHzv2ubGIKjQ/XJ7duXAxO+t1Jpyza4I77KtD9L95bGU870vGTWg38UV7b
zNKxFYW6lGtS/05bGtjRTVqM4XHRy2yb2PiWlhQlQovT3t7uBFuasNYweeyOWbDnJHKsTKdQg6FBKpP
PLj3HJ31tpb/rkdr1DrW8oxe1rYU3U6+ll0b9Q6yWtXUVhcVF09BqE8s1saKOcXCbhJZnb7taTo28YSe1F
TeTU7mbjuzyzDPPG6xM/bgWWEb6v0/yVpZYRvq/T/JI+2eDzhZh7gHuOn0VPZfy9X6lwU9I/L1fqXB1Y4f
EABWyjxDtt6fgikrEO21Pp+CMR8rJ5SAAjkAAAAAAAAAU7lbZw9CkLuy2WcPQr0/j+Sqq29SnNxcW
+GSLsypyowY01k3tOEMTSWVGb4pnG5xCVWOhBaMXv4hazSn+oltaTUsRk1ueZti3vvllrXvvWU2s13
HS8uo3Li4xcculcc4nHMftztO10/Un4t2eHzf4K2jPo60ZtZ5Mk3I5G5pxioOOTz2gralxzCZhvZfqVdf9+fQS
bW+jQo6Dg28+4iVJ6dSUkss3mQvaJpEQt8O7HH1f5KyOy9T/wBxItb+NCgqbg213ohSk3UclxzKt7xMV1
+lviUXK1/Sm9pUNNPJpr1LGlii0MqkG3xRGvLiNxJOMMsKQyzS3+oIGLawoU40FUa0pPaVJLTL6VCOhKOI
HuDjFMRO70+sJuroRo9+RJxDsc/p+SLPeaa206X6uLNbi/jWoOnoNN95W/UrETEy6YT7tT1luldrkbWd3
G2UIKLlnwONzVVau5pZZhla0d0IScK/f6GcV/ej8pw57hW9VycW81lsF5cK4qKSi0kstoTIHS4/tth9GNavi
Paks8ibd3PVWoQpprLMrKFaVCopx+vxJ7xGIKP66TbDvHavDW9SIwtWVajpTho/AqLI5XU2t+ZMhiaWe
IB5dyXcQZ6VetKUlt57ckDLeLVlnutqbV3ZZPe1k/U1qNWljkt+WX10WFqptFTU4tR+JyxOtpVVTT2R3hp
NtU5T9oOebzYBkjwrjDuyR+GZUzTVV8cywwupnCVPvTzlt1Hq983Int0g9V+9Kykvxa9JKNWk38STRqQu
4PSptfMiPHEoNZTpms8TSj1Tp5fFlaVvWPu22LWCpYjOK3LM6YpRnNRIBZ5b8jhh8pTvXKW1tEy7u3bVY
pxzi0ErxnHO/pBsKFR3EZaLUVvbJOKNONOHe5GJYpHL9EHn8SBVqzq1HOzbz+wcTala8Y7rbRhaWunCC
bS5nO1vKleso9Fku9nChiWUNCrDSyW9GzxGMZLo6WUc9vxDTqV7altiz/TD1JP79plB747CtvLyNyoqMX
HRfea2t3O32e9HgHHVrF59S5uhV09DQlnnwN6ltVoNSmtFE1YpDL9t5kS5vJ3GzLKPAOLRSsbiVu5t0N
Omlj5bEQevVtLLq+0j21503Wj70OBKeKU0v25NjbXqRaPvTjdXdZwdOpSUdJcSCSlq7dw1+lJlh58lzt9g
AlzAAAAAAAAAAAAAAAAAAAAAC3wrs8vmKgt8K7PL5iw3wea
cACvoAAAAAAYe5mTD3MCownttX0f5Lgp8J7bV9H+S4JLD4hyuP2Knys6nK4/YqfKyu7fUvPgA5fKDvZ1
429fTmm1llsOAC1nj04Tb67p3NOMYKSaee0hwejN9xgBbXm07laVMSopypiozzay3FWAFvkm/2AAOF
zhnZfqTCHhnZfqTCvqY/CFHiPbZ/T8EYk4j22f0/BGI+dk8pWFnfUqFBQIGTfwR21pQ8E+SKkDbuM94js21
nQ8E+SI3W6fXem0zaOW7lhAbSc1pW2tKPhnyQ1nQ8E+SKkDbr5F0+8vaVehoQjPPPaiAGV7zadylYb
2yPoy7KTDe2R9GXZY+nt/G8Ea/7HP6FGXI/2Of0KMSx/J8na2uHbz0kk8ywhiVJr9SkmvIJtITJanaFvLEqK
WxSbK+5up3E85bIrcjgBtbZbW7LGjiaSSqQ3bM4kjWNvI38imA26jPeIWfziWknGims+9IfvADK17W7yF7
Y9kp+hRF7Y9kp+hYb/AI3IKqv1z9SOSL7tc/UjkY38pAAHCTZXEBeq5TTaay2G99dwuYxUFJZPvIYDvqWi
vEMmAHCRZ1429XSmm1l3HW+u4xEIqCksn3kIB3GS0V4gADhb4V2d+pyxaffA64V2d+pyxaffAr2z/AMVa
Sr8duFxTiyKZl8lbTwdtliVB71JfQ1qYpTS/04tv4IU0C7a9e7rK4qSrqrJ5yTzJ9PFIZZVINP4FWCbcVy2r
3hcPEqCX/AHciFdX8qycILrj+SIAtmtaNBvR/eh6o0N6P70PVBIx7eiW5ELFezfUmrciFivZvqdPpZPCVOE
2mmmnk0AcvmJ1LEqkl5TipfE3lijy/TT2/FlcA16t9a26Vq060tKb2mKVWdGelB5M0AZ7ne1jDFGl+unn6M
1q4nOUcoRUfiQTA2061/bLbk83tbMABn+3oah7EPQ4YI2SR3ofsQ9DhiXZJHT6Nv+alAbY+amW9+6FN
QVNP45nC5ru4q6bWXckcgHU3tMaDaElGabWaT3GoDnafLE5ODiqaWa35kBgyHVrzB7YBkwHL0Ft2eH
ocMU7I/U723Z4ehwxTsj9SvpX/AOamABHzUi3vKIBZL9UeDJSxTjT+5WmQ0jLeP2mVcSqTTUEo595Cbz
bb7zJgObXm32IUL+rRjo+9FdzJGtVl+3t9StA27jLeI1tMr4hVqx0YpQXwZDADI1ptO5Cwyjfv+n+StLLCN9
X6f5ENMH/SFmHuAe46fRU9I/L1fqXBT2X8vV+pcEjh8QAFbKPEO21Pp+CKSsQ7bU+n4IxHysnIIACOQA
AAAAAAAAACfQv1SoRp6GeSyzzIADqtpr3gAAcgAAAAAAAAAAAAAAEi0uurNtwUszg
YCxaazuFjUxTOLVOGTfeyulJyk5SebZkwHrv2t9gAdh1oV3Qqqa+q4nS6uo30T0NFr4kYB1ynXEAACu1rX
VvV02s9mWRvd3SuZJ6Ojl8SMA65zrj+gAByAAAAAAAAADBkAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAABb4V2eXzFQW+Fdn18xYb4PNOABX0AAAAAMpczJh7mBUYT22r
6P8lwU+E9tq+j/JcEhlh8Q5XH7FT5WdTWpHTpyjxWRWk94edMEmpY14TyUHJdzRp1Sv5bOXzJpaJ+nE
Hfqlfy2Y6pX8thOFvTiDt1Sv5bHVk/lshC3pxB26pX8tqlfy2Dhb04g7dUr+WzMbOu3l0bBwt6WWGdl+p

MOFpRdCioPfvZ3K+ljjVViVHiPbZ/T8EYs8Qs5zqOrTWee9Elqlfy2HgyUtyns4g7dUr+Wx1Sv5bl44W9OIO
3VK/lsdUr+WwcLenEHbqlfy2OqV/LYOFvTiDt1Sv5bHVK/lshC3p0w3tkfRI2V2H2c6U+kqLJ5bEWJYe7B
Wa17o1/wBkn9CjPQ16fTUZQzyzRSzsq8ZNaDfoJZfkVtM7hwB26pX8tmeqV/LZhM4W9OAO3VK/lsdUr
+Wxo4W9OIO3VK/lsdUr+Wxo4W9OIO3VK/lsdUr+WwcLenEvbHsIP0KqFlXm8tBr4suaFPoqUYLuRYen8
esxMzKmvu1z9SOWN/Z1JVXUprST3oidUr+XIMclLcp7OIO3VK/lyHVK/lyl44W9OIO3VK/lyHVK/lyBwt6c
QduqV/LkOqV/LkDhb04g7dUr+XIdUr+XIHC3pxB26pX8uRIWddvJU2U4W9LDCuZv1OWL74EuzoOhQU
ZPa9rNMQtXFNOHvR7ule2az0tKUZnfqlfy2Y6pX8tkeLhb045jM7dUr+Wx1Sv5bBwt6ccxmduqV/LY6p
X8tg4W9OOYzO3VK/lsdUr+WwcLenE3o/vQ9Ub9Ur+Wzva2NV1oynHRjF57Suq0tM/S3W5EPFey/Umo
5XNBV6Lg9nxK+heN1mHnwTXhIZPY00Y1bX+Bzp8/pX9IYJmrj4DVtf4DR0r+kMEzVtf4DVtf4A6V/SICX
q2v8AAatr/AHSv6QwTNW1/gb08MqOa6RpR78i6lx9LKh+xD0OGJdkkSorRiktyNK9JV6UoPvK+hau6ae
eBNeGVk8k00Y1ZX+Bzp8/pX9IYJmrK/wGrK/wGjp39IYJmrK/wABqyv8Bo6d/SGCZqyv8Bqyv8Bo6d/SG
CZqyv8AA3p4XUclptKPeUjFf0sbbs8PQ4Yp2R+pKhFQgorclkaXFFV6Tg+8r32iZpp58E14ZWt2NNGNW1
/gc6eDpX9IYJmra/wGra/wGjp39IYJmra/wGra/wABo6d/SGCZq2v8Bq2v8Bo6d/SGCZq2v8Bq2v8AAaO
nf0hllhG+r9P8nLVtb4E+ztVbU2s85PeytCOO0X3MJle4B7ivcp7L+Xq/UuCnsv5er9S4JDHD4gAK2UeIdtqf
T8EYk4h22p9PwRSPIZPKWQARyAAAAAAAAAAAAAdKVCrV9yDa4hYiZ+nMEmVhcJZ6GfozhKLhJxksmu4
LNZr9tQdqVtVqrOEG1xNp2deCzcNnwC8La3pHAO8rStGOk4bA5Jn6cASVY12s9A41Kc6TynFphZrMfcN
Ab06U6ryhFs7qrwrte79wRW0/UloN6IKdJ5Ti0zNgjOu2qaza3hNTvTmDv1StpuCg81vOcqU4VHBx/VwQ
JrMNASY2NeSz0MvVnOrbVaKznB5cQvC0RvTkDrbUpVai0YaST2ljdWkXQ/wBKktL4LaV1XFNo2qQb1Kc
6Uspx0WZpUKtb3NriHEVmZ05gkTs8I5uGfocMsnt2EJrMfYYN6dKpVeVOLZ1IZV4xz0M/QLFbSOzrKn
p5LRyz3mtG1q1ouUEmvUtqmyxa/2kDDKujXcO6SDecVYtEe0OUXCTjZNbzB0xSlo1VNf928ghhevGdA
ADkAA
AAAAAlfCuzy+YqC3wrs8vmLdfB5pwAK+gAAAAABh7mZMPcwKjCe21fR/kuCnwttX0f5LgkMsPiAArUA
AAwZARjIGQBgGQBgyAAAAUAADYNgAQ2DYAA2DYAA2AAAAAoAAAAAAEA+vqGH0eluZ6Mdy+IEkHC
zvKN9bqtby0oPZ6M7gAAEAFAAAAQAAAAAAAGMgZAGAZAGAZARgGQBgyAFAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAD3APcBT2X8vV+pcFPZfy9X6lwSGOHxA
AVso8Q7bU+n4IxJxDttT6fgikfKyeUsgAjkAAAAAAAAABF7FOFsuhSby2loifSd7QprRjpRe3LgVvhtxmezZ
3N5Tec6ea79hHox63efr3Payxta9Ws2qtLRWW84ScKOJxyySktvqGk13qd9ki5nOhTjGhTb9FuOdrXrzno
VqbSf/dkdbp11FSt8nlvRFp1b6by0Evi0Glp1b9uOIUFsrKUDil3Fo5qnQ05LcinuK1WpUUayycWW04djb
OGeTaDnHO5txQNaVNPPRWjnuJlxThc2ull3ZplYrGu56Og/XuLSo1b2eUn7scgljtMTF/pW2lzojmoX0nLc
iRG6vNJaVF5Z+FnTC4Q6vp5Lsb2nOpdXXTuEYZbdmwJWjrWJ2kXtNvbVvLalms+4i4Sn0IT0RNuM+qS0t
+jtiWE/uVPRB3aP/WqTd3itpaMY5ye0rY3UutdM0s+C0uJ9q+gwyEZ125bclsDK9rWvxdu3c9sKL0e79L
JlJuvb/6sNFvemiPeXFejUUaUM4tb8iRbSqSoJ1klJhtT7mJnast60ra4ICKTUpZbSyuqzoUNOKTfBITJ5Xjb8
RaXsJVLVqCzfAOmczxmIVkpyvbmCkks9mzgWIWTt6CjQpuT4JFVQUr5hKrFx9S2rup0WlbtZ/kGLvEzP2
4W9xcuuoo1aT0X35bjjiVuOM4bNLYzaNS/ILLQS+LRHubit0ihWS/S89gS1o46IYpQs7VvL3VzIcMTqaf6or
RzJtSKubRqD95bCsjZV3PRcGlxYXJNo1FPpa12naya3NFNatq5ptcS4rlRtJR4RKa2WdxTS8SCZp/1VZYos7
dPgypLfFOzJd7ZUEIn+R5gADAAAAAAAAAAAAALfCuzy+YqC3wrs8vmLdfB5pwAK+gAAAAABh7mZMPcwKjCe21fR/ku
CnwttX0f5LgkMsPiAArUAAAAAAAAAAAAAAAAAAAAAAAAADzXtv2G2/5H+d0p5n2
37Dbf8AI/wB09i/4up/yP8ACPRHnfYv+Mq/8j/CPRAAAAAAAAAAAAAAA9wD3AU9l/L1fqXBT2X8
vV+pcEhjh8QAFbKPEO21Pp+CMScQ7bU+n4IxHysnIIACOQAAAAAAAFseZnpYnUispRTyIQDqtpR9J
08Um1+iCT4kOc5Tk5SbbZqAtr2t9yI0b+rSSTymvidJ4pNr9Mlr4kAFWMt4jW20pyqVNkbzbZd1pOf05R
3pbCiWxkupf1KIJ03GOTWQd4rxWJ26LFjqOTgmyNXualeWcnkuCOIJtnOS0xqZd7e6nb+7tXBkiWKTa/T
TSfEgAbWMIojUSmTxGpOj0bgtqyzzOVtcu3cnGKexOAG0nJaZ3t1uKzr1dNpLzlsNaVWVGanB5NGgDnI
O9rDWksttNNmscTqlPOEXmQQNtOf22nNzm5ZZNvMlOsRqU4KMkpfEhGQ4i9zuHe5upXLWIFLgbU
L6rQio7JRXcYkBtedt72sHik2nIBlhVasqs3ObzZoZBbJa327295Uobi5OPBneeKVGmowSfEgGQsZLRGolLn
iNSVJwcFtWWZjdqencqWWyO0iEi3u5W8XGMYvPvYWt92ibJGKVc5xp8NrlBtUqOpOUpb2ahze3K22A
AHAAAAAAAAAAAAAAAAAAAAAA

AAAAAW+FdnI8xUFvhXZ5fMWG+DzTgAV9AAAAADD3MyYe5gVGE9tq+j/ACXBT4T22r6P8lwSGWHx
AAVqAAAAAAAAAAAAAAAAAB5n237Dbf8AI/wemPM+2/Ybb/kf4A6exf8
AGVf+R/hHojzvsX/GVf8Akf4R6IAAAAAAAAAAAAAAAAAB5n237Dbf8AI/wemPM+2/Ybb/kf4A6exf8
AAAAAAAAAAAAAAHuAe4Cnsv5er9S4Key/l6v1LgkMcPiAArZr
4h22p9PwRSViHban0/BGI+Vk8pAARyAAAAAAAAdadtVqr0oQbQWImfpkB36nX8tmHaVoptweS
C8benEAByAAAFAtaXEACXUsJ06LqOSyXcRA6tWa/YDJgjkAAAAAAAAABQAAAAAAAAAC3wrs8
vmKgt8K7PL5iw3weacACvoAAAAAYe5mTD3MCownttX0f5Lgp8J7bV9H+S4JDLD4gAK1AAAAAAAA
AAAAAAAAAAAAAAAz7b9htv+R/g9K2ks28kjy/tNwp1LK3VOpGTVR7nn3Ad/Yv+M
q/8j/CPRHmfY6tSp4dUU6kYvpHsby7kemTzWaAAAAAAAAAAAAAB7gHuAp7L+Xq/UuCnsv5er
9S4JDHD4gAK2UelqtF8EUIyh22p9PwRiPlZPKQAECgAAAAAAAAAF1h/Y4IKXeHdkWHO/H8kaeJuE3
Ho9zy3mk8Tc4OPR71lJE6dm5Nycc/Uj3cLZUG6WjpZ9zDu/Oln/SNb28riTUGk1xOur6zqaOzJd50wn92
fodr+7nRkoU8k9+bDtK8OVkaeG1orNZSibTTyZb4fdTr6Uam+PeRLyknCiLpNMF8deMWq50LSrXWcvk
uLOrw6tDKSaeT3InXNRWltBbtIlfLEK3SrSacW9qDqaUpOp+0+87FLORSF3evOzm+KKQjn8jyAAR5wAA
AAAAAAAAAAAAAAFAAAAAAAAAAAAAAAAB7gHuAp7L+Xq/UuCnsv5er
9S4JDHD4gAK2UelqtF8EUIyh22p9PwRiPlZPKQAECgAAAAAAAAAF1h/Y4IKXeHdkWHO/H8kaeJuE3
Ho9zy3mk8Tc4OPR71lJE6dm5Nycc/Uj3cLZUG6WjpZ9zDu/Oln/SNb28riTUGk1xOur6zqaOzJd50wn92
fodr+7nRkoU8k9+bDtK8OVkaeG1orNZSibTTyZb4fdTr6Uam+PeRLyknCiLpNMF8deMWq50LSrXWcvk
uLOrw6tDKSaeT3InXNRWltBbtIlfLEK3SrSacW9qDqaUpOp+0+87FLORSF3evOzm+KKQjn8jyAAR5wAA
AAAAAAAAAAAAAAALfCuzy+YqC3wrs8vmLdfB5pwAK+gAAAAABh7mZMPcwKjCe21fR/ku
CnwnttX0f5LgkMsPiAArUAAAAAAAAAAAAAAAbugrq0qUG2IOOWaP
m1/a1rK6nQr55xexvvXE+nIRjuCRxWkpQahXh7sn3rgB4rDLktiF5G3otrPbj8EfSlEkqFvToxbahFRTe/YQ
MEweGFW2i2pVpe/NfgswAAAAAAAAAAAAAAE4B7gKey/l6v1Lgp7L+Xq/UuCQxw+IActlHiHban
0/BGJOIdtqfT8EUj5WTylkAEcgAAAAAAAAAF1h/Y4IKSKV7WowUINZL4CGuK8UtuXaph1eVSUlo7xs2
mjw64Sfu8xrK44x5GNZV3vcERxzcin264Uv8AVn6GuKdpXoR6NxOhJuGWb4oxWrTrz0ptZ/BBzN46fFM
wj36nov8AJrfS0MQjlhkyNQuKlu26bW3fmjWtWnWqac8s/gDnHT4/tc1qcbby2yjLftTINLDavSrTyUVvyZ
Ho3NWh7kttnA6TxCvOOwl6INJyY7am32sr7ZZzRRkid7Wns6OTwjllu5GeW8XncAAIxAAAAAAAA
AAAAFAAAAAAAAAAAAAAAAt8K7PL5ioLfCuzy+YsN8HmnAAr6AAAAAGHuZkw9zAqMJ7bV9H+S4KfCe21fR/k
uCQyw+IAc7ibp0JyW9IrSZ1G27IFb2jGnHxLmediKUpOTbbZjNk28nyf49Hpx8SGnHxI85mxmzs+T/Ho9O
PiQ04+JhnM2M2Nnyf49Hpx8SGnHxI85mxmzs+T/Ho9OPiQ04+JhnM2M2Nnyf49Hpx8S5m2ee481my
wwqrLpZU2845Zh3TPytrS1DaW9gpMQqyndSjm8o7EitcmThG1zpx8S5jTj4lZPO5jNk28/yf49Fpx8S5jTj4
lZPOZsZsbPlfx6PTj4lZGnHxLmecyzY2fK/j0enHxLmNOPiXM85mxmzs+V/Ho9OPiXM2TT3M81myxwqr
LpHTb2ZZh3T8jlbWloYcore0cbuo6dtOUd5RNtvNvaV1lzdOdPRdjhxldjhxi85t4jaTbL5P8ej6SPiQ6SPiR5
zbxG3iNnyf49Hpx8S5jpl+JhnNo28Rs+T/Ho+kj4kOkj4lZPObel2jZ8n+PSKUXuaMnm1JxeabTRf203Ut4T
e9oNcWXnOnUw5Jb2jL3FDdVZVLlibbex5IrrJk4QvNOPiQ04+JhncxmTbD5M+notOPiQ04+JhncxmNnyZ
9PRacfEhp8SPO5jMbPkz6ei04+JDTj4kedzGY2fJn09Fpx8SnsszeZ4VVKM4SeajuDvHn5W1pYgEHfas
oUoxi8tJ7St72412macfEhp8SPOZjMm3l+T/Ho9OPiQ04+JczzmYzY2fJn09Hpx8SGnHxI85mxmzs+TPp6
PTj4kNOPiR5zNjNjZ8mfT0enHxLmNOPiR5zNjNjZ8mfT0enHxI2TTWxnms2TsLqyVbo884tbg7pn5TrS3M
NpLa8jt4nVm7jQz/AEpbtc+Ebw2nHxLmNOPiXM85t4jaTbz/K/j0enHxIacfEjzm0bRs+V/Ho9OPiXMac
fEjzm0bRs+V/Ho9OPiQ04+JhnNo2jZ8r+PR6cfEjZNpczzW0nYZVkj/R5/pa3B1T8jlOtLcNpbwU+j1ZSuN
DP9KW4rbJfhG1t0kPEuY6SPiR5zNjNk28/yv49H0kfEuY6SPiR5zNjNjZ8r+PR9JHxldjhxi85mxmzs+V/Ho+
kj4kOkj4lZPOZsZsbPlfx6Ppl+JczKae483m+JowyrJv9DP8AS1uDqn5HkdaW5hyS3tGsgu6sqlxNt7E8kg1y
ZOEbXunHxIacfEjzmbGbG2HyZ9PR6cfEhp8SPOZsZsbPkz6ej04+JDTj4keczYzY2fJn09Hpx8SGnHxI85m
xmzs+TPp6PTj4lZnszzWbLTCqspRnCTzUcsg7x5+U60sQ9wD3FeT2X8vV+pcFPZfy9X6lwSGOHxAAVso8
Q7bu+u4lxJxDttT6fgikfKyeUsgAjkAAAAAAAAAAABkAAAAAAAAAAAAAAABQAAAAAAAAAAAAAA
AAAAAAAAAAAt8K7PL5ioLfCuzy+YsN8HmnAAr6AAAAAGHuZkw9zAqMJ7bV9H+S4KfCe21fR/kuCQyw
+lcbzstT0OxxvOyz9Cu7+MqAAHL5UgBILPiBgAAAAk+AAAACbhxaX8pCJuFdpfylhpi84XBQ3vbKnqXxQ3
vbKnqJer8nxhwAH0I8IAPoAAAAACbhfaX8v/ghE3C+1P5RDTF5wn3/AGOf0KMvL/sc/oUYa/k+QAA8w

DOT4GAaABk+4ADOTW9GAaC9seyU/Qoi9seyU/QsPT+N5SkPcedrvfT+ZnonuPO1v3p/MxLv8n6hoACP
AAAAAAAHOYAFjhHv1fRFcWOEe/V9El4fOVoVuL+5D1LircX9yHqdPbm8JVgAOXzQAAAAAAAAACXh
va16EQI4b2tegaYvKF0UmI9sl6luykxHtkvRFev8jxRQAR4AD6MfQAAPowAH0YAErDe2RlpKw3tkQ7x+U
LspMR7XL0RdlJiPa5Fev8nxRQAR4QGcuBqKABbQgBkABKw3ta9CKSsN7WvRhpj84Xb3Hna/78/U9Ezztb
9+fqWXp/J+oaAAjxAH0AAAAAAALHCPeq+iK4scI96r6IQ1wecLQPcA9x0+kp7L+Xq/UuCnsv5er9S4JDHD
4gAK2UeIdtqfT8EYk4h22p9PwRiPZPKQAEcgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAW+FdnI8xFvhXZ5fMWG+DzTgAV9AAAAADD3MyYe5gVGE9tq+j/JcFPhPbav
o/yXBIZYfEON52Wfodjjedln6Fd38ZUAOXypTrCzjWXSVpd7lxJs6lrQejRi/gjnhtWMrdQXvR3nG7satSu
509qZXsrHGkTSNpTo29zTzjGOXFllqbqi6FZwb2b0Wtjbzt6TU2s288I3FfiNVVLjOOTUVkEyxE0iZjuk2VID
o41Kq0m9uR3nXtKctB6Ka7IE3takattHR4ZP4FfUw6t0kssmm9+YdT/mscl2m1bSjcQ0opJvc0U9SDhUIF
708i7tqbt7dRm9y2IPc1FVrzlHc2HGesaif25E3Cu0v5SETcK7S/IEMcXnC4KG97ZU9S+KG97ZU9Q9X5PjD
nTwdWCe7SRdulQjHOVOCXFoo4SOZxllnk8yZXxBVqLp9G1n35hjtWsTtN/6ThS5I6KhRazVOHlprWk6te
MVu3st7mqqFvKXwyQb0vFomZhFt4U53tzaMXFblkSpQtoPKUaafxSKu1uerzlJx0s/ia3lwrmcZKOWSyD
KMta13+1tGNrOWjGNNvgkjS7pUYW85KEU8t+RHwuuhkpVXvexGMVq7Y0l6sNJtHT5TCuJuF9qfy/wDgh
E3C+1P5f/BHIxecJ9/2Sf0KMVL/ALP6FGGv5Hm2hB1JqMd7La3sKVKOdRKUu/Mi4XTUqspvuWwlXyq1
FGIR3veFxUiK8phv0lqv05w9ChiPQKMeiUdj98TR4ZWS20lfqRqtOdKWjUjkype9tamG9rbyuKmitiW1st
Y29vbwzcY+sjXDqahbKSW2W04XNGvdV2orKEd2Yd1pwrvW5SelKmzOD9UVd50auJkkorgbw+vBZp
J+hGaabT3kZZLWmNTGmC9seyU/Qoi9seyU/QsOvxvKUh7jztb96fzM9E9x52t+9P5mJd/k/UNC1w6jCV
BuclyefeiqLnDllaR+OZGWCN2byVpFtNU018Ec6rtOinoqnnk8til9ewrVK05rlJvZtlla2q0ffjs4laWvaP8A5
ZtYqdzCLWaz2ouJ07ems5whFfFlqsPWD3H4Fle0JXFJRhvTzBhj/Ezo0rPhS5lrb/o+n/0tHRy/7TM8Prx2qK
a+DI0k4vJrJojJa0xqY0wWOEe/V9EVxY4R79X0Qj7cYfOFoVuL+7D1LircX92Hqds9mbwlWAA5fOC0sbO
Kh0lWKbe5Mj2Fr0s+kn7kfuTL646GGhD35bPQr04qREc7Id/UpynoUoxSW9pEM7O1rKLk4PLfmcimB7
mdzDBaWNnFQ06sU29yZww+26WppzX6Vu+JKv7noafR0/fi9itsVlrHOyFfVKbnoUoRSjvaW8inaVrWjFy
cHkjiRjfzcuzQl4b2tehEjeG9rXoFxeULopMR7ZL0RdlJiPbjieOnr/I8UUtbR23QQ01T0stuaRVB7DI46X4zteL
qsnklTbfwRtKjRjFt04JehGw+0UIKrNZye1fa0xKrV9xRah3viV7OWqcphwtS623PR0M3v3FhnacKX2KT
M7WtCVxV0c8l3sPPTJMdohcQp281nGEGvgiDiapQUlwjGMs9uSJ0l1e3ypQbyWxIpKspym3U974hrmtq
utd2hKw3tcSKSsN7XEjzY/KF2UmlkrkXZSYh2uRZev8nxRiwsrWjKn0tSSI8O5FeFjpNjt7yPJS0VncwtutW
cXorR/9pmra0binpU0k8tjSKcucPg6VrnPv2leil+pOphXW9CNSu6c5aORYN2lstFqOfpmyqrT0q8px2Zs0b
bebcb4kZRkinbS5hK1us4xUW13ZFfe23V6i0fdluNsNhKV0pLdHedcWmnKEvvW0NLavj5TCvJWG9rXoyI
S8N7WvRhj84XTPPVv35+p6Fnnq378/UsvT+T9Q0ABHihdU+qtRWVNy9ELujTjbTcacU8t6RVW3aYepcX
nZZ+hXtpaL1mdllh0HQf6qhpZ95NVGhKOapwafwKAvtTstP5URMFuXbSlrJktNjZLM5nSv8Avz9TmHkt9
hZYR71X0RWllhHvVfRCGmDzhZh7gHuOn0IPZfy9X6lwU9l/L1fqXBIY4fEABWyxjDttT6fgikrEO21Pp+CM
R8vJ5SAAjgAAAC3wrs8vm
Kgt8K7PL5iw3weacACvoAAAAAAYe5mTD3MCownttX0f5Lgp8J7bV9H+S4JDLD4hxvOyz9Dscbzss/Qru/j
KgABy+UubC3jSpKeX65LNs4VsScadjTjsXeyRZV41KMY5/qismjjWwzTqOUJ6Ob3Mr3Ty4R03a0u+sqUZL
KS3ohX1tGFWOhsU+4m2trG1Tbecn3kO/ulzqxUnuhtzCX815/afRpQtaGxbIm3xIUsUnpfpgtH4snUqsLij
nF7GsmuBDlhTcv01P0+gdX56jppVCrG7oZuOzc0V5RVG4cVue1FtQowtaOWezvbKm8rKvcOUfdWxC
Webwj9uBNwrtL+UhE3Cu0v5Qwxeclgob3tlt1L4ob3tlt1D1fk+MOBkwdKVN1asYLvZHiiNzpY4ZS0aTq
NbZDFK2lUVJPZHa/UsZONC3b3KK2FFOTnNzfe8yvVknhSKQ1Mxi5zUY728jBowyjpVXUa2R3eoaleV
ohYwSoUEtyiijrVHVqym+9lnidbQo6C3y3ISRnt34x+gm4X2p/l/4IRNwvtT+x/wGeLzhPv8Ask/oUzeX/ZJ
/Qow0/l8llhLX60Sry46vTTyzb2lqrSv0FZsfuvYy1r0oXdfzs+KaK1xTvHqPtDoYjOVWMaiTTfcSr+IGrbSfet
qZxoYboVFOc88u5G+IXEYUXTT/AFS2egWOUUnm62LztIZdyyOF3fSo1dCmls3tnLDbqMP9KbyTexki6sV
cVNOMtF9/xBym2OOH2xZXkq8nCa2rc0cMvpKmozWxveSbSzVtnJy0pPvleI141ZqEXmo95HN5mMer/
aExTj2Sn6FFkXtj2Sn6Fhx+N5SkPcedrvfT+ZnonuPO1v3p/MxLv8n6hoS6V/OjSUFFZlipZtLiW6sKFSjFLfI7
yZGGKLTM8UaOKTzWdNZFinC4oJ5bJlhxwuCltm2iVUnTtqO1pJLJlr1U5xE81RCo7W5k4pNptbTvrOp4U
ZsaVO4q1J1cnnuTO08Mpyf6ZSiGNa31usulpe9Yk4yjkzhitKKUaiWTexkm2tIW7ck22+9kLERhVJKnFppb

w0v2x6v9oJY4R79X0RXFjhHv1fREj7efD5wtCtxf3YepZFbi/uw9TqXszeEqw3oU+lrRhuzZobUqjpVFOO9
HL58ffdf06apU1CC2IhTsKtSrpzqJvPMj6yrfDkZjiNZxitm18CvXOTHbUSs509Ki4Z71lmVksPICpCMpp6TL
OpNxoSmt6WZUzvqspKTazju2B1lmnba3hBU6ajBZJLYQZ4fUqVXOdRNt5kdYIWY3rkbRxGs5pbMm+Ac
zKx21ErSpDTpOOeWayKa6tJWytIJPMuKsnGjKS3pFNXuZ3CSnlsfcDPx1/XAI4b2tehEJeG9rXoR58XIC6KT
Ee2S9EXZSYj2yXoivX+R4ooBa0bKjO2ink34kyPHSk3+nCniU6clx0E0llvLCjUjdUdLLY96ZF1XDP35ZEqKp2
tHLNKKK9mOLx5/Smuas03EoLcnsM21w7ablFztrIkUFC7vZTnlo9yfeSKmGUPOcfQMlx2meVS1xDpq
qhOOTe5mMTpRdHpMlmjpB2EKE9PNyfxOOJV4un0UWm3v+AbTvpzzVhKw3tcSKSsN7XEjyY/KF2UmI
drkXZSYh2uRZev8nxRjBkn4faRnlVnty3lJyUpN51BZWWl/qVlsW1Ji+vE06VJ/p72TbmlVqw0Kc1Bd5C1V
LzFyK9NqWrXjSFetp2tradxPKOxd7Ois2rtUHLuzzRbU6KpU9Cns+LDPHhmZ7/pHqVKVjQ0Y5aXcuJU1Kk
qs3KTzbLGphtWrJynVTb+BHuLCVc5uafwluSt5/XZDJeG9rXoyIS8N7WvRhIj84XTPPVv35+p6E89W/fn6
ll6fyfqGgAl8Lra9ph6lxedln6FPa9ph6lxedln6FezB4Soi9tOy0/IKlvbPstP5SJ+N9yp7j9+fqcjrcfvz9TkHmt9
hZYR71X0RWlhhVvfRCGmDzhZh7gHuOn0IPZfy9X6lwU9l/L1fqXB1Y4fEABWyxHttT6fgjEzFIZXOfiRDI+X
kjVpAARwAAAB4T2eXzFQ
XWGw0bRPi2yw9H4/mlgAr3gAAAAAYe5mTD3MCownttX0f5Lgp8J7bV9H+S4JDLD4hxvOyz9Dscbzss/
Qru/jKgABy+UzGTi808md1e3CWXSPL4kcB1Fpj6dalzWqrKc21wOQASZmfttCpODzhJr0OyvrhLLpGRwFi
0x9S6VK9Wr782zmAEmZn7CbhXaX8pCJuFdpfyld4vOFwUN72yp6l8UN72yp6h6vyfGHA2hUITlpQbT4o
1BHh+nWdzVqR0Z1JNcGcgAszM/YdKdxVpx0YTcVwRzASJmPptUqTqy0qknJ/E1ABM7Cbfan8v/ghE3C
+1P5f/AaYvOE+/wCyT+hRI5f9kn9CjDT8jyDpTr1KX7cmjmAwiZj6SJX1w010jWZwbb2yeb4mAFm0z9h2
hd1oLKM3kcQEiZj6dp3dapHRIN5HEAEzM/YXtj2Sn6FEXtj2Sn6Fh6PxvKUh7jztb96fzM9E9x52t+9P5mJa
fk/UNDpCvUp+5No5gjxxMx9JHXrjL3lUqzqPOcm/U0AJtM/cspuLzi2mdle3EVkqj+pwAltMfUu1S7r1I5S
qPI4gAmZn7Cwxj36voiuLHCPfq+iENMP/SFoVuL+5D1LircX9yHqdPbm8JVGaoXzQLY80AB2d1XIbxR5P
YcQAszM/YE2nmgAjtK6ryi4upJp9xxACzMz9hLw3ta9CIS8N7WvQO8XIC6KTEe2S9EXZSYi2yXoivX+R4op
vCrOm84Sa+poCPBEzH0kdeuPGcqlarVf65tmgDqb2ntMibTzTO8buvBZKo/qcAVImY+neV5XksnNnBvN
5gEjtM/YSsN7XEikrDe1xDrH5QuykxDtc1KTEe1yLL1/k+KKdYXFwmtGE3FcEcgr4tzH079cuPNkouXhm
yOADrb26dPV6TpNN6e7M365ceb14Ajyn279cuPNkaTuK1SOjOo2uBzAJtb2ErDe1r0ZFJWG9rXoVcfnC
6Z56t+/P1PQs89W/fn6iXq/J+oaAAjwsxk4yTTyaOkrmtoLjkpJp9xyAdRMx9B1jdV4RUY1JJLuOQBEzH0z
JuUm282zAAQLLCPeq+iK0ssl96r6lQ1wecLMPcA9x0+kp7L+xq/UuCnsv5er9S4JDHD4gAK2QcUpadFTW
+P4Kg9JKKIFxa2NZFDd0HQrOL3f9r4keL8infk4gAjygAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAK2pwdSpGK3tnoacdCnGPBFfhls1/rTXylkWht/HpqNyAAr0gAAAAAYe5mT
D3MCownttX0f5Lgp8J7bV9H+S4JDLD4hxvOyz9DsYklJNNZplaWjcaebBbzwnKTcZOOfcaaqh5j5E08E4L
qsFpqqPmPkNVQ8x8iaOhdVgtNVQ8x8hqqHmPkNHQuqwWmqoeY+Q1VDzHyGjoXVYLTUfMflaqj5j5
FOhdVk3Cu0v5TvqqHmPkSra1p26ehve9sO8eG0WiZdyhvtl5u+LL4jXNnTuHm9kuKdfNSb17KMFpqqHj
Y1VDxsaeXoXVYLTVMGPxqmHjY0nx7qsFpqqmHjY1TT8bGj491WC01TDxsaph42NHx7qsm4V2p/KSFhU
PGyVb20LeOUD/ew0x4bRaJlpf9kn9CjPRyipRcWs095CnhIOUtkmkHebFa07hUgtNVQ8bGqoeNhj0LqsF
rqhqHjY1VDxsHQuqgWuqoeNjVUPGwdC6qBa6qh42NVQ8bGjoXVRe2PZKfocYYXSjLOUm1wJsUopJbkIb
4cvqTuWXuPO1v35/Mz0REuLCIXm57Yye9oOs2Obx2UoLXVUPGxqqn42NPL8e6qBa6qp+NjVVpxsaPj3
VQLXVVPxsaqp+NjR8e6qBa6qp+NjVVpxsaX491UWOEe9U9EdNVU/GyZQowoQ0YLLiGmLDatty6Fbi/u
w9SyOdajCtDRms0V6cleVdPPAtdVQ8bGqoeNk08XQuqgWuqoeNjVUPGxo+PdVAtdVQ8bGqoeNjR8e6
qBa6qh42NVQ8bGj491UC11VDxsaqh42NHQuqiXhva16ErVUPGyTb2tO3zcdre9sO8eG0WiZdykxLtkvR
F2R7i0p3G2WyS3NFejLsb11CiBa6rj5jGq4eY+RNPJ8e6qBa6qj5j5DVUFmfIHQuqgWuqo+Y+Q1VHzHyB
0LqoFrqqPmPkNVR8x8gdC6qJWg9rjkS9VQ8x8iRbWtO3T0dre9sO6YLRaJlIKTEe1y9C7I9zaU7jy2Nd6K
3zUm9dQogWuq4eN8hquHjZHI6F1UC11XDxsarh42DoXVQLXvcPGxquHjY0dC6qBa6rh42NVw8bGjoX
VRLw1Z3a9GStVw8bjNta07fPR2t94dY8NotEy7dx5+vsrz9T0JEuLcnWnp+7J78g3zUm8dIKC11VDxvkNV
Q8b5DTzdC6qBa6qh43yGqoeN8ho6F1UC11VDxvkNVQ8b5DR0LqoFrqqHjflaqh43yGj491UWWEe9V+
hvqqHjfII0KEKENGcyGmmLDatty6h7gHuK9insv5er9S4Key/l6v1LgkMcPiAArYONzbxulaMlt7nwOwCTE
TGpefr287eeU08u59zOR6OpTjUi4zSa+jXXGGd9F/Rk08WTBMD6q0HadpXp+9TeXFHPQI4XyDCazH6ag

answer

```
#include<iostream>
```

```
using namespace std;
```

```

int main()
{
    int n,i;
    cin>>n;
    int arr[n];
    for(i=0;i<n;i++)
        cin>>arr[i];
    for(i=0;i<n/2;i++)
    {
        int temp;
        temp=arr[i];
        arr[i]=arr[n-1-i];
        arr[n-1-i]=temp;
    }
    for(int i=0;i<n;i++)
        cout<<arr[i]<<" ";
    return 0;
}

```

question

<p>CTS Interview Question</p><p>Question description</p><p>Issac is a Language teacher at a high school in Madurai. </p><p>Sabari is a student, he is studying programming while during Language class. </p><p>Issac tells Sabari to leave the other subject and learn Tamil. </p><p>Sabari asked permission for 10 minutes, Finally, Sabri got permission to solve the program.</p><p>The computer teacher has given homework on the topic of Arrays and where he need to do Array Rotation as per the input of question. . But Sabari is not good at C Programming. </p><p>Can you help him to solve the programming problem?</p><p>Constraints</p><p>0< n < 100</p><p>0 <= arr < 1000</p><p>Input Format
First line predicts the total number of elements present int the array.
 -Second line contains the elements of array
 -Third line contains the number of time rotation need to be done
 -Fourth line contains the character 'L' or 'R' that defines what type of rotation needs to be done.</p><p>Output Format
 -Output contains only line that is only the resultant output</p>

answer

```
#include <iostream>
using namespace std;

int rotLeft(int arr[],int n,int d){
    for(int i=d;i<n;i++)
        cout<<arr[i]<<" ";
    for(int i=0;i<d;i++)
        cout<<arr[i]<<" ";
    return 1;
}

int rotRight(int arr[],int n,int d){
    for(int i=n-d;i<n;i++)
        cout<<arr[i]<<" ";
    for(int i=0;i<n-d;i++)
        cout<<arr[i]<<" ";
    return 1;
}

int main()
{
    int n,d;
    char c;
    cin>>n;
    int arr[n];
    for(int i=0;i<n;i++)
        cin>>arr[i];
    cin>>d;
    int z;
    z=d%n;
    cin>>c;
    if(c=='L')
```

```

    rotLeft(arr,n,z);
else
    rotRight(arr,n,z);
return 0;
}

```

question

<p>Problem Description:
How many Y's did a Roman Centurion make a day in cold hard Lira? About a C's worth! Turns out, Martians gave Rome the idea for their number system. Use the conversion charts below to help translate some Martian numbers!

Note, that unlike the Roman Numerals, Martian Numerals reuse symbols to mean different values. B can either mean '1' or '100' depending on where it appears in the number sequence.

Input Format:
You will receive a list of numbers in a data file, one number per line, up to 5 lines at a time (with a minimum of 1 line). No number will exceed 1000, or be less than 1.

Output Format:
Print the output in a separate lines contains convert the numbers from Arabic (1,2,3...10...500...1000) to Martian (B,BB,BBB...Z...G...R)
numerals.</p>

answer

```

#include <bits/stdc++.h>
using namespace std;
void print(int number)
{
    int num[] = {1,4,5,9,10,40,50,90,100,400,500,900,1000};
    string sym[] = {"B","BW","W","BK","Z","ZP","P","ZB","B","BG","G","BR","R"};
    int i=12;
    while(number>0)
    {
        int div = number/num[i];
        number = number%num[i];
        while(div--)
        {
            cout<<sym[i];
        }
    }
}

```

```

    i--;
}

}

int main()
{
    int number,n2,n3,n4,n5;

    cin>>number>>n2>>n3>>n4>>n5;

    print(number);

    cout<<endl;

    print(n2);

    cout<<endl;

    print(n3);

    cout<<endl;

    print(n4);

    cout<<endl;

    print(n5);

    return 0;
}

cout<<"char buf[] buf[i++]='R'; while(n>=10)";

}

```

question

<p>Problem Description:
Public school have arranged an Annual Day Function.</p><p>Volunteers have decorated a floor on various places of the school using Rose and Tulip flowers. </p><p>But one of the coordinators requested the volunteers to rearrange the decoration like a triangular size.</p><p>Coordinator also told them that tulips flowers need to be positioned at the middle of the roses</p><p>School has 20 buildings and as per Principal order the numbers of rows in the decoration should also match the building number.
The Principal of the school is interested in seeing the final decoration but he is quite busy with the other works.</p><p>So he likes to see how the final decoration have come through online mode if he gives the building number.</p><p>So can you display him the final decoration layout?
Note:
Roses are represented by 1.</p><p>Tulips are represented by 0.

Constraints:
1 ≤ rows ≤ 20</p>
Input Format:
Only line of input has single

integer representing the building number.</p><p>
Output Format:
Print the final layout of the decoration.</p><p>Refer sample testcases for format specification.</p>

answer

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int n;cin>>n;cout<<"1 \n";
    for (int i = 0; i < n-2; i++) {
        cout<<"1 ";
        for (int j = 0; j < i; j++) {
            cout<<"0 ";
        }
        cout<<"1 \n";
    }
    for (int i = 0; i < n; i++) {
        cout<<"1 ";
    }
    return 0;
    cout<<"for(i=1;i<=rows;i++)";
}
```

question

<p>Problem Description:

saravanan with his friends going to the theatre for a movie. </p><p>The seating arrangement is triangular in size. </p><p>Theatre staffs insisted the audience to sit in odd row if the seat number is odd and in even row if the seat number is even.</p><p>But the instruction is very confusing for saravanan and his friends.</p><p>So help them with the seating layout so that they can sit in correct seats.

Constraints: </p><p>4 ≤ N ≤ 20

Input Format:
Only line of input has single integer value representing the number of rows in the theatre.

Output Format:

Print the layout based on the number of rows specified in input.</p><p>Refer sample testcases for format specification.</p>

answer

```
#include <stdio.h>
```

```
int main()
{
    int i,j,k,N;
    scanf("%d",&N);
    for(i=1;i<=N;i++)
    {
        if(i%2==0)
        {
            k=2;
        }
        else
        {
            k=1;
        }
        for(j=1; j<=i; j++,k+=2)
        {
            printf("%d ", k);
        }
        printf("\n");
    }
    return 0;
}
```

question

<p>Problem Description:
Simon work with Greek squares and matrix traces.

The trace of a square matrix is the sum of the values on the main diagonal (which runs from the upper left to the lower right).

An B-by-B square matrix is a Greek square if each cell contains one of B

different values, and no value is repeated within a row or a column. In this problem, we will deal only with "beautiful Greek squares" in which the B values are the integers between 1 and B.

Given a matrix that contains only integers between 1 and B, we want to compute its trace and check whether it is a beautiful Greek square. To give some additional information, instead of simply telling us whether the matrix is a beautiful Greek square or not, show the number of rows and the number of columns that contain repeated values.

Constraints:

$1 \leq T \leq 100$

$2 \leq B \leq 100$

$1 \leq A_{i,j} \leq B$, for all i, j .

Input Format:

The first line of the input gives the number of test cases, T . T test cases follow. Each starts with a line containing a single integer B : the size of the matrix to explore. Then, B lines follow. The i -th of these lines contains B integers $A_{i,1}, A_{i,2}, \dots, A_{i,B}$. $A_{i,j}$ is the integer in the i -th row and j -th column of the matrix.

Output Format:

Print the output in a single line contains the number of rows and the number of columns that contain repeated values.

Explanation:

Input&nbsnbsp;	Output&nbsnbsp;
----------------	-----------------

Case #1: 4 0 0

Case #2: 9 4 4

Case #3: 8 0 2

In Sample Case #1, the input is a natural Latin square, which means no row or column has repeated elements. All four values in the main diagonal are 1, and so the trace (their sum) is 4.

In Sample Case #2, all rows and columns have

repeated elements. Notice that each row or column with repeated elements is counted only once regardless of the number of elements that are repeated or how often they are repeated within the row or column. In addition, notice that some integers in the range 1 through **N** may be absent from the input.

In Sample Case #3, the leftmost and rightmost columns have repeated elements.

answer

```
#include <bits/stdc++.h>
using namespace std;
int t,i,j,tes,n,x,y,sum;
int a[1007][1007];
map<int,bool> udah;
void solve(){}
int main() {
    solve();
    scanf("%d",&t);
    for (tes=1 ; tes<=t ; tes++) {
        scanf("%d",&n);
        for (i=1 ; i<=n ; i++) {
            for (j=1 ; j<=n ; j++) {
                scanf("%d",&a[i][j]);
            }
        }
        sum = 0;
        x = 0;
        y = 0;
        for (i=1 ; i<=n ; i++) {
            udah.clear();
            for (j=1 ; j<=n ; j++) {
                if (udah[a[i][j]]) x++, j = n;
                udah[a[i][j]] = true;
            }
        }
    }
}
```

```

    }

    for (j=1 ; j<=n ; j++) {

        udah.clear();

        for (i=1 ; i<=n ; i++) {

            if (udah[a[i][j]]) y++, i = n;

            udah[a[i][j]] = true;
        }
    }

    for (i=1 ; i<=n ; i++) sum += a[i][i];

    printf("%d %d %d\n",sum,x,y);

}

return 0;

cout<<"for(i=0;i<n;i++); int g[105][105];";

}

```

question

<p>Problem Description:
For some reason, your school's football team has chosen to spell out the numbers on their jerseys instead of using the usual digits. Being great fans, you're going to be ready to cheer for your favorite players by bringing letter cards so you can spell out their number. Each fan has different favorites, so they each need to bring different sets of letters.

The English spellings for the numbers 0 to 12 are:
ZERO ONE TWO THREE FOUR FIVE SIX
SEVEN EIGHT NINE TEN ELEVEN TWELVE

Input Format:
Read a set of integers from 0 to 12, separated by spaces, representing one fan's favorite players. The last integer will be 999, marking the end of the line.

Output Format:
Print the same numbers, then a period and a space. Then, in alphabetical order, print all the letters the fan needs to be able to spell any one of the jersey numbers provided</p>

answer

```

#include <bits/stdc++.h>

using namespace std;

int main()

{
    int curr;

```

```

multiset<char> mp;

string names[] =
{ "", "ONE", "TWO", "THREE", "FOUR", "FIVE", "SIX", "SEVEN", "EIGHT", "NINE", "TEN", "ELEVEN", "TWELVE"
};

while(cin>>curr){

    if(curr==999){

        cout<<"0999"<<'.'<<' ';
        break;
    }

    cout<<curr<<' ';

    if(curr>12)continue;

    string now = names[curr];

    for(auto ch:now){

        mp.insert(ch);
    }
}

for (auto ch : mp) {

    cout<<ch<<' ';
}

return 0;
printf("char nums[13][256]for(n=0;n<26;n++)");
}

```

question

<p>Problem Description:</p><p>Umesh has n mixtures in front of him, arranged in a row. Each mixture has one of 100 different colors (colors have numbers from 0 to 99).</p><p>He wants to mix all these mixtures together. At each step, he is going to take two mixtures that stand next to each other and mix them together, and put the resulting mixture in their place.</p><p>Functional Description:</p><p>When mixing two mixtures of colors a and b, the resulting mixture will have the color $(a+b) \bmod 100$.</p><p>Also, there will be some smoke in the process. The amount of smoke generated when mixing two mixtures of colors a and b is $a*b$.</p><p>Find out what is the minimum amount of smoke that Umesh can get when mixing all the mixtures

together.</p><p>Constraints:</p><p>1 <= n <= 100</p><p>Input Format:</p><p>There will be a number of test cases in the input.</p><p>The first line of each test case will contain n, the number of mixtures, </p><p>The second line will contain n integers representing the initial colors of the mixtures.</p><p>Output Format:</p><p>For each test case, output the minimum amount of smoke.</p>

answer

```
#include<stdio.h>

typedef long long unsigned LLU;

LLU min_smoke[100][100];
int color[100][100];

LLU smoke(int n){
    int i,j,l;
    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            min_smoke[i][j]=10000000000000000000;
        }
    }

    for(i=0;i<n;i++){
        min_smoke[i][i] = 0;
    }

    for(l=2;l<=n;l++){
        int e = n-l;
        for(i=0;i<=e;i++){
            int k = i+l-1;
            for(j=i;j<k;j++){
                LLU sm = min_smoke[i][j] + min_smoke[j+1][k] + color[i][j]*color[j+1][k];
                if(sm < min_smoke[i][k])
                    min_smoke[i][k] = sm;
            }
        }
    }
}
```

```

        int cl = (color[i][j]+color[j+1][k])%100;
        if(sm<min_smoke[i][k]){
            min_smoke[i][k] = sm;
            color[i][k] = cl;
        }
    }

}

return min_smoke[0][n-1];
}

```

```

int main(void){
    int n;
    while(scanf("%d",&n)!=EOF){
        int i;
        for(i=0;i<n;i++){
            scanf("%d",&(color[i][i]));
        }
        printf("%llu\n",smoke(n));
    }
    return 0;
    printf("scount[100][100]colours[100]");
}

```

question

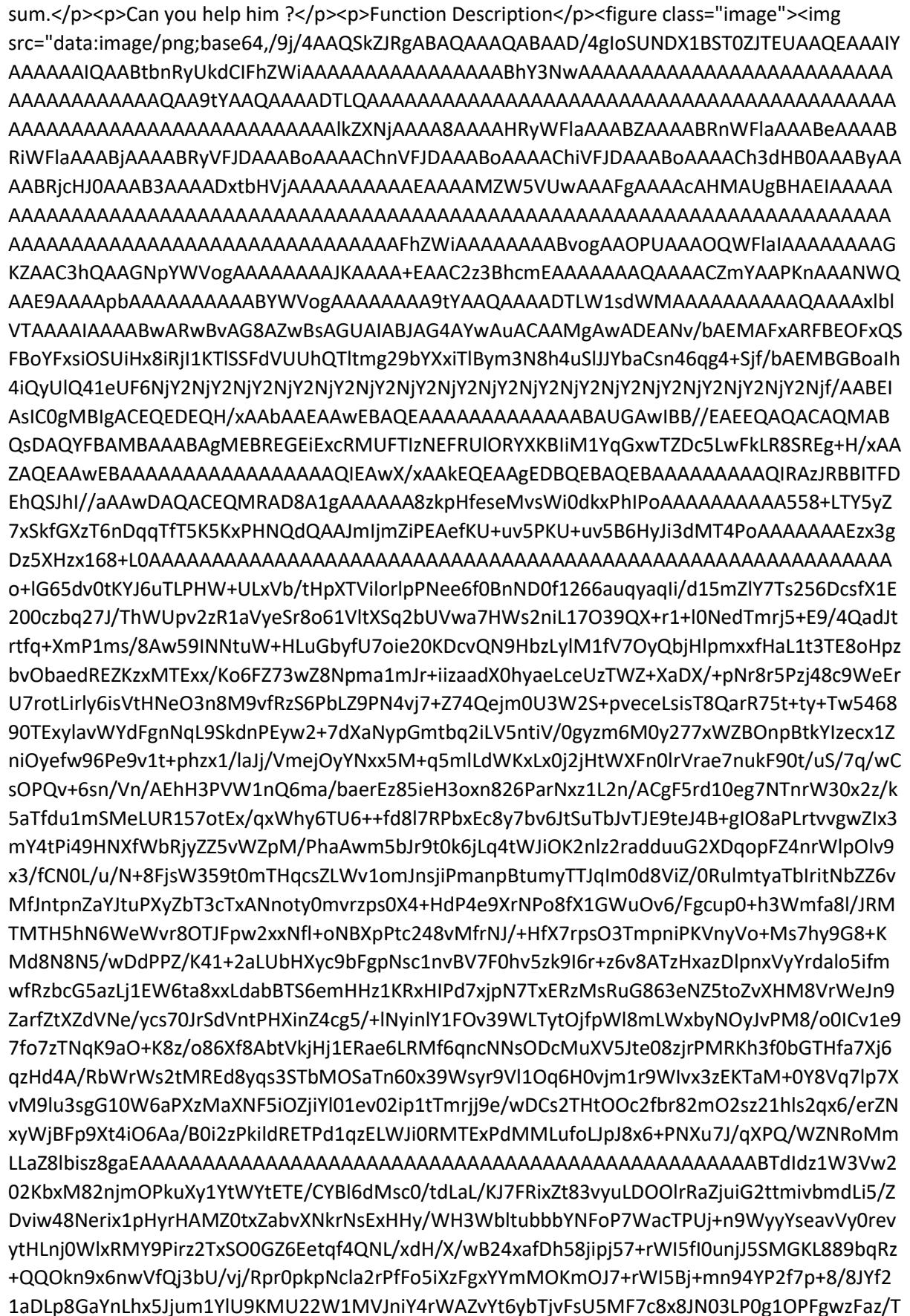
<p>Question description</p><p>Malar is a First year student in reputed institution. </p><p>Although he scored well in many subjects, he did not an expert in Algorithms.</p><p>But malar's computer examination is scheduled for next week. </p><p>As

per the blueprint, many questions would come from the Arrays topic.

He collected previous year's questions. one of the repeated questions is you need to find the pairs in Array with given sum.

Can you help him ?

Function Description



WP9ZXfRPSxba8tNTgif7XuyV/SPmvsOlwYPY4cdP1rWIBn960nmPROun+Nerz48xy+dFMMajYtRht3Xv
Mf5Q0mTFjy16uWlb1+Vo5h8xYcWGvVxY6Y47+KxwDAUx5di3C0avS1zY+7+qOyY+cJ2TfdFkmlDntVLW
mY5iatjlw4s1eMuOl4+Vo5c8ei0uK3Wx6fFW3zikcg+6SmOmmp5LFGGto56nHHDsAKTp/CDv31Ruhf9
35v3tFkx48terlpW9flaOYfMWHFhiYxY6Y4n4VrEArOkW3X3HbprijnLjnrVj5/oy+27jh26ltPr9BXJMT2TNf
6ob5xy6TT555y4Md5+dqxMgzO37lpNfuNcNNqp5G3/uivbHimb1vWPRauNLI0flsHVjmZjs/heYtPhwRx
hxUx/trEPuTDizRxlx0vH+KsSDBbnrNr1eHq6HQ2xZpnvilj/Rodh0Wor0ey4c0TW2XnqxPwiYW9NBpMd
utTTYon5xSEgH59tG4W2PcMs6jDaeYmsx8YfbddXcNFTUUrnYTz2T8HTLo9Nmt1suDHe3ztWJI0x46Yq
9XHStKx8KxxAPmbFXNhvjvHNbRxLCZ9Nreju5eVwxM0if6bcdlo+Ut883pTJXq3rW1Z+Fo5gGU/8AWc+S
7NJHIP3diXsG47l7Z/OMf8AZTEzW89nE/KFxG2aGLdaNLh5/ZCTWlaVitaxWI7oiAfnunti0G9XtumC2Sv
WtzExzzPze991ml1lcc6HSeSxU779XjmW6zaTT555zYcd5+dqxJ5ppupFPN8XUjur1I4BR6LWX2/orizVxTk
t3RX+Vbl3nZ9RSZ1G2zGWe+axHf4tlXFjrj6laVin0xHZ+HCdv0drdadLh5/ZAMh0W0ubJuts2O1qYlrMTz8
p+DluGizbNu055wRmwzabRzHMTE/Bu6Y6Y69XHStK/KscQXx0yV6t61tHytHMAxuTf9BbFxh2unlZ+ExH
DtBr5O+hpmx6aNPOSObV447Xaug0lIldaumwxPz6kJAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAABMRPfHIARER3RwAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA
ADhrM04NPN69/wETOlzLuKCdXnmefKWPOs/3blyzezXhfig86z/AHbHnWf7tj7NeF+KDzrP92x51n+7Yy
ezXhfig86z/dsedZ/u2Mns14X4oPOs/3bPWPW56WievMx8pMp9mvC9HnHbr0rb5w9T3JaMgpdRrcs1
oraa1ielihLzrP92yMs89RWJxhfig86z/dsedZ/u2Mo9mvC/FB51n+7Y86z/dsZPZrwvxQedZ/u2POs/3bGT
2a8L8UHnWf7tjzrP8AdsZPZrwvxD2/U2z45i/bavx+aYloraLRmARNw1FsGKOOp2WtPHPyVfnWf7tkZcr60
UnC/FB51n+7Y86z/AHbGVPZjhfig86z/AHbHnWf7tj7McL8UHnWf7tjzrP92xk9mOF+KDzrP92x51n+7Y
yezHC/FJg1uamWvWvNqzPbErqj5iJS7aepF/j6Pl7dWk2+UcqPLrc97zMZJiPhElRqakU+r0Z/zrP92z751n
+7Yy5ezXhfig86z/dsedZ/u2Mns14X4oPOs/3bHnWf7tj7NeF+KDzrP8AdsedZ/u2MnsxwvxQRqs/3bLfR
Z5z4ltbvjskdNPWi84SAVm46vJTL5PHbq8R28JXveKRmVmcs/51n+7Y86z/AHlly4ezHDQcnLP+dZ/uWP
Os/wByxk9mvDQcnLP+dZ/u2POs/wB2xk9m0Gg5OWf86z/dsedZ/u2Mnsxw0Az/AJ1n+7ZN27VZL5fJ5L
dbnu5MrV162nCzeb5K469a9oiP1elVu1p8pSvPZxzwI01L9lcpvnuD7kHnuD7kKIRlI9m3C989wfcg89wfc
hRBk9m3C989wfcg89wfchRBk9m3C989wfcg89wfchRBk9m3C989wfcg0x58eX1LxLPO+jtNdTj4nvngyt
XqJmcTC+OeBw1tprpMkxPE8Ja7TiMltXgrPE5K8+l557p/uQohGWL2bcL3z3T/AHIPPdP9yFEGT2bcL3z3
T/cg890/3IUQj7NuF757p/uQee6f7kKIMns24Xvnun+5D7XV4LTERkjmvCJyezbhpRx0dpvpqTPfw7JbYn
MZABIAAhbp7r/ADCahbp7r/MDnq7JU4CrzAAAAAAAGh0/sKeD3b1ZeNP7Cng929WVnqxtZ7L7W/7p
eHvL7W/7peFXl9ABAAAAACy2fuyLNWbP/AMxZph6OjshXbv6IPFVrTd/Up4qtEsmvvAEoIAkAAAED7
X148Wjr6seDOV9ePFo6+pHgtDX039ec3sb/ALZZ2e9os3sb/tlnZ7yTqfsACGQAAAAAAW+0+72/cqFvtPu
9v3Jh36fenKTcfe7fwu1JuPvdv4JaOo2IoCGAAQACQAAS9t98r4SiJe2++V8JF9Pfc6VO7e2p4LZU7t7ang
mW3X2IACHngAA+1pa3q1mX2209fWrMCcS8gCB10vvOP90OTrfecf7oFq/YaBG3D3LJ/58UIG3D3LJ/
wCfZ6V9sqMBV5YBxM90AD71bfKTq2+UicPgTEx3wCAAF7ofdKeCQj6H3SngkLPVptgAFgABC3T3X+YT
ULdPd5gc9XZKnAVeYJGixVzajqXjs45R07ao5z2n5QL6cZtEJfo/TR8J/J6P03yn8vG40y3tTyUW7u3hXXjp
T1/KR48pabzWs47TU0rjz2rTuhyJmZnmZ5kQyTOZ/AAQ0On9hTwe7erLxp/YU8Hu3qys9WNrO5fa3/A
HS8vWX2t/3S8qvLn6OmDFbPkilXN9raaWi1Zmj5BGM/q1roMGOInLbmf1nh68y0uSOKcc/pKry5r5rc3
tMy6aGbedUinPf2+A0RqUmcRV81Wmtp8T21nulwWm7THkqr8eVWWWrWK2xAAOaz2juyLJW7R/zF
kmHo6OyFdu/qU8VWtN39Sniq0Sya+8AHFZaTR4MuCt7xPM/q7+j9N8p/Koi94jiLTEeKft+nvkmMuS1ur
HdHPelq07Vt+dqRO3afj1Z/KoyxFct617omYhZ7hq/Jx5KnrT3z8lUsrrdsTisACGd9r68eLR19SPBnK+vHi0
dfUJwTDX039ec3sb/tlnZ72izexv+2WdnvJR1P2ABDKPWlhbkile+ZeY7ZXGg00YcfXvH9U/wCQ6adO+x
m2h0+HF1snPZHb2qq0xNpmSC8ITnfqZzX6IO2lf8ANCmJvgjW1ZjOKwADiLfafd7fuVC32n3e37kw79P
vTlJuPvdv4Xak3H3u38EtHUbEUBDAJ+h02LNim2Tv5+aA9Re1Y/ptMeEi9JiJzMLf0fp/hE/IVZ6RTNete6J4
hcaKZnS0mZ5nhU6r3nJ+5Mu+tEdsTEOICGUtt98r4SipW2++V8JF9Pfc7VO7e2p4LZU7t7angmW3X2IA
CHnid0NFGWPkzfV+EfnBiOZ4aClYphiKx3R2EO+hSLTmXjJnwaalrMxHyiHPz7T3iY5/iYRvR2XNab5LxWZ
7eHjLt2THXmkxf9Eu031P5H4iXtFr2tEcRM9zydsT2iGKR10vvOP90OTrfecf7oE1+w0CNuHuWT/z4pKN
uHuWT/wA+Kz09TbKjAVeWLjbYjzWPFTpWDW3wY+pWsTH6kOulaK2zKyvq8FLTW1oiY7+x7w5seaJnH

PMR39ijyWnLIm3HbaV1pMMYNPwvxntlTpak3t/xF3W1YpSnx55VjvrMvlRa3wjshwQy6tu60yADmvd
D7pTwSEfQ+6U8EhZ6tNsAAsAAIW6e6/zCahbp7r/MDnq7JU4CrzBK0Wprp5tNomefkip2k0MZ8M2tMx
M9w6acW7v8pHpTF8a2SMWbFqaTNe2PjEq+dry89lqzCbo9L5tSeZ5tPelqpOpM4t8VuvwRgzf0+rbthG
TNyyXkzxFZ9WENEsupERacAA5tDp/YU8Hu3qy8af2FPB7t6srPVjazuX2t/3S8vWX2t/3S8qvLn6A9Y8dst
4pxvkRjJTHbJaK1jmZW2DDj0WGb3n+r4y96XS102PsjrXmO2UTU4NXql8zTivwjmE4a60nTjOMyi6rUTq
MvWnujuhd8mjz46Te9eKx+rghtmt3Z/0ACqz2j/mLJW7R/zFkmHo6OyFdu/qU8VWtN39Sniq0Sya+8AH
EWeDcMWPFWkbmI47njFt0ZNPFptNbz2vPozLz61RopXUp+xCw4w6rFzx Fon4qXUYvI5rU+XcutNhjT4
epzz8ZIUa3JGXVWtXu7kyvr7YmfrogAqyPtfXjxaOvqR4M5X148Wjr6seC0NnTf15zexv+2WdnvaLN7G/7Z
Z2e8IXqfsACGVP27S9efK37o7kzWeWnF1MNeZnvn5lOm18YMUU6nLr6Vj7aWultOKYy97fpr4rXnLxm
I4fNfpcmXJWcdeYiHfSavznrf08cPOr1kae8V6vPMDp208f/FPMTW0xPfE8Pj1e3Xva3znI5VYBb7T7vb9y
oW+0+72/ctDv0+9OUm4+92/hdqTcfe7fwS0dRsRQEMAfAPgC80PulPBU6r3nJ+5baH3SngqdV7zk/clq1
tkOICGUStt99r4SipW2++V8JF9PfC7VO7e2p4LZU7t7angmW3X2IAChnsTxaj+TQ0tE44m07hnVpoNZ
WaRiyTxMd0z8Uw0dPalmYIHZ67NOW0Ut1YieyIT9BntnxTN47Ynh4y7diy3m8WmvPb2OtYxaPFxzERH
+Y7Vrets2n8Vm4Uimqnq9nPaiuupzeXzTf4fByQx3mjtOB10vvOP90OTrpfecf7oCv2GgRtw9yyf+fFJrtw
9yyf+fZ6WptlRgKPLAEiT0MPIdREz6te2Vlr83kdNPE/wBVuyHnbsPk9PEz327UHcc3INRNynsr2Ja4/wD
PS/7KJzyAhkAAxuh90p4JCPofdKeCqs9Wm2AAWAAElDp5hNQt091/mBz1dkqcBV5g7YdVlw+pbs+U
ulJiZj9hPjdMnHbSrnl3DNkjiOKx+iC86t5/oAOYADQ6f2FPB7t6svGn9hTwe7erKz1Y2s7l9rf8AdLy9Zfa3/
dLyq8uf094ss4ckXrxMx83gCJx+pvpTN9NPxJ6UzfTT8IQZX8t+UrLr8ubHNLVrxPyhFAUtabTmQBCFhtH/
ADFkrdo/5iyWh6OjshXbv6lPFVrtD/Up4qtEsmvvAbxSMOtzyY4ieYj4S7xumT40qgA6RqXj8iUnNrc2aOJ
mKx8oRgFjtNvoAlfa+vHi0dfUjwZynrx4tHX1I8Ew19N/XnN7G/7Z2e9os3sb/tlnZ7yTqfsACGQAB302qv
put1lrPPzfNRql6i0WvERMFjxBbunHaACot9p93t+5ULfad7fuTDv0+9OUm4+92/hdqTcfe7fwS0dRsRQE
MAACXi1+XFjila14j5wjZLzkva88czPPY8gtNrTGjkAFRK233yvhKKLbb75XwkX098Lt7t7angtltu3tqeCzb
dfYgAleeAA6VzZK+re0fy82va8/1WmfGXkE5kAEDrpfecf7ocnxs+84/3QJr9hoEbcPcsn/nxSubcPcsnh/us
9PU2yowFXlgAPUZLxHExt+XnvAMyAAAAvdD7pTwSEfQ+6U8EhZ6tNsAAsAAle6e6/zCY558Nc+KaW7p
FLxmsxDPCfbav8vM8WrMPnovN9VVcPP8AFfhBE70Xm+q6LzfVUweK/CCJ3ovN9VT0Xm+qpg8V+EETf
Reb6qvovN9VTB4r8lnei831Ve8W1260eUtHV+UGExo34WGn9hTwe7erJWlrWiju9WejEfmgcy+1v+
6XlaajbZvkm+O0Rz28S4+i831VRh59tK+fiCJ3ovN9VT0Xm+qqMK+k/CCJ3ovN9VT0Xm+qpg8V+EETvRe
b6qnovN9VTB4r8lnei831VPRex6qmDx4ddo/wCYsnDS6aNnj6sTzM98u6zfp1mtYiVdu/s6eKrX+q08aj
H1ZnifhKuna8vPZaswhm1tO02zCCJ3ovN86novN9VUYcfFfhBE70Xm+qr56LzfVUweK/CEJ3ovN9VT0Xm
+qpg8V+EETvReb6qnovN9VTB4r8lVfxjaOvqR4K7T7bNckWY2iYjt4j4rKEw1aFjrE5eM3sb/tlnZ72kmOY
mJ+Ksy7XabzOO8cT8J+BJr0tbGFcJ3ovN9VT0Xm+qqMMvivwgid6LzfVU9F5vqqYPFfhBE70Xm+q6LzfV
UweK/CCJ3ovN9VT0Xm+qpg8V+EFb7V7vb9yNG15ee21YhZafBXBiile39fmmHfR07RbMuqk3H3u38Ld
1mhjUT16zxb/VLtrVm1cQphO9F5vqqei8v1VRhi8V+EETvReb6qnovN9VUPFfhBE70Xm+q6LzfVUwe
K/CCJ3ovL9VT0Xm+qpg8V+EFK233yvhLp6LzfVVL0eh83t17Tzb/ROF9PSvFomYTFVu1Z8rS3HZxwtXnjr
kr1b1iY/VLzqU764ZwXvmOn+3B5jp/twjDL61uVEL3zHT/bg8x0/24MhrW5UQvfMdP8Abg8x0/24Mhr
W5UQvfMdP9uDzHT/bgwetbIRO2krNtVj47e3lb+Y6f7cPeLT4sPbjpESYTxp5icy6o+urNtHkiO2eEg45jtS
2WjMYzoXttFgtPM44fPMdP9uEYYvWtyoxeeY6f7cHmOn+3Bg9a3KjF55jp/tweY6f7cGD1rcqMXnmOn
+3B5jp/twYPWtyoxeeY6f7cPVdHgrMTGOOYMHrW5NHWA6WkTHbw7gltiMRgAEgAAAAAAAAAAAAAAA
AAA
AAA
AAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAA
5XwlESTt98r4SQvp74XYCz1AAAAAAAAAAAAAAAAAAffve4Tt23ZM9Y5v3V8QWEzEd8
wRMT3Swei0W57/wBfNOo/oi3HN7T3/om4Nk3nQavF5LPzjm0daaWnsj9Qa8Rtzr9NoKUtq80Y4t2RM
xPa66fUYtThrmwW6+O3dMA6CFm3fQ4NV5t1Fa5eYjq8T8e511ev02hpW+qyxjrbsiZie0EgV+TfNtx463
tqq8W7Y457XXS7no9XS1sGopaK9/w4BLFZbpBtIMnUnVV5+fE8LDFlx5scZMV4vSe6YB7Bluke+5NPqM
VNBqK9kTGSljukGpFbtW66fV4MNJ1Fb6i1e2sd/Kdn1GHTY5vnyVpWPjMg6Csr0h2u1+rGqr4zErHHkp1

pF8dotWe6YkHpwz6zTaa0Vz56Y5mOYi08O7F9Nv7ywf9H/eQbOt0tWLVmJie6YfVRi3vb9JpsOPNqKxeKxzEczwsNLrdPrKdfTza5i/QHceMubHgpN8t4pWO+Zlxx0h2ucnU86rz8+j4Bzjzy0zUi+O0WrPdMS9ADzkyUxUm+S8VrHfMyrp6RbXF+rOqr48SCzmYijme6HDFrdNmy+TxZ8d7/TE9p5fFqNJfjhvF6zWe2J/RiuJmbHg36+TLeKUiL8z Mg3grKdINsvk6kaqvPdHZPCyraLVi1ZiYnumAfRF1e5aPQ+8560n5d8uel3jQay/Uwams2+U9gJwlGq3rb9JfqZtTWlfKO0E9wza3TafJ1M2fHS8zxaeJc9Huej13Zps9bz8u6WS6X/AN+4/wDp1/1kG3iYtETE8xPxfXDFkpi0VL5LRWtaRMzPw7HDTbxodXm8lp88ZL/KlkE4cdTqsGkx9fUza46/OZQ8XSDbMt+pXVV5nu5iYBZD5W0WrE1mJie6YRdZumj0M8ajPWlV3yCRIzY8GOb5b1pSO+bT2PODUydtWbYMtckR3zWeVBv26aPXbJljTZ63tExzHdJ0K9wzfvGkBDtu2hrq/NZ1FYzd3V/UEwABSbj75b+F2pNx98t/CJZ+o2loCGB6pScmStK98zwlejc/8Ah/LjpPef7oW+qzzgxdeI5nlLvpUraszZxejc/8Ah/J6Nz/4fy7Y9yyZMlaRjjmZ471hkVGPHNrfCDDrXT07RmGevWaXms98Txl49Xt172t855eUMk/RK233yvhKKlb75Xwkbt3uwFnqAAAAAAAAAAAAAAACFu+3xuWgvp+t1bT21n5Smq/e9dl0G32zYKtfJzER2cxAMpTS75sI7Rgrfqc8z1I60SnaLpdkrJh8MRHPE2r2THi64OmOKcfGp01ov8erPPKI3PUzvm40800015/p7Pj+sguemI65NFo70nmtrTMT+nC16N/wBx6fwV3SLbss7Fpq0ib208R1ojwV20djfR+i82yYJvNfVmJ4/gHLev+K4/6mP/AGWnTT3HTfun/Znb58uq3vHnzV6tr5azx8o5jhounmuOl/dP+wImxdHMOv0HnOpvfm3ZWI+Cnz6Pj91vosd5iZv1OeeOYn5/ltoJh9x4f5znXf8YR/16f7An7h0W0+m2u+bHkvOXHXrTz3S99CdReaZ8EzM1r/VEfje7v8A3RqP2M70l9vqf2/7g189zCdK9vxalWY74pnnP1rW5+fLdsr010+S8afPWszSnNZ4+HIJewbJpsOPT66s38rNee2eztUe75s27dIPNIvMu/UrHw8VtsfSHBfHptDbHaMnqc/BW75o9Rtm8+fYqTaI9eJj4T8gWep6JaTz0OYbXjLWvMWme+ULojrcuPW30N7TNJiZiJ+Ew66jphXj07Uxae0ZrV47Z7IfOiW25o1F9fmrNYmJivPx57wa1ium3954P+j/vLasV02/vLB/0f95BO27orpcmhpfU2vbLevPMT3KjD5XYekUYa3ma9eKz+tZ//AMIYaHpv5ro64dVp7TkphETHxQdBh1G+77GpvSYpF4tafhER3QDp0p199RuUaWbzTDj45j9fm9WxdHPNjpXNfyvHzk4nv8HvpTtmXFrY1uKnXxW4636S+Yt72ryMeW2yvleO2I7pkHrohrb0199H15tjvEzXn4cNko+juXTayls+PQ10+SkzHMR2TC8Bi+lesy6jdK6GIpilerrHzmf/ALWOLojo50kVva85Zj1+e6ULpXt2emtjcMFZtWeOtx8jh1x9MKV0kVvp7eWiOO/skfjtOz32rS54vqJydes/0x6sMjt2g9JbvOn601rNrTaY+UNXsm5azcdFntqsMvrWs8Xjs5/hR9F/+IreFwd9+604NBofONPe3NZ4tFvimdHdwvj6O6JaZtOn56vPhzCd0p/uTL4wreiUCNTsutwz2eUtNefi/SckOOTsazXZM275rdXv47Z5dN2ja6dTltWW0Xie2vb2frzLzgidm3C9NfpIzU7uLf6wsMm87Xa9K6fbK35mOeY+AJP3nN/wClceWtpjNf+zm3/n6lvR/YMO46a2q1drWi1pilif8ANC7tddZskY9JijHaP6604/yUWzb9bZ8VtJqsFprFuY+ExLTB0Wrptypnwam9MdZ5i7+fI4Knph2b5SZ+GKv8ArKx0nSTV67dk49NpecE9kx8Y/XIxDMI53ukfPFX/AFkHHc95y7nOLSYreTwRxXtnjn9ZazZdpw7Zpo6vFsto5tf5qLcOjNabTjz6Tm2Wletf/Ek9Ft6nLWNDqrf119S0/H9AU256yu4b5MarLNNPW/Vjj4Q76/H0fto7eaZb0zVjsnif6p/I43jQZdr3edROGMuC1+tHMDk/pKTfe9p8j/RtlZy8d093IO/Rrdclsd1dMkzbzevWrM/qrNm0Ft93HJfVZLdWP6rT8Z/Rp9uw6bXbRe+LSxp5zV6to4ZjQarUdHdyvZ4pms/02j5x84BN6QdHsGh0XnOlm0RWeLmeU7oV7hm/fCt3vpDO56TyGnw2rj772lZdCvcM374BpGby9FpybtOqjURGKbdaa8drSMrm6Saym9zp4xV8If+r1eO2QaoABSbj75b+F2pNx98t/CJZ+o2loCGB9iZieYniY+l7bJe0cWvaY+Uy8vsRNpil75E5n4n7Xh5vbLMdkdkOu6ZurjjFE9s9spOnxxg09a93Ecyp9VlnNntb4d0Jarf+enj+y4glZBK233yvhKKlb75XwkX098LsBZ6gAAAAAAAAAAAAAAATTHE9sACHI2nQZp5vpMUz8+pDrp9Hp9L7DBjx/rWsQ7gExz3ok7XoZyeUnSYet8+pCWA4X00kyZYyX02K1444tNlmezues+mwaisVz4aZyui9Ynh1AeMWLHhpFMVK0pHdWscQ520Oktm8tbTYpy889eaRzz4u4DzelclJpesWrPZMTHMS54NjP9NMzp8GPFM9/UrEcuwA83pXJSaXrFqz3xMcxL0Ai4tt0WHJ18elxVvHbExSOYSL0krNb1i1Z74mHoBErtehrfrV0mGJ+fUhLiliOljiAAC+j02ptFs+nxZbRHETekTxDsAjZtu0ee0TI02K0x8ZpDtiw48FOpx1pX5VjiHsB8msWiYtETE/CUS21aC1+tOjw8/shMAeaUrjrFaVitY7oiOlegAmImjiY5ifgiTtegm/WnR4Zn59SEsB5ila06laxFeeO0ljscsWi0uDJ5TDp8VL/VWkRLuA8ZcOPPSaZsdclJ/9to5h8wafDpqzXBpirM8zFKxHLoA55tPh1FernxUyR8rV5csO3aPBfrYtNipaPjFI5SQBHz6HS6m3Wz6fFkn52rEyKAOWDS4NNExgw0xxPf1axDzm0WI1GTr5tPiyX4461qRMu4D5FYivVili7OEeNv0dcnlK6XDF4nnrRSOeUkB5vjkpnNclYtWe+JjmEau1aCt+vGjwRP7ISwClil4i0lhxZ6XbqY4z4aZlju61Yi2ARqbfo6Y5x102KKW746kdrgo2HTVmuDFTFE98UrEOoA4To9NOo8vODHOX6+rHP5dwAABSBj75b+F2pNx98t/CJZ+o2loCGAd9HOKmelZrcVjt7nAExOjytNXrsVsE1xW5mezuVYC17zecyAcgIbb75XwlFStt98r4SL6e+F2As9QAAAAAAAAAAAAAAAUM4++W/hdqrcdNknN5SIztE/JEs+vEzT8V49+Ry/bt+DyOX7dvwhh7Z4eB78jl+3b8Hkcv27fgO2eHge/l5ft2/B5HL9u34Dtih4HvyO

X7dvweRy/bt+A7Z4eEvbffK+Eo/kcv27fhN23TZK5vKXrNYj5wQ6adZ74WoCz0gAAAAAAAAAAAAAAA
AAADiDiAA4g4
gAOIOIADiDiAA4gAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEPcc9sOKlp2
Tae9VTnyz/AMy35T939WnirFZYNe09735fL9y35ffL5fuW/LmDhmXTy+X7lvyExy/ct+XMDMunl8v3Lf8
vl+5b8uYHdLp5fL9y35PL5fuW/LmB3Sk6bVzaZa83mYmeJiV58Gcx+Or4w0VfVhMNnTTMxOxy9urSbfK
FFk1WXJebTeY/SF5m9jfWZ0IHUTMYh08vl+5b8nl8v3LflzEMmZdPL5fuW/J5fL9y35cwMy6eXy/ct+Ty+X
7lvy5gZl08vl+5b8nl8v3LflzAzLpGfLE8+Ut+Vzoc1s+ni1u+OyVEuNq91nxTDR09p7sJoCW4AAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAABVZukW3YM18WTJaL0maz2fEFqKivSbbLTx5aY
/WYWeDUYtTjjJgyVvSfjEg6AADxnyTiwZMKR1ppWbcfPiGd2TpBq9fuc6fPhrFjiZiaxxNfEGIAAAAAAAA
BW7v6tPFWLPd/Vp4qxWXna+8AHEErQaemotaL88R8k30bg/xfkda6VrRmFQLf0dp/1/KqyRFclojuiRF9
OafXkAc3rH7SvjDRV9WGdx+0r4w0VfVhMNnTf15zexv4M60Wb2N/BnSVep+wAIZQddPp7ai/Vr2cd8r
Gu34Mcf2luZ/WR0rp2tGVSLa234Mlf7OeJ/SVbnw2w5Jpb8hfTtX65gDmLjavdp8VOuNq92nxId+n3poC
z0AAAAAAAAAAAAAAAAAAAAAAAB+eZcFNT0myYckTNL6i0Tx4v0
N+cazNkwDIM+XDXrXrntMRMd/aDUZ+iW32xTGKL478dk9aZUnRrU5dHvcaSLc0yTNbR8Oz4/wCTl33e
9RScdNP1Zt2c0pPKX0a2LPg1PnmtrNLR6lZ7+fniJe8dIcmg1fm2DS2yZOOeZ7p8OFdXpdqsWSI1WjtZ8Y
IL3Tf6Ydx830mlrnz1/p60xz2/KFZvmo3LUaKJ1mgphpFo4vFziY/QGuwa3FqtD51gnrVmszH/wpNj3nzd
b4I0mHFzEzN6R2zwdFZmdhzxm9kdbj8Sq+in/ABBf9tgW28di8u27n5tGClqRFZm0zPPa+aDpLk1+6U0+
LT9XDafWnvU/SqvX6RdWe6a0hsdDpMGm0uKuLFWvFY7e00FJuHSqceqt9Dp/LWrPHPb2+HDnpuk2
unUY8Wo0Ex17RWoYnt8UHLtG6bVuFtRo8U5o5niYjs/VJw9KM2LPXHuWjpWYnvivEx+vaDXR2xHZ
wrt73Odq0Xlq4+va1urEfBYUtF6VvXutHMOep0+HVYXzailb4574kFfsO723bTXvfHFL0txPHdK1cNHpdPp
MPk9LjrSnf/S7gAArd39WnirFnu/q08VYrLztfeADis9pr/TknwedfGec/9I1+rx8HPQ6zHp8U1vE8zPPYlek8
M98W/CWus1nTiJnCttPSf6rZ18Zly71/HktTi5iltWVLqcXkM9qfD4lctXTmsZzlyAHF6x+0r4w0VfVhncftK
+MNFX1YTDZ039ec3sb+DOtDn9jfWZ4IXqfsACGV7xZb4b9ak8SZMt8tpte0zLw94sVst4rSOZFomfkJW2T
fzjiJ/p47XvdpjyII+PCTSULQYObTzaf85VWfLbPlm9vj3fol3v/AI0+2frmAhmFxtXu0+KnXG1e7T4kO/T700
BZ6AAAAAAAAAAAAAAAAAAAAAAAXtdn1//qOdROmt5Hy8263
MccctkAcR8gAZDdNI1+n3adbt9fKc260cd8S867T7/uulmNRiitaTzGOOlmtiAoujuh1Ok2jNi1GKaZLc8V
mY7exA6O7VrtJvNs2o09seOa2/qmYawBkt/2rXarfYz4NPa+Lin9UTHwarHE1w1jj+qKw9gMllr0k0movO
PnLW1ueyeY/zcJ2jd951dMm4VjHWvZMz8v04bQB4x0jFipSvdWliFZ0j0uq1e2Tj0fM2i0TasTx1o+S2AU
nRfR6vR6G9dXE15tzWkzzxC7AAFbu/q08VYs939WnirFZedr7wAcU/Dt3ldPF5txae55nbM3PfWf5c8O
uzYY6sT1q/KXf0rbjtxxylojxTH6m6XB5vh6szzPfKq12SMmpntNZ7I7HrNr8uWvVj+mP0RUI1dSJjtqAdg9Y/
aV8YaKvqx4M7j9pXxhoq+rHgmGzpv68Z/Y38GealN7G/gzpKvU/YAEMrphw2z5IpTvXGLBXTYZ8nXrW/
wBVRp89tPebViJ5j4pPpPL9FRo0rUrGZ+vmfDq89+tak/pHPcj5dNlxV62SnEeKT6Uy/RVY1GtvqMfUtWij
nnsFb9k5nP6jAdiLjavdp8VOuNr91/IMO/T700BL0AAAAAAAAAAAAAAAxtdn1//qOdROmt5Hy8263
AAAAAAAAAAAAAAAAAAAAAAAFbu/q08VYs939WnirEPO194AhxAaaaaAesftK+MNFX1Y8
Gdx+0r4w0VfVhMNnTf15zexv4M60Wb2N/BnSVep+wAIZQAAAABcbX7r/KnXG1e6/wAkO/T700BZ6A
AAAAAAAAAAAAAAAAAAAAAAADjqdPX
UY+rbs47YIB9E2+7H4WgOdtKtpzKr9E2+7H/aeibfdj/tWgK+Cir9E2+7H/AGnom33Y/wC1aBg8FFX6Jt92
P+09E2+7H/atAweCir9E2+7H/aeibfdj8LQDwUV+Dblx5lte/W47Yil4WADpWkVjEPkxzExPxV2Taom8zS/
EfKY5WQItStvqr9E2+7H4PRNvux+FoCngog/RNvux+D0Tb7sfhaBg8FFX6Jt92P8AtPRNvux/2rQMhgoq/
RNvux/2nom33Y/7VoGDwUVcbTPPbl7PBY4cVcOOKV7oewWrp1r+wADoAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAA55s1MNOveelRPSuL6LPO7T/RSPhy
q0ZZNXWtW2IW3pXF9Fj0ri+iypEZcvYutvSuL6LHpXF9FISGT2Lrb0ri+ix6VxfRZUhk9i629K4voselcX0WVI
ZPYuusO4Yst4r21me7lLzZHPGWvHzhoo7oTDTTo6k3icvszxCFk3LFS81ijtx8YSs08Yb+DOko1tSafkLb0ri+i
x6VxfRZUhln9i629K4voselcX0WVlj7F1t6VxfRY9K4vosqQyexdbelcX0WPSuL6LkkTk9i62jdMXPq2hNx3
rkpFqTzEs4udrnnS+EmXbR1bWtiUwBLUAAAAAAAAAAAAAA
AAAAAAAAAAAAAAARd29WnirFnu3q08VYrLzfflAOlOtdNlvWLVxzMT8X3zTP9u34Fu23Di
O3mmf7Vvw4zExPE94iYmPoAlesftK+MNFX1Y8Gdx+0r4w0VfVjwTDZ0vyXjP7G/gzzQ5/Y38GeJV6n7AA
hIAAB1rps1qxMY7cT+jnas1tMT2TAmyMPr4AlFxtXu0+KnXG1e7T4kO/T700BZ6AAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAACt3b1aeKsWe7erTxVisvO198
gA4rDT7hjw4K0mtpmISMOurmrvFaY7f/CnXeiwVxYKzHrWjmZS1aN72nH8dM+auDFNrf8A2obW61pt8
5Wev02bLPXi3WrHdWFx3Eq9RaZnAAhnesftK+MNFX1Y8Gdx+0r4w0VfVjwTDZ0vyXjP7G/gzzQ5/Y38G
eJV6n7AAhIEQaac2TrWj+iv+bhgw2z5YpX4/Fcz1NJpvIFY/I76VM/6n4563URp8XVpx157v0U8zzPMpW
LFbX5b2tbiXPVaedNeKzbnmOUmr3X/wBfxwAQ4C42r3afFTrjavdp8Uw79PvTQEvrQAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAMAAAAAAAAMAAAAAAAAMAAAAAAAAMAAAAAAA
vkAHF66lur1urPV+b3XU5qxERktEQstNqNPbDXHzEcR2xZ18lpreeerj/AMktNdl8zWz7pMI8unra8dsqnW
1iuqvFe7Iz59Ziw04rMWt8lhT3vOS82t3zPMkp1pjtv8AXkBDK9Y/aV8YaKvqx4M7j9pXxhoq+rHgmGzpf
kvGf2N/Bnmhz+xv4M8Sr1P2ABDKttppWuDr/wDume9IzYceeli8936qOL2iOlMR4nIL/Xb8py011oiuML
zBp8eCZ8nHe+Z9NizT1skdsQh7blilv5S/hzLzuWXnLXd+y4+EjtOpXszhCyViuS0R3RLyd4hgFxtXu0+KnX
G1e7T4ph36femgJegAAAAAAAAAAAAAAAAAAAAAAAAMAAAAAAAAMAAAAAAA
AAAAAAAAAArd29WnirFnu3q08VYh52vvAEoI+8z85fAAAAAhrH7SvjDRV9WPBncftK+MNFX1Y8Ew2dL
8I4z+xv4M80Of2N/BniVep+wAlZQAAAABcbV7tPip1xtXu0+jDv0+9NAWegAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAMAAAAAAAAMAAAAAAAAGbpitfFW1Y56s9qpaXh48IT61/C
MM+pod85yzo0Xksf0R+DyWP61/Bhz9aeWdGi8lj+iPweSx/RH4MHrTyzo0Xksf0R+DyWP61/Bg9aeWdGi
8lj+iPweSx/RH4MHrTyotPjtIzVikTPb2/o0ERxEQ+VpWvqxEPqXfS0/HDzkr1sdo+cM9kpbHea2iYmPm0b
zNK276xKEaul3s4NF5Kn0x+DyVPpj8GHH1p5Z0aLyVPpj8HkqfTH4MHrTyzo0XkqfTH4PJU+mPwYPWnl
nRovJU+mPweSp9Mfgwj1p5Z2I5niF3t+K2LTF44mZ54d/JUj/2x+Hsw66Wj2TkAS0AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAMAAAAAAAAMAAAAAAA85MlcVjteelhF9J6f52/A
rN6x9IMEP0np/nb8HpPT/ADt+BXyU5TBD9J6f52/B6T0/zt+A8IOUwQ/Sen+dvwek9P8AO34DyU5TBD9
J6f52/B6T0/zt+A8IOUwR8WtwZr9Wtu39YSBelRPwHLU5owYzv8fgqp3LPM9kxH8DnfVrScSuhs+kdr8
4/B6R1Hz8lyp7FF0KX0jqPnH4PSOo+cfgyexRdCI9I6j5x+D0jqPnH4MnsUXQpfSOo+cfg9I6j5x+DJ7FF0kf
HuWaLx1+Jr8exb1tFqxaO6Y5S6U1K3+PoA6AAAAAAAAAAAAAAAAMAAAAAAA
AAAAAAAAAAAAAAAAAAAAAK7d5nqUjns5Vaz3f1aeKsVI52vvABxAAAAAAesUzGWkx84aKvdD
O4/aV8YaKvqwmGzpvkoe6e6/wAqdcbp7r/KnJc+o3gCGcAAAAAAaHT+74/2wzzQ6f3fH+2Ew1dN9l0AS
2gAAAAAAAAAAAAAAAAMAAAAAAAAMAAAAAAAAMAAAAAAAAMAAAAAAAAMAAAAAAAAMAAAAAAA
Vp4qxZ7v6tPFWKy87X3gA4vWPHfLbq0jmXXzLuFbk0meNPI68xz2cdi20uqjUxaYrMcfnMO+nSt/s/qlyY
r4piMlerMvCdvvFf2oKHO9e22AAUesftK+MNFX1YZ3H7SvjDRV9WEw2dN/UPdPdf5U643T3X+VOS59
RvAEM5ETMxERzMp2LbL2jnJbq/o97Xgiecto547IdddrLYreTxeTpfpPyS0006xXus8ztdOOy88oOp086fj1zt
EvUazUVnmbz/LillyWy3m9++UKXmkx/mHgAcRodP7vj/bDPNDp/d8f7YTDV032XQBlaAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAMAAAAAAAAMAAAAAAAAMAAAAAAAAMAAAAAAAAMAAAAAAA
r4piMlerMvCdvvFf2oKHO9e22AAUesftK+MNFX1YZ3H7SvjDRV9WEw2dN/UPdPdf5U643T3X+VOS59RvAEM71jyWxXi1J4mfjj3W
3dOQAvesftK+MNFX1YZ3H7SvjDRV9WEw2dN/UPdPdf5U643T3X+VOS59RvAEM71jyWxXi1J4mfjj3W
vHGsk8/OFYGV66lq/Fpk3SnH9FJmf1V2bLfNkm155eATbUtf6ADmNDp/d8f7YZ5odP7vj/bCYaum+y6AJ
bQAAAAAAAAAAAAAAAAMAAAAAAAAMAAAAAAAAMAAAAAAAAMAAAAAAAAMAAAAAAA
q08VYtd1x2tjral5i7VUrLztfeAdiAAAAAA9Y/aV8YaKvqwz+Gk3zUisczy0EdkQmGzpvkoe6e6/yp1/qsPn
GGac8T8FTOg1ETx1eSvdelptmlRhI8x1H0HmOo+hGHDx24RxI8x1H0HmOo+gwdiuEcSPMdR9B5jqPo
MHZbhHEjzHuFQeY6j6DB2W4R2h0/u+P9sKnFt2e1460RWvxlc0r1KRWPhHCYaeNPnczL6AlrAAAAAAA
AAAAAAAAAAAAAAAAMAAAAAAAAMAAAAAAAAMAAAAAAAAMAAAAAAAAMAAAAAAA
w6giYiXLzbD9uv4PNsP26/h1BHbDI5th+3X8Hm2H7dfw6gdsOXm2H7dfwebYft1/DqB2w5ebYft1/B5th
+3X8OoHbDxTFSnqViPCHsBMRgAEgAAAAAAAAAAAAAAAAMAAAAAAAAMAAAAAAA
AAAAAAAAAAAAAAAAMAAAAAAAAMAAAAAAAAMAAAAAAAAMAAAAAAA
pPP8A4fwek8/+D8IYZU8t+Uz0nn/w/g9J5/8AD+EMMnlvymek8/8Ah/B6Tz/4fwhhk8t+Uz0nn/wfg9J6j/
B+EMMnlvuuNDrZ1Fppeli3f2fFNUu2+9x4SulobdG02rmQAdgAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAeL5sdJ4vkpxxtEPtMIL+petvCeQegAAAAAAAAAAAAAAA
Um4++W/hFStx98siqvL1N0gPtI6161+c8Cj4LONqrMe0n8Hoqv3J/Bh28F1YJ+fbq4cNr9eZ4j5IA52pNZxl
AKpe2+9x4LpS7b73HhK6TDf0+wAS0AAAAAAAAAAAAAAA
AAAAAAAAAAAAACi6Sb1bbcdOnmPL5I55+mPmvWC6U8zv14yerxXjw4gHvR7HuW7084zZ5pW3bFsk
zPP8Os7LvG16nHOky2yRM8c0ns/mGx08Vjt44px1erHHDoDnS1senrbU2rFor/AFz3Qq83SfbMWsAtm
m0x3zWszCv6Z6zJx4dNSzrF+bW4+L3s3RzRZntxztTWcmTLXrd/dyC70W5aXX1mdNli/HfHdMfw57ju2
k2yaRqr2r5Tnq8Vme5jtXjtsO/RGnvPUrMTH6x8lh01nreZW+dbT/oDV6fPTU4KzsU80vHMTMINt90Ndf
5nOS3lut1eOrPe6bj/c+l/wCnD15/+Mf/AO8f7A1+4brpd6nnV7V6/dxWZRs/STbcNazOabTaOYiscz/ACq
em/8A+r/L1snRzSajbaZ9VFrXydvZPHEAvNv3fR7jzGmy82jvrPZKo6Ubz5vWmDR55pnrf+ulj4cKOmKdr6
SxjxWnimTiP1hadMdDgxUpq6VmMuTJxaef0BN2LfcGXS4sOpzzbU2tx2x3tAzPRrZ9Jm0ODWXrM5otZE
8tMCHuG56bbcdB6m/V63dERzMu2l1WLWaeubBrfut3Sgb3std3pj/tZx3pZPHKvtmgptuipp6Wm3V7
ZmfjIJYAKTcffLlqVuPvlkVV5epuklmYmjvgBRIjWaj7k/hZ6OM3k+vntPM/CfgqNPNa6ik39WJ7VzXVYLz1
YyRMz8Ew16M5/ZIA3DWeUmcWOf6Y75+aCtNZoaeTm+KOJjt4+arRLlrRaLf6ABxS9t97jwXSI233uPBdj
hv6fYAjaAAABnulGzX
11K6nTV5y0jia/VDQgMPt/STV7bijTajD14p2R1uyYdb9JNy3DUY66HDNYi3q1jnxa3LpNPmnjhpfN
XrFp8OH2WKIP2xwCh6SbZn3DQYc9Kf2+OP6qR8VZt/Sflt+kRPC+nm0444rz2Tx+rauNtLp72618000/Oa
wDFaTTarpBu/nGak1xdaJtPHZER8IXHS3b8mo0eHLhrNpw8xMR38T/8ATQ1pWkcUrFY+URw9TETHaD
D7d0nz6LR10s4lyWr2Vn4oOjtlv0hw31ETGS2WjtEv0DzTTxfrxgx9b59WHryGKb9ecV0t9XVjkGX6b/8A
6v8AK82H+5tN+3/dNvix5ePKY6347utHL1WsVrFaxERHdEQDCbp/xXb/AKKl/pdpcup2ytsVztOK/Wml+
XcuZwYbX684qTb5zWOXSY5jiQYvZOkUaHTY9HkwzM9bjrR8lRxjSaeL9aMGPrfPqw7Aoek+6arbseLzX
+nr995jIP2TV5tdtmPPqk8Xns7uOf1TcmLHrxkpW8fk0cvtaxWsVrEREd0RAPoAKTcffLlqVuPvlkVV5epu
kfaVm9orXvmXx6x3tjvFqzxMCkf9Ssm3Zq8TTi0POLQ55yRzTjie+Zd8e6TEf2IOfB0ndKcdlJS0xXS+5S8tox
4LTaeyKs/PekanWZNR2T/TX5QjoU1tSLz+AA4pe2+9x4LpS7b73Hgukw39PsAEtAAAAAAAAAAAAAAA
AAACk3H3yyKlbj
75ZFVeXqbpABQAAAABL233uPBdKXbfe48F0mG/p9gAloAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAVuv0WTJl8pjnnvhE8x1H25XojD
hbQracqLzHUfbk8x1H25XoYV9aqi8x1H25PMdR9uV6GD1qqLzHUfbk8x1H25XoYPwqovMdR9uTzHuf
blehg9aqv0GjviyTkydk8dkLAEu1KRSMQAC4AAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAD//2Q==></figure><p>Constraints</p><p>0 <n>100</p><p>0 <arr>1000</p><p>Input Format
<p>Output Format
&brsp;&brsp;&brsp;

 -Output contains as many lines as number of pairs with each pair written in each line

 -Last line of output contains the total count of pairs.</p><p>&brsp;&brsp;</p>
First line contains the value of n that is the total number of elements in the array
&brsp;&brsp;&brsp; -Second line contains the elements of array
&brsp;&brsp;&brsp; -Third line contains the Sum to be checked.</p><p>Output Format
&brsp;&brsp;&brsp;

 -Output contains as many lines as number of pairs with each pair written in each line
&brsp;&brsp;&brsp; -Last line of output contains the total count of pairs.</p><p>&brsp;&brsp;</p>

```
#include <bits/stdc++.h>

using namespace std;

/*int getPairsCount(int arr[], int n, int sum)

{

    return 1;
```

```

}*/
```

```

int main()
```

```

{
```

```

    int n;
```

```

    cin>>n;
```

```

    int array[n];
```

```

        for(int i=0;i<n;i++)
```

```

            cin>>array[i];
```

```

        int sum;
```

```

        cin>>sum;
```

```

        int count = 0,i,j;
```

```

        for(i=0;i<n;i++){
```

```

            for(j=i+1;j<n;j++){
                if (array[i] + array[j] == sum){
                    count++;
                    cout<<"["<<array[i]<<" "<<array[j]<<"]"<<endl;
                }
            }
        }
```

```

        cout<<"Total Number of Pairs:"<<count;
        //getPairsCount(arr, n, sum);
        //return 0;
    }
```

question

<p>Problem Description:
Good news! Suresh get to go to America on a class trip! Bad news, he don't know how to use the Dollar which is the name of the American cash system. America uses coins for cash a lot more than the Kuwait does. Dollar comes in coins for values of: 1, 2, 10, 50, 100, & 500 To practice your Dollar skills, suresh have selected random items from Amazon.co.us and put them into a list along with their prices in Dollar. Suresh now want to create a program to check suresh Dollar math.

Suresh goal is to maximize your buying power to buy AS MANY items as you can with your available Dollar.

Input Format:
File listing 2 to 6 items in the format

of:

ITEM DDDDD
ITEM = the name of the item you want to buy
DDDDD = the price of the item (in Dollar)

Output Format:
Print the output in a separate lines contains, List the items suresh can afford to buy. Each item on its own line. Suresh goal is to buy as many items as possible. If suresh can only afford the one expensive item, or 2 less expensive items on a list, but not all three, then list the less expensive items as affordable. If suresh cannot afford anything in the list, output "I need more Dollar!" after the items. The final line you output should be the remaining Dollar he will have left over after make purchases.</p>

answer

```
#include <bits/stdc++.h>
using namespace std;
int main()
{   int money,n;
    cin>>money>>n;
    int price;
    string name;
    map<int,string> mp;
    map<string,bool> mp1;
    vector<string> vecs;
    for (int i = 0; i < n; i++) {
        cin>>name>>price;
        vecs.push_back(name);
        mp.insert({{price,name}});
    }
    price = money;
    for(auto pr:mp)
        if(pr.first<=money){
            money-=pr.first;
            mp1[pr.second] = true;
        }
    else
        mp1[pr.second] = false;
```

```

for(auto s:vecs)
    if(mp1[s])
        cout<<"I can afford "<<s<<endl;
    else
        cout<<"I can't afford "<<s<<endl;
    if(price!=money) cout<<money;
    else cout<<"I need more Dollar!";
    return 0;cout<<"char name[MAX][LEN];int price[MAX]afford[MAX]for(i=0;i<items;i++)";
}

```

question

<p>Question description</p><p>Professor Shiva decided to conduct an industrial visit for final year students, </p><p>but he set a condition that if students received a passing grade in the surprise test, </p><p>they would be eligible to go on the industrial visit. </p><p>He asked the students to study a topic linked list for 10 minutes before deciding to conduct a surprise test.</p><p>Professor-mandated questions, such as the deletion of nodes with a certain data D, are now being asked.

For example </p><p>if the given Linked List is 5->10->15->10->25 and delete after 10 then the Linked List becomes 5->15->25.
 </p><p>Constraints </p><p>1<lt; N < 100</p><p>1<lt; D < 1000</p><p>
Input Format
First line contains the number of datas- N. </p><p>Second line contains N integers(the given linked list). </p><p>Next line indicates the node data D that has to be deleted.</p><p>
Output Format
Single line represents the linked list after required elements deleted.</p>

answer

```

#include <bits/stdc++.h>
using namespace std;
void mandatoriesSuck(){
    cout<<"struct node node *next;void create()p2=p2->next;void del()";
}
int main()
{
    int n,t;cin>>n;

```

```

int arr[n];
for (int i = 0; i < n; i++) {
    cin>>arr[i];
}
cin>>t;
cout<<"Linked List:";
for (int i = 0; i < n; i++) {
    if(arr[i]==t)continue;
    cout<<"->"<<arr[i];
}
return 0;
}

```

question

<p>Question description</p><p>Varman's Dream came true after he got an Appointment order from Google.Simon's family was very happy of his achievement. </p><p>The company mentioned Basic Salary, DA, HRA with some other benefits. </p><p>But not highlighted the Gross salary in the order. </p><p>varman's father wanted to know the Gross salary of his son. </p><p>varman try to his gross salary from HR department. they informed that you have to get pass grade in first month entry test. the entry test has 5 questions. one of the question was, Sorted insert in circular linked list.</p><p>Can you help varman?</p><p>Function Description</p><p>First case one is if linked list is empty then since new_node is only node in circular linked list, make a self loop.and change the head pointer to the new_node pointer.</p><p> Second case is new node insert in starting or before the head node.</p><p>A- Find out the last node using a loop .
 While(present->!*head_ref)
 present=present->next;
B- Change the next of last node;
 present->next=new-node;
C- Change next of new node to point to head.
 new_node->next=*head_ref;
D- Change the head pointer to point to new node.
 *head_ref=new_node;</p><p>Third case is when we insert the new node after the head in any position,then</p><p> A- Locate the node after which new node is to be inserted.
 while(present->next!= *head_ref &&lt;br> present->next->data
 { present = present->next; }
B- Make next of new_node as next of the located pointer
 new_node->next = present->next;
C- Change the next of the located pointer
 present->next =

new_node; </p><p>Constraints</p><p>0<lt;n<100</p><p>Input Format:
The First line of the input represents the number of elements</p><p>Second line represents the elements of circular linked list</p><p>
Output Format:
single line prints the results as per sample test cases</p><p> </p>

answer

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int n;cin>>n;vector<int> v(n);
    for(auto &el:v) cin>>el;
    sort(v.begin(),v.end());
    for(auto el:v) cout<<el<<' ';
    return 0;
}

cout<<"struct Node *next; void sortedInsert(struct Node** head_ref, struct Node* new_node)">>
```

question

<p>Question description</p><p>sanam's Dream came true after he got an Appointment order from Google.Simon's family was very happy of his achievement. </p><p>The company mentioned Basic Salary, DA, HRA with some other benefits. </p><p>But not highlighted the Gross salary in the order. </p><p>sanam's father wanted to know the Gross salary of his son. </p><p>sanam try to his gross salary from HR department. they informed that you have to get pass grade in first month entry test. the entry test has 5 questions. one of the question was, Split a circular linked list in two halves. you have to split the circular Linked List with the same size of Divisions.Maybe if circular Linked List is odd, you have to change the number of node, it is even .</p><p>Can you help sanam?</p><p>Function Description</p><p>First count the number of node in Circular Linked List.</p><p>Second, you have to make the List even.</p><p>Third, you need to make the List half. the front is the same size like the rear. Finally, I have to make two circular Linked List.</p><p>Constraints</p><p>0<lt;n<10</p><p>Input Format:
The First line represents the number of input in the circular linked list elements

Output Format:
First Line indicates the complete linked list</p><p>second line indicates the odd list</p><p>third line indicates the even list</p><p>Refer sample test cases.</p>

answer

```
#include <iostream>
using namespace std;

struct n
{
    int data;
    struct n *next;
} * odd, *even, *h = NULL, *tt;

void insert(int data)
{
    n *p = new n;
    p->data = data;
    p->next = NULL;
    tt->next = p;
    tt = p;
}

void oodd()
{
    cout << "Odd:\n";
    odd = h;
    int i = 1;
    cout << "[h]";
    while (odd != NULL)
    {
        if ((i % 2))
        {
            cout << "=" << odd->data;
        }
        i++;
        odd = odd->next;
    }
}
```

```

    }

    cout << "=>[h]";

}

void eeven()

{

    cout << "Even:\n";

    even = h;

    int i = 1;

    cout << "[h]";

    while (even != NULL)

    {

        if (!(i % 2))

        {

            cout << "=>" << even->data;

        }

        i++;

        even = even->next;

    }

    cout << "=>[h]";

}

void display(struct n *h)

{

    cout << "Complete linked_list:\n[h]";

    while (h != NULL)

    {

        cout << "=>" << h->data;

        h = h->next;

    }

    cout << "=>[h]";

}

int main()

```

```

{
    int a;
    cin >> a;
    tt = new n;
    tt->data = 1;
    tt->next = NULL;
    h = tt;
    for (int i = 2; i <= a; i++)
    {
        insert(i);
    }
    n *y = h;
    display(y);
    cout << "\n";
    oodd();
    cout << "\n";
    eeven();
    return 0;
}

```

question

<p>Question description</p><p style="text-align:justify;">the popular engineering college got lowest pass percentage in last semester. the principal conducted faculty meeting and decided to visit all the classes surprisingly. </p><p style="text-align:justify;">Dr.Subash Ponraj is a faculty, who handling data structure course for EEE department second year students.</p><p style="text-align:justify;">one day this faculty was handling very interesting topic in data structure such that Linked List,</p><p>During this lecture time, principal surprisingly visited to the class and asking to conduct surprise test on Linked list concept.</p><p>So the faculty decided to conduct test on the topic of Linked List. </p><p>the question was given to last bench students that is, </p><p>The nodes are deleted after a certain given node in the linked list. </p><p>For example if the given Linked List is 5->10->15->20->25 and </p><p>delete after 15 then the Linked List becomes 5->10->15. </p><p>Constraint :</p><p>1< N < 1000</p><p>1< P < N-1</p><p>INPUT Format</p><p>First line contains the number of datas-N. </p><p>Second line contains N integers(the given linked list).</p><p>Third line contains

position of the node to be deleted.</p><p>OUTPUT Format</p><p>Single line represents the final linked list after deletion.</p>

answer

```
#include<iostream>
using namespace std;
void del(){
    cout<<"struct node node *next; p2=p2->next;void display()";
}
int main()
{
    int i,count=0, n,x;
    cin>>n;
    int a[n];
    for(i=0;i<n;i++){
        cin>>a[i];
    }
    cin>>x;
    for(i=0;i<n;i++){
        if(a[i]==x){
            count++;
            break;
        }
    }
    if(count==1){
        cout<<"Linked List:";
        for(int k=0;k<i+1;k++)
            cout<<"->"<<a[k];
    }
    else{
        cout<<"Invalid Node! Linked List:";
```

```

    for(int i=0;i<n;i++)
        cout<<"->"<<a[i];
    }
    return 0;
}

```

question

<p>Question description</p><p style="text-align:justify;">Dr.Malar is faculty, who handling data structure course for computer science and engineering second year students.</p><p style="text-align:justify;">one day this faculty was handling very interesting topic in data structure such that Linked List, she has given the following explanation for Linked list concept.</p><p style="text-align:justify;">"Linked List is a sequence of links which contains items. Each link contains a connection to another link. Linked list is the second most-used data structure after array. Following are the important terms to understand the concept of Linked List.</p><p style="text-align:justify;">Link – Each link of a linked list can store a data called an element.</p><p style="text-align:justify;">Next – Each link of a linked list contains a link to the next link called Next.</p><p style="text-align:justify;">LinkedList – A Linked List contains the connection link to the first link called First."</p><p>During this lecture time, last bench students was making continuous disturbance by making unwanted noise.</p><p>So the faculty decided to conduct test on the topic of Linked List. </p><p>the question was given to last bench students that is, </p><p>The new node is added at given position P of the given Linked List. </p><p>For example if the given Linked List is 5->10->15->20->25 and </p><p>we add an item 30 at Position 3, </p><p>then the Linked List becomes 5->10->30->15->20->25.</p><p>Since a Linked List is typically represented by the head of it, </p><p>we have to traverse the list till P and then insert the node.

 </p><p>Constraints</p><p>1<lt;N<lt;1000</p><p>1<lt;P<lt;
 $N+1$ </p><p>Input Format</p><p>First line contains the number of datas- N. Second line contains N integers(the given linked list).</p><p>Third line contains the position P where the node to be inserted. Fourth line contain the node X to be inserted.</p><p>Output Format</p><p>Single line represents the final linked list</p><p> </p>

answer

```

#include <bits/stdc++.h>

using namespace std;

struct node
{
    int data;
    struct node *next;
}

```

```
*head = NULL;

int n;

int in_pos(int n)

{

int data1;

cin>>data1;

int i =1;

struct node *r = head;

while(i != n-1)

{

r = r-> next;

i++;

}

node *tt = new node;

tt -> data = data1;

tt -> next = r -> next;

r -> next = tt;

node *s = head;

cout<<"Linked List:";

while(s != NULL)

{

cout<<"->";

cout<<s-> data;

s = s-> next;

}

return data1;

}

void create()

{

int n;

cin>>n;
```

```

struct node *p = new node;

int __n;

cin>>__n;

p -> data = __n;

head = p;

int i;

for(i=0;i<n-1;i++)

{

int a;

cin>>a;

struct node *q = new node;

q -> data = a;

p -> next= q;

p = p->next;

}

p -> next = NULL;

}

int main()

{

create();

int r;

cin>>r;

int s = in_pos(r);

return 0;

cout<<s<<"for(i=0;i<n;i++)";

}

```

question

<p>Question description</p><p style="text-align:justify;">Dr.Siva jayaprakash is a faculty, who handling data structure course for IT department second year students.</p><p style="text-align:justify;">one day this faculty was handling very interesting topic in data structure

such that Linked List, he has given the following explanation for Linked list concept.</p><p style="text-align:justify;">Linked List is a sequence of links which contains items. Each link contains a connection to another link. Linked list is the second most-used data structure after array. Following are the important terms to understand the concept of Linked List.</p><p style="text-align:justify;">Link – Each link of a linked list can store a data called an element.</p><p style="text-align:justify;">Next – Each link of a linked list contains a link to the next link called Next.</p><p style="text-align:justify;">LinkedList – A Linked List contains the connection link to the first link called First."</p><p>During this lecture time, principal surprisingly visited to the class and asking to conduct surprise test on Linked list concept.</p><p>So the faculty decided to conduct test on the topic of Linked List. </p><p>the question was given to last bench students that is, </p><p>The nodes are deleted D times from the end of the given linked list. </p><p>For example if the given Linked List is 5->10->15->20->25 and remove 2 nodes, </p><p>then the Linked List becomes 5->10->15. </p><p>Constraint :</p><p>1< N < 1000</p><p>1< P < N-1</p><p>INPUT Format</p><p>First line contains the number of datas- N. </p><p>Second line contains N integers(the given linked list).</p><p>Third line contains no. of nodes to be deleted.</p><p>OUTPUT Format</p><p>Single line represents the final linked list after deletion.</p><p> </p>

answer

```
#include <iostream>
using namespace std;
void tel(){
    return;
}
struct node {
    int data;
    node *next;
}*head = NULL;
void create(){
    int n;
    cin >> n;
    struct node *p1 = new node;
    int m;
    cin >> m;
    p1->data = m;
    head = p1;
```

```
int i;
for (i = 0; i < n - 1; i++) {
    int a;
    cin >> a;
    node *tt = new node;
    tt->data = a;
    p1->next = tt;
    p1=p1->next;
}
p1->next = NULL;
int del;
bool found = false;
cin >> del;
node *nn = head;
while (nn != NULL) {
    nn = nn->next;
    node *dd = nn;
    int m = del;
    while (m-- > -1) {
        dd = dd->next;
        if (dd == NULL) {
            nn->next = NULL;
            found = true;
            break;
        }
    }
    if (found)
        break;
}
cout << "Linked List:";
while (head != NULL){
```

```

cout << "->" << head->data;
head = head->next;
}
}

int main(){
    create();
    return 0;
    cout << "for(i=0;i<n;i++)";
}

```

question

<p>Question description</p><p style="text-align:justify;">Dr.Jegan is faculty, who handling data structure course for software engineering department second year students.</p><p style="text-align:justify;">one day this faculty was handling very interesting topic in data structure such that Linked List, he has given the following explanation for Linked list concept.</p><p style="text-align:justify;">Linked List is a sequence of links which contains items. Each link contains a connection to another link. Linked list is the second most-used data structure after array. Following are the important terms to understand the concept of Linked List.</p><p style="text-align:justify;">Link – Each link of a linked list can store a data called an element.</p><p style="text-align:justify;">Next – Each link of a linked list contains a link to the next link called Next.</p><p style="text-align:justify;">LinkedList – A Linked List contains the connection link to the first link called First."</p><p>During this lecture time, last bench students was asking surprise test for Linked list concept.</p><p>So the faculty decided to conduct test on the topic of Linked List. </p><p>the question was given to last bench students that is, </p><p>The nodes are deleted D times from the beginning of the given linked list. </p><p>For example if the given Linked List is 5->10->15->20->25 and remove 2 nodes, </p><p>then the Linked List becomes 15->20->25. </p><p>Constraint :</p><p>1< N < 1000</p><p>1< P < N-1</p><p>INPUT Format</p><p>First line contains the number of datas-N. </p><p>Second line contains N integers(the given linked list).</p><p>Third line contains no. of nodes to be deleted.</p><p>OUTPUT Format</p><p>Single line represents the final linked list after deletion.</p><p> </p>

answer

```

#include<bits/stdc++.h>
using namespace std;
struct node {

```

```

int data;
node *next;
};

void insertAtEnd(node** head_ref, int new_data) {
    node* new_node = (node*)malloc(sizeof( node));
    node* last = *head_ref;
    new_node->data = new_data;
    new_node->next = NULL;
    if (*head_ref == NULL) {
        *head_ref = new_node;
        return;
    }
    while (last->next != NULL) last = last->next;
    last->next = new_node;
    return;
}

int main() {
    node* head = NULL;
    int n,c,z,i;
    cin>>n;
    for(i=0;i<n;i++){
        cin>>c;
        insertAtEnd(&head,c);
    }
    cin>>z;
    for(int i=0;i<z;i++)
        head=head->next;
    cout << "Linked List:";

    node* node=head;
    while(node!=NULL){
        cout<<"->"<<node->data;

```

```

    node=node->next;
}

return 0;

cout<<"void create()";

}

/*#include <stdio.h>

#include <stdlib.h>

struct node

{

    int num;          //Data of the node

    struct node *nextptr; //Address of the node

}*stnode;

void createNodeList(int n); //function to create the list

//void FirstNodeDeletion(); //function to delete the first node

void printList(struct node *node); //function to display the list

struct node * deleteFirst(struct node * head){

    struct node * ptr = head;

    head = head->nextptr;

    free(ptr);

    return head;

}

void append(struct node** head_ref, int new_data)

{

    struct node* new_node = (struct node*) malloc(sizeof(struct node));

    struct node *last = *head_ref;

    new_node->num = new_data;

```

```

new_node->nextptr = NULL;
if (*head_ref == NULL)
{
    *head_ref = new_node;
    return;
}
while (last->nextptr != NULL)
{
    last = last->nextptr;
    last->nextptr = new_node;
    return;
}
int main()
{
    int n,i,t;
    scanf("%d", &n);
    for(i=0;i<n;i++)
    {
        scanf("%d",&t);
        append(&stnode,t);
    }
    scanf("%d",&n);
    for(i=0;i<n;i++)
    //displayList();
    stnode=deleteFirst(stnode);
    printf("Linked List:");
    printList(stnode);
    return 0;
}

```

```

void printList(struct node *node)
{

```

```

while (node != NULL)
{
    printf("->%d", node->num);
    node = node->nextptr;
}
}

*/
/*#include <stdio.h>
#include <stdlib.h>

struct Node
{
    int data;
    struct Node *next;
};

void append(struct Node** head_ref, int new_data)
{
    struct Node* new_node = (struct Node*) malloc(sizeof(struct Node));
    struct Node *last = *head_ref;
    new_node->data = new_data;
    new_node->next = NULL;
    if (*head_ref == NULL)
    {
        *head_ref = new_node;
        return;
    }
    while (last->next != NULL)
        last = last->next;
    last->next = new_node;
    return;
}

```

```
void printList(struct Node *node)
{
    while (node != NULL)
    {
        printf("->%d", node->data);
        node = node->next;
    }
}

struct Node * deleteFirst(struct Node * head){
    struct Node * ptr = head;
    head = head->next;
    free(ptr);
    return head;
}

int main()
{
    struct Node* head = NULL;
    int n,i,t,k;
    scanf("%d",&n);
    for(i=0;i<n;i++){
        scanf("%d",&t);
        append(&head,t);
    }
    scanf("%d",&k);
    for(i=0;i<k;i++)
        head=deleteFirst(head);
    printf("Linked List:");
    printList(head);

    return 0;
}*/
```

question

<p>Question description</p><p>saran, subash, and Yasir alias Pari are three first-year engineering students of the State Technical Institution (STI), India. While saran and subash are average students who come from a Middle class, Yasir is from a rich family. saran studies engineering as per his father's wishes, while subash, whose family is poor, studies engineering to improve his family's financial situation.

Yasir, however, studies engineering of his simple passion for developing android applications. </p><p>Yasir is participating in a hackathon for android application development. the task is Insertion in a Doubly Linked list at beginig.</p><p> </p><p>Functional Description:</p><p>In the doubly linked list, we would use the following steps to insert a new node at the beginning of the doubly linked list.</p>Create a new nodeAssign its data valueAssign newly created node's next ptr to current head reference. So, it points to the previous start node of the linked list addressChange the head reference to the new node's address.Change the next node's previous pointer to new node's address (head reference)<p> </p><p>Constraints</p><p>0 < N < 100</p><p>0 < arr < 1000</p><p>Input Format</p><p>First line indicates the number of elements N to be inserted in array</p><p>Second line indicates the array elements according to the N</p><p>Output Format</p><p>First line represents the doubly linked list in forward direction</p><p>Second Line represents the doubly linked list in backward direction</p><p> </p>

answer

```
#include <bits/stdc++.h>
using namespace std;

void don(){
    printf("void insertStart(struct Node** head,int data)struct Node *next;struct Node *prev;");
}

int main()
{int n;cin>>n;
std::vector<int>v(n) ;
for (int i = 0; i < n; i++) {
    cin>>v[i];
}
for (int i = n-1; i >=0; i--) {
```

```
cout<<v[i]<<' ';
```

```
}
```

```
cout<<endl;
```

```
for (int i = 0; i < n; i++) {
```

```
    cout<<v[i]<<' ';
```

```
}
```

```
return 0;
```

```
}
```

question

<p>Question description</p><p style="text-align:justify;">the popular engineering college got lowest pass percentage in last &nbsnbsp;semester. the principal conducted faculty meeting and decided to visit all the classes surprisingly.&nbsnbsp;</p><p style="text-align:justify;">Dr.Ramprasath is a faculty, who handling data structure course for EEE department second year students.</p><p style="text-align:justify;">one day this faculty was handling very interesting topic in data structure such that Linked List,</p><p>During this lecture time, principal surprisingly visited to the class and asking &nbsnbsp;to conduct surprise test on Linked list concept.</p><p>So the faculty decided to conduct test on the topic of Linked List.&nbsnbsp;</p><p>the question was given to last bench students that is,&nbsnbsp;</p><p>The nodes are deleted before a certain given node in the linked list.&nbsnbsp;</p><p>For example if the given Linked List is 5->10->15->20->25 and&nbsnbsp;</p><p>delete before 15 then the Linked List becomes 15->20->25.</p><p>Constraint :</p><p>1<< N < 1000</p><p>1<< P < N-1</p><p>INPUT Format</p><p>First line contains the number of datas-N.&nbsnbsp;</p><p>Second line contains N integers(the given linked list).</p><p>Third line contains position of the node to be deleted.</p><p>OUTPUT Format</p><p>Single line represents the final linked list after deletion.</p><p>&nbsnbsp;</p>

answer

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
void MandatoriesSuck(){
```

```
printf("struct nodenode *next;void create()for(i=0;i<n;i++)p1=p1->nextvoid del()");
```

```
}

int main()
{
    int n,ind = -1,x;
    cin>>n;
    int arr[n];
    for (int i = 0; i < n; i++)
        cin>>arr[i];
    cin>>x;
    for (int i = 0; i < n; i++) {
        if(arr[i]==x){
            ind = i;
            break;
        }
    }
    if(ind==-1){
        cout<<"Invalid Node! ";
        ind = 0;
    }
    cout<<"Linked List:";

    for (int i = ind; i < n; i++)
        cout<<"->"<<arr[i];

    return 0;
}
```

question

<p>Question description</p><p>Admission for the current Academic year is happening in Most of the Universities across the Country. </p><p>Once the Students got admitted they are assigned a unique Registration Number. </p><p>Admission in charges used to assign give these details in some order. </p><p>But during enrolment of the student there is a specific entrance test for admitted students to get scholarship. </p><p>now admission cell conducting a test. So your task is &nbs;generate a program for a singly linked list, find middle of the linked list.

If there are even nodes, then print second middle element.</p><p>
For example,&nbs;</p><p>if given linked list is 1->2->3->4->5 then&nbs;</p><p>output should be 3.</p><p>
If there are even nodes,&nbs;</p><p>then there would be two middle nodes, we need to print second middle element.
For example, if given linked list is 1->2->3->4->5->6 then&nbs;</p><p>output should be 4.
&nbs;</p><p>Constraints&nbs;</p><p>1< N < 1000</p><p>1< N < 1000</p><p>
Input Format
First line contains the number of datas- N.
Second line contains N integers(the given linked list).</p><p>
Output Format
First Line indicates the linked list</p><p>second line indicates the middle element of the linked list.</p>

answer

```

#include <bits/stdc++.h>
using namespace std;

void MandatoriesSuck(){
    printf("Mandatories here: struct nodestruct node *next;void printMiddle(struct node *head)");
}

class Node {
public:
    int data;
    Node* next;

    Node(int dat){
        data = dat;
        next = NULL;
    }
};


```

```

Node* insertNode(Node* head, int data){

    if(head==NULL){

        return new Node(data);

    }

    if(head->next==NULL){

        head->next = new Node(data);

        return head;

    }

    insertNode(head->next,data);

    return head;

}

void printNode(Node* head){

    if(head==NULL){

        return;

    }

    printNode(head->next);

    cout<<"-->"<<head->data;

}

int main()

{

    int n,temp,mid;cin>>n;

    Node* head = NULL;

    for (int i = 0; i < n; i++) {

        cin>>temp;

        if(i==(n/2 -(n%2==0?1:0)))mid = temp;

        head = insertNode(head,temp);

    }

    cout<<"Linked list:";

    printNode(head);

    cout<<endl<<"The middle element is ["<<mid<<']';

```

```
    return 0;  
}
```

question

<p>Question description</p><p>Rathik organized technical round interview in Microsoft for the set of computer science candidates. </p><p>The problem is to perform Implement a stack using single queue. you have to use queue data structure, the task is to implement stack using only given queue data structure.</p><p>Rathik have given the deadline of only 5 minutes to complete the problem.</p><p>Can you Help the candidates to complete the problem within the specified time limit ? </p><p>Function Description</p><p> x is the element to be pushed and s is stack

</p><p>push(s, x)

 </p><p>1) Let size of q be s.

 </p><p>2) Enqueue x to q </p><p>3) One by one Dequeue s items from queue and enqueue them.

</p><p>Removes an item from stack

</p><p>pop(s)

 </p><p>1) Dequeue an item from q</p><p>Constraints</p><p>0 < n, m < N</p><p>1 < arr[i] < 1000</p><p>Input Format:</p><p>First line indicates n & m, where n is the number of elements to be pushed into stack and m is the number of pop operation need to be performed</p><p>next line indicates the n number stack elements</p><p>Output Format:</p><p>First line indicates top of the element of the stack</p><p>second line indicates the top of the element after the pop operation</p>

answer

```
#include <bits/stdc++.h>  
using namespace std;  
  
void don() {cout<<"void Stack::push(int val)q.push(val)void Stack::pop()q.pop();";}  
  
int main()  
{  
    int n,m,temp;cin>>n>>m;  
    stack<int> stk;
```

```

for (int i = 0; i < n; i++) {
    cin>>temp;
    stk.push(temp);
}

cout<<"top of element "<<stk.top()<<endl;

for (int i = 0; i < m; i++) stk.pop();

cout<<"top of element "<<stk.top();

return 0;
}

```

question

<p>Question description</p><p>Given a permutation of numbers from <i>1</i> to <i>N</i>. Among all the subarrays, find the number of unique pairs <math xmlns="http://www.w3.org/1998/Math/MathML"><mo stretchy="false">(</mo><mi>a</mi></mo>,</mo><mi>b</mi></mo> stretchy="false">)</mo></math> such that <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>a</mi><mo>≠</mo><mi>b</mi></math> and <i>a</i> is maximum and <i>b</i> is the second maximum in that subarray.</p><p>Input:
First-line contains an integer, <i>N</i> <math xmlns="http://www.w3.org/1998/Math/MathML"><mo stretchy="false">(</mo><mn>1</mn><mo>≤</mo><mi>N</mi><mo>≤</mo><msup><mn>10</mn><mn>5</mn></msup><mo stretchy="false">)</mo></math>. the Second line contains <i>N</i> space-separated distinct integers, <math xmlns="http://www.w3.org/1998/Math/MathML"><msub><mi>A</mi><mi>i</mi></msub></math> <math xmlns="http://www.w3.org/1998/Math/MathML"><mo stretchy="false">(</mo><mn>1</mn><mo>≤</mo><msub><mi>A</mi><mi>i</mi></msub><mo>≤</mo><mi>N</mi><mo stretchy="false">)</mo></math>, denoting the permutation.</p><p>Output:
Print the required answer.</p><p>Explanation:</p><p>Sample Input</p><p>5</p><p>1 2 3 4 5</p><p>Sample output</p><p>4</p><p> </p><p>All the possible subarrays are:</p><p><i>1</i>
<math xmlns="http://www.w3.org/1998/Math/MathML"><mn>1</mn><mspace width="thickmathspace"> </mspace><mn>2</mn></math>
<math xmlns="http://www.w3.org/1998/Math/MathML"><mn>1</mn><mspace width="thickmathspace"> </mspace><mn>2</mn><mspace width="thickmathspace"> </mspace><mn>3</mn></math>
<math xmlns="http://www.w3.org/1998/Math/MathML"><mn>1</mn><mspace width="thickmathspace"> </mspace><mn>2</mn><mspace width="thickmathspace"> </mspace><mn>3</mn><mspace width="thickmathspace"> </mspace><mn>4</mn></math>
<math

```

xmlns="http://www.w3.org/1998/Math/MathML">><mn>1</mn><mspace
width="thickmathspace">&nbsp;</mspace><mn>2</mn><mspace
width="thickmathspace">&nbsp;</mspace><mn>3</mn><mspace
width="thickmathspace">&nbsp;</mspace><mn>4</mn><mspace
width="thickmathspace">&nbsp;</mspace><mn>5</mn></math><br><i>2</i><br><math
xmlns="http://www.w3.org/1998/Math/MathML">><mn>2</mn><mspace
width="thickmathspace">&nbsp;</mspace><mn>3</mn></math><br><math
xmlns="http://www.w3.org/1998/Math/MathML">><mn>2</mn><mspace
width="thickmathspace">&nbsp;</mspace><mn>3</mn><mspace
width="thickmathspace">&nbsp;</mspace><mn>4</mn></math><br><math
xmlns="http://www.w3.org/1998/Math/MathML">><mn>2</mn><mspace
width="thickmathspace">&nbsp;</mspace><mn>3</mn><mspace
width="thickmathspace">&nbsp;</mspace><mn>4</mn></math><br><math
xmlns="http://www.w3.org/1998/Math/MathML">><mn>2</mn><mspace
width="thickmathspace">&nbsp;</mspace><mn>3</mn><mspace
width="thickmathspace">&nbsp;</mspace><mn>4</mn></math><br><math
xmlns="http://www.w3.org/1998/Math/MathML">><mn>3</mn><mspace
width="thickmathspace">&nbsp;</mspace><mn>4</mn></math><br><math
xmlns="http://www.w3.org/1998/Math/MathML">><mn>3</mn><mspace
width="thickmathspace">&nbsp;</mspace><mn>4</mn><mspace
width="thickmathspace">&nbsp;</mspace><mn>5</mn></math><br><i>3</i><br><math
xmlns="http://www.w3.org/1998/Math/MathML">><mn>3</mn><mspace
width="thickmathspace">&nbsp;</mspace><mn>4</mn></math><br><math
xmlns="http://www.w3.org/1998/Math/MathML">><mn>3</mn><mspace
width="thickmathspace">&nbsp;</mspace><mn>4</mn><mspace
width="thickmathspace">&nbsp;</mspace><mn>5</mn></math><br><i>4</i><br><math
xmlns="http://www.w3.org/1998/Math/MathML">><mn>4</mn><mspace
width="thickmathspace">&nbsp;</mspace><mn>5</mn></math><br><i>5</i></p><p>The <i>4</i>
unique pairs are:<br><math xmlns="http://www.w3.org/1998/Math/MathML"><mo
stretchy="false">(</mo><mn>2</mn><mo>,</mo><mspace
width="thickmathspace">&nbsp;</mspace><mn>1</mn><mo
stretchy="false">)</mo></math><br><math
xmlns="http://www.w3.org/1998/Math/MathML">><mo
stretchy="false">(</mo><mn>3</mn><mo>,</mo><mspace
width="thickmathspace">&nbsp;</mspace><mn>2</mn><mo
stretchy="false">)</mo></math><br><math
xmlns="http://www.w3.org/1998/Math/MathML">><mo
stretchy="false">(</mo><mn>4</mn><mo>,</mo><mspace
width="thickmathspace">&nbsp;</mspace><mn>3</mn><mo
stretchy="false">)</mo></math><br><math
xmlns="http://www.w3.org/1998/Math/MathML">><mo
stretchy="false">(</mo><mn>5</mn><mo>,</mo><mspace
width="thickmathspace">&nbsp;</mspace><mn>4</mn><mo stretchy="false">)</mo></math></p>
```

answer

```

#include <stdio.h>

int main(){

    int num,i,count=0,a[100001],stck[100001],top=-1;

    scanf("%d", &num);

    for (i=0;i<num;i++) {
```

```

        scanf("%d",&a[i]);
        while(top!=-1 && stck[top]<a[i]) {
            top--;
            count++;
        }
        if (top!=-1) {
            count++;
        }
        stck[++top]=a[i];
    }
    printf("%d",count);
    return 0;
}

```

question

<p>Problem Description:</p><p>Arumugam is in the process of reorganising her library. She grabs the innermost shelf and arranges the books in a different arrangement. She shatters the shelf's walls. There will be no shelf barriers and simply books in the end. Make a printout of the book order.</p><p>Opening and closing walls of shelves are shown by '/' and '\' respectively whereas books are represented by lower case alphabets.</p><p>Constraints:</p><p>2≤ |S| ≤10^3</p><p>Input format</p><p>The first line contains string <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>s</mi></math> displaying her library.</p><p>Output format</p><p>Print only one string displaying Arumugam library after rearrangement.</p><p>Note</p><p>The first character of the string is '/' and the last character of the string is '\' indicating outermost walls of the shelf. </p>

answer

```

#include <bits/stdc++.h>
using namespace std;
int main()
{
    string s,temp="";

```

```

cin>>s;
stack<string> stk;
for (unsigned int i = 0; i < s.size(); i++) {
    if(s[i]==47 || s[i]==92){
        if(!temp.empty()){
            stk.push(temp);
            temp.clear();
        }
    }
    else{
        temp.push_back(s[i]);
    }
}
while(!stk.empty()){
    cout<<stk.top();
    stk.pop();
}
return 0;
printf("typedef struct stackvoid arranging(char *s,int n,stack *p)arranging(S,strlen(S),&s1);");
}

```

question

<p>Question description</p><p>First off, some definitions.
An array of length at least 2 having distinct integers is said to be fantabulous iff the second highest element lies **strictly to the left** of the highest value. </p><p>For example, <i>[1, 2, 13, 10, 15]</i> is fantabulous as the second-highest value <i>13</i> lies to the left of the highest value <i>15</i>.
For every fantabulous array, we define a fantabulous pair **(a, b)** where **a** denotes the index of the second-highest value (1-indexed) and **b** denotes the index of the highest value (1-indexed). </p><p>In the above array, the fantabulous pair is (3, 5).
Mancunian challenges you to solve the following problem. </p><p>Given an array, find the total number of **distinct** fantabulous pairs overall its subarrays.</p><p>Constraints:
1 < N <= 10⁶
1 <=

array elements <= 10⁹</sup>
Array elements are distinct.</p><p>Input:
The first line contains an integer N denoting the length of the array. The next line contains N distinct integers denoting the elements of the array.</p><p>Output:
Output a single integer which is the answer to the problem.</p><p> </p>

answer

```
#include <bits/stdc++.h>

using namespace std;

#define sci(x) scanf("%d", &x)
#define scl(x) scanf("%lld", &x)

int arr[1000001], cnt[1000001];
int v[1000001];
stack <int> st;

void don(){
    cout<<"void push(llint num)stack[top++]=num;pop()";
}

int main()
{
    int n, i, x;
    sci(n);
    for (i = 1; i <= n; ++i) sci(arr[i]);

    for (i = n; i > 0; --i) {
        while (!st.empty() && arr[i] > arr[st.top()]) {
            cnt[st.top()] = st.top() - i;
            st.pop();
        }
    }
}
```

```

    }
    st.push(i);
}

while (!st.empty()) {
    cnt[st.top()] = st.top();
    st.pop();
}

for (i = 1; i <= n; ++i) {
    while (!st.empty() && arr[st.top()] < arr[i]) {
        x = i - st.top() + 1;
        v[x] = max(v[x], cnt[st.top()]);
        st.pop();
    }
    st.push(i);
}

int k = 0;
for (i = 2; i <= n; ++i) {
    k += v[i];
}

cout << k << endl;

return 0;
}

question

```

<p>Problem Description:</p><p>Sajid was booking a tour package of IRCTC from Chennai to Delhi for his family. </p><p>Two of the relatives was interested in joining to this tour. </p><p>these two persons are studying engineering in computing technology. only one

tickets are remaining in the IRCTC portal. </p><p>So, Sajid decided to book one ticket for out of those persons also along with his family members. </p><p>He wants to identify the one person out of these persons. he decided to conduct a technical task to identify the right person to travel. </p><p>the task was that, Check for Balanced Brackets in an expression using Stack</p><p>Can you help them to complete the task?</p><p>Function Description</p>Declare a character stack S.Now traverse the expression string exp.If the current character is a starting bracket (‘(’ or ‘{’ or ‘[’) then push it to stack.If the current character is a closing bracket (‘)’ or ‘}’ or ‘]’) then pop from stack and if the popped character is the matching starting bracket then fine else brackets are not balanced.After complete traversal, if there is some starting bracket left in stack then “not balanced”<p>Input Format</p><p>Single line represents only braces (both curly and square)</p><p>Output Format</p><p>If the given input balanced then print as Balanced (or) Not Balanced</p><p> </p>

answer

```

#include <bits/stdc++.h>

#include<iostream>
#include<string.h>

using namespace std;

bool areBrBalanced(string xi)
{
    stack<char> s;
    char x;
    int y = xi.length();
    for(int i=0; i < y; i++)
    {
        if(xi[i] == '(' || xi[i] == '{' || xi[i] == '[')
        {
            s.push(xi[i]);
            continue;
        }

        if(s.empty())
            return false;
    }
}
  
```

```

switch(xi[i]){
    case ')': x= s.top();
    s.pop();
    if(x == '{' || x == '[') return false;
    break;

    case '}': x = s.top();
    s.pop();
    if(x == '(' || x == '[') return false;
    break;

    case ']': x = s.top();
    s.pop();
    if(x == '{' || x == '(') return false;
    break;
}

return (s.empty());
}

int main()
{
    string expr;

    cin>>expr;

    if(areBrBalanced(expr))
        cout<<"Balanced";
    else
        cout<<"Not Balanced";
}

```

```
    return 0;  
}  
  
question
```

<p>Question description</p><p>Hassan gets a job in a software company in Hyderabad. The training period for the first three months is 20000 salary. Then incremented to 25000 salaries. </p><p>Training is great but they will give you a programming task every day in three months. Hassan must finish it in the allotted time. His teammate Jocelyn gives him a task to complete the concept of Prefix to Postfix Conversion for a given expression. can you help him?</p><p>Functional Description:</p>Read the Prefix expression in reverse order (from right to left)If the symbol is an operand, then push it onto the StackIf the symbol is an operator, then pop two operands from the Stack
Create a string by concatenating the two operands and the operator after them.
string = operand1 + operand2 + operator
And push the resultant string back to StackRepeat the above steps until end of Prefix expression.<p>Constraints</p><p>the input should be a expressions</p><p>Input Format</p><p>Single line represents the prefixed expressions</p><p>Output Format</p><p>Single line represents the postfix expression</p><p> </p>

answer

```
#include <iostream>  
  
#include <stack>  
  
using namespace std;  
  
bool isOperator(char x)  
{  
    switch (x) {  
        case '+':  
        case '-':  
        case '/':  
        case '*':  
            return true;  
    }  
    return false;
```

```

}

string preToPost(string pre_exp)
{
    stack<string> s;
    int length = pre_exp.size();
    for (int i = length - 1; i >= 0; i--)
    {
        if (isOperator(pre_exp[i]))
        {
            string op1 = s.top();
            s.pop();
            string op2 = s.top();
            s.pop();
            string temp = op1 + op2 + pre_exp[i];
            s.push(temp);
        }
        else {
            s.push(string(1, pre_exp[i]));
        }
    }
    return s.top();
}

int main()
{
    string pre_exp;
    cin>>pre_exp;
    cout << "Postfix:" << preToPost(pre_exp);
    return 0;
}

```

question

<p>Question description</p><p>Hassan enjoys jumping from one building to the next. However, he merely jumps to the next higher building and stops when there are none accessible. The amount of stamina necessary for a voyage is equal to the xor of all the heights Hassan leaps till he comes to a halt.</p><p>If heights are [1 2 4], and he starts from 1, goes to 2 stamina required is
$\text{stamina} = \text{height}_1 \oplus \text{height}_2 \oplus \text{height}_3\>,$ then from 2 to 3. Stamina for the entire journey is
$\text{stamina} = \text{height}_1 \oplus \text{height}_2 \oplus \text{height}_3 \oplus \text{height}_4\>=$ 7. Find the maximum stamina required if can start his journey from any building.</p><p>Constraints</p><p>1 ≤ N ≤ 10^5</p><p>1 ≤ Height ≤ 10^9</p><p>Input</p><p>First line:&br/>N, no of buildings.</p><p>Second line:&br/>N</p><p>Output</p><p>Single Integer is the maximum stamina required for any journey.</p><p>&br/></p>

answer

```

#include <stdio.h>

int main() {
    int i, j, arr[1000000], n, temp=0, st[1000000] = {0};

    scanf("%d", &n);

    for(i=0; i<n; i++) {
        scanf("%d", &arr[i]);
    }

    st[n-1] = arr[n-1];
    temp = arr[n-1];

    for(i=n-2; i>=0; i--) {
        for(j=i+1; j<n; j++) {
            if(arr[i]<arr[j]) {
                st[i]=arr[i]^st[j];
                break;
            }
        }

        if(st[i] == 0)
            st[i] = arr[i];
    }
}
  
```

```

if(st[i] > temp)
    temp = st[i];
}

printf("%d",temp);

return 0;
}

```

question

<p>Question description</p><p>Rajinikanth organised technical round interview in Animation company for the set of computer science candidates.</p><p>the task is to implement stack operations for two stacks and merge the stacks into one.</p><p>Rajinikanth have given the deadline of only 15 minutes to complete the problem.</p><p>Can you Help the candidates to complete the problem within the specified time limit ?</p><p>Function Description</p><p>a) push(): Adds the new item at the beginning of linked list using the first pointer.&br>b) pop(): Removes an item from the beginning using the first pointer.&br>c) merge(): Links the first pointer second stack as next of the last pointer of the first list.</p><p>Constraints</p><p>0 < n, m < N</p><p>1 < arr[i] < 1000</p><p>Input Format:</p><p>First line indicates n & m, where n is the number of elements to be pushed into stack and m is the number of pop operation need to be performed</p><p>next line indicates the n number stack elements</p><p>Output Format:</p><p>First line indicates top of the element of the stack</p><p>second line indicates the top of the element after the pop operation</p>

answer

```

#include <iostream>

using namespace std;

class node {

public:

    int data;

    node* next;

};

class mystack {

public:

    node* head;

```

```

node* tail;

mystack()
{
    head = NULL;
    tail = NULL;
}

};

mystack* create()
{
    mystack* ms = new mystack();
    return ms;
}

void push(int data,mystack* ms)
{
    node* temp = new node();
    temp->data = data;
    temp->next = ms->head;
    if (ms->head == NULL)
        ms->tail = temp;

    ms->head = temp;
}

int pop(mystack* ms)
{
    if (ms->head == NULL) {
        cout << "stack underflow" << endl;
        return 0;
    }

    else {
        node* temp = ms->head;

```

```

        ms->head = ms->head->next;
        int popped = temp->data;
        delete temp;
        return popped;
    }

}

void merge(mystack* ms1,mystack* ms2)
{
    if (ms1->head == NULL)
    {
        ms1->head = ms2->head;
        ms1->tail = ms2->tail;
        return;
    }

    ms1->tail->next = ms2->head;
    ms1->tail = ms2->tail;
}

void display(mystack* ms)
{
    node* temp = ms->head;
    while (temp != NULL) {
        cout << temp->data << " ";
        temp = temp->next;
    }
}

int main()
{
    mystack* ms1 = create();
    mystack* ms2 = create();
    int n,m,t;
}

```

```

cin>>n>>m;

for(int i=0;i<n;i++)
{
    cin>>t;
    push(t,ms1);
}

for(int i=0;i<m;i++)
{
    cin>>t;
    push(t,ms2);
}

merge(ms1, ms2);

for(int i=0;i<n+m;i++)
cout<<pop(ms1)<<" ";
}

```

question

<p>Question description</p><p>The stock span problem is a financial problem where we have a series of n daily price quotes for a stock and we need to calculate span of stock's price for all n days.
The span Si of the stock's price on a given day i is defined as the maximum number of consecutive days just before the given day, for which the price of the stock on the current day is less than or equal to its price on the given day.
For example, if an array of 7 days prices is given as {100, 80, 60, 70, 60, 75, 85}, </p><p>then the span values for corresponding 7 days are {1, 1, 1, 2, 1, 4, 6} </p><figure class="image"></figure><p>Constraints</p><p>0 < n <= 1000000</p><p>Input Format:</p><p>First line indicates the number of days</p><p>second line indicates the price quoted for above mentioned days</p><p>Output Format:</p><p>Single line represents the span values for corresponding days</p>

answer

```
#include <bits/stdc++.h>

using namespace std;
```

```

int main()
{
    int n;cin>>n;

    int arr[n+1];arr[0] = 10000;

    for (int i = 1; i < n+1; i++)
        cin>>arr[i];

    for (int i = 1; i < n+1; i++) {
        int j = i-1;
        while(arr[i]>arr[j]) j--;
        cout<<i-j<<' ';
    }
    return 0;
}

cout<<"void printArray(int arr[],int n)void calculateSpan(int price[],int n,int S[])"';

}

```

question

<p>Question description</p><p>Hassan gets a job in a software company in Hyderabad. The training period for the first three months is 20000 salary. Then incremented to 25000 salaries. </p><p>Training is great but they will give you a programming task every day in three months. Hassan must finish it in the allotted time. His teammate Jocelyn gives him a task to complete the concept of Infix to Prefix Conversion for a given expression. can you help him?</p><p>Functional Description:</p>Step 1: Reverse the infix expression i.e A+B*C will become C*B+A. Note while reversing each '(' will become ')' and each ')' becomes '('.Step 2: Obtain the "nearly" postfix expression of the modified expression i.e CB*A+.Step 3: Reverse the postfix expression. Hence in our example prefix is +A*BC.
 <p>Constraints</p><p>the input should be a expressions</p><p>Input Format</p><p>Single line represents the Infix expressions</p><p>Output Format</p><p>Single line represents the Prefix expression</p>

answer

```

// CPP program to convert infix to prefix

#include <bits/stdc++.h>

using namespace std;

```

```

bool isOperator(char c)
{
    return (!isalpha(c) && !isdigit(c));
}

int getPriority(char C)
{
    if (C == '-' || C == '+')
        return 1;
    else if (C == '*' || C == '/')
        return 2;
    else if (C == '^')
        return 3;
    return 0;
}

string infixToPostfix(string infix)
{
    infix = '(' + infix + ')';
    int l = infix.size();
    stack<char> char_stack;
    string output;

    for (int i = 0; i < l; i++) {
        // If the scanned character is an
        // operand, add it to output.
        if (isalpha(infix[i]) || isdigit(infix[i]))
            output += infix[i];
    }
}

```

```

// If the scanned character is an
// '(', push it to the stack.

else if (infix[i] == '(')
    char_stack.push('(');

// If the scanned character is an
// ')', pop and output from the stack
// until an '(' is encountered.

else if (infix[i] == ')') {
    while (char_stack.top() != '(') {
        output += char_stack.top();
        char_stack.pop();
    }

    // Remove '(' from the stack
    char_stack.pop();
}

// Operator found
else
{
    if (isOperator(char_stack.top()))
    {
        if(infix[i] == '^')
        {

            while (getPriority(infix[i]) <= getPriority(char_stack.top()))
            {

                output += char_stack.top();
                char_stack.pop();
            }
        }
    }
}

```

```

        }

        else

        {

            while (getPriority(infix[i]) < getPriority(char_stack.top()))

            {

                output += char_stack.top();

                char_stack.pop();

            }

        }

    }

    // Push current Operator on stack

    char_stack.push(infix[i]);

}

}

while(!char_stack.empty()){

    output += char_stack.top();

    char_stack.pop();

}

return output;

}

```

```

string infixToPrefix(string infix)

{

/* Reverse String

* Replace ( with ) and vice versa

* Get Postfix

* Reverse Postfix * */

int l = infix.size();

```

```
// Reverse infix
reverse(infix.begin(), infix.end());

// Replace ( with ) and vice versa
for (int i = 0; i < l; i++) {

    if (infix[i] == '(') {
        infix[i] = ')';
        i++;
    }

    else if (infix[i] == ')') {
        infix[i] = '(';
        i++;
    }
}

string prefix = infixToPostfix(infix);

// Reverse postfix
reverse(prefix.begin(), prefix.end());

return prefix;

}

// Driver code
int main()
{
    string s;
    cin>>s;
    cout << infixToPrefix(s) << std::endl;
    return 0;
}
```

```
}
```

question

<p>Question description</p><p>Consider the following string transformation:</p>append the character # to the string (we assume that # is lexicographically smaller than all other characters of the string)generate all rotations of the stringsort the rotations in increasing orderbased on this order, construct a new string that contains the last character of each rotation<p>For example, the string babc becomes babc#. Then, the sorted list of rotations is #babc, abc#b, babc#, bc#ba, and c#bab. This yields a string cb#ab.</p><p>Constraints</p>1≤n≤10^6<p>Input</p><p>The only input line contains the transformed string of length n+1. Each character of the original string is one of a–z.</p><p>Output</p><p>Print the original string of length n.</p>

answer

```
#include<bits/stdc++.h>
using namespace std;

int main() {
    int i;
    string s; cin>>s;
    vector<int> v;
    vector<int> a[26];
    int n= s.size();
    for(i=0;i<=n;i++) {
        if (s[i] == '#')
            v.push_back(i);
        else
            a[s[i]-'a'].push_back(i);
    }
    for (int i = 0; i < 26; i++) {
```

```

        for (auto j: a[i])
            v.push_back(j);

    }

    string ans;
    int j = v[0];
    while(s[j] != '#') {
        ans += s[j];
        j = v[j];
    }
    cout<<ans;
    return 0;
}

```

question

<p>Question description</p><p>Given a string, you want to reorder its characters so that no two adjacent characters are the same. What is the lexicographically minimal such
Constraints</p>1≤n≤10^6
Input

The only input line as a string of length n consisting of characters A–Z.

Output

Print the lexicographically minimal reordered string where no two adjacent characters are the same. If it is not possible to create such a string, print -1.
 </p>

answer

```

#include <stdio.h>
#include <string.h>

#define N 1000000
#define A 26

int main() {
    static char cc[N + 1];

```

```
static int kk[A];

int n, i, p, a, b, c;

scanf("%s", cc);
n = strlen(cc);
for(i=0;i<n;i++) {
    a = cc[i] - 'A';
    kk[a]++;
}
for (a = 0; a < A; a++)
if (n < kk[a] * 2 - 1) {
    printf("-1\n");
    return 0;
}
p = -1;
for (i = 0; i < n; i++) {
    a = 0;
    while (a < A && (a == p || kk[a] == 0))
        a++;
    b = 0;
    for (c = 1; c < A; c++)
        if (kk[b] < kk[c])
            b = c;
    a = a != b && n - i - 1 < kk[b] * 2 - 1 ? b : a;
    kk[a]--;
    cc[i] = a + 'A';
    p = a;
}
printf("%s\n", cc);
return 0;
}
```

question

<p>Question description</p><p>Lalitha is a B.Tech student. During her final year Campus Interview, she has an opportunity to get a job in a software company in Bangalore. </p><p>The company provides Five months training period with Rs.30000/month Package. Then it will be incremented to Rs.55000 per month. </p><p>At the end of the training, the examination was conducted for all freshers, Lalitha got a question paper and one of the questions comes under the concept of Queues in data structure that is Circular Linked List Implementation in Circular Queue. </p><p>Can you help?</p><p> </p><figure class="image"></figure><p>Constraints</p><p>First '3' elements inserted in array 0, next '3' elements inserted into array1, remaining elements inserted into array2.</p><p>Input Format</p><p>First line indicates the number of elements to be inserted in the queues.</p><p>Second line indicates the elements.</p><p>Output Format</p><p>first line indicates the dequeue element of Q2</p><p>Second line indicates the dequeue element of Q1</p><p>Third line indicates the dequeue element of Q0</p><p>Note: Refer sample input and output test cases</p>

answer

```
#include <stdio.h>
#include <stdlib.h>

struct node *front = NULL;
struct node *rear = NULL;
struct node
{
    int data;
    struct node *next;
};
void linkedListTraversal(struct node *ptr)
{
    printf("Elements in Circular Queue are:");
    while (ptr->next != NULL)
    {
        printf("%d ", ptr->data);
    }
}
```

```

ptr = ptr->next;
}

printf("%d",ptr->data);

}

void enqueue(int d)
{
    struct node* new_n;

    new_n = (struct node*)malloc(sizeof(struct node));

    if(new_n==NULL){

        printf("Queue is Full");

    }

    else{

        new_n->data = d;

        new_n->next = NULL;

        if(front==NULL){

            front=rear=new_n;

        }

        else{

            rear->next = new_n;

            rear=new_n;

        }

    }

}

int dequeue()

{

    int val = -1;

    struct node *ptr = front;

    if(front==NULL){

        printf("Queue is Empty\n");

    }

    else{

```

```

    front = front->next;
    val = ptr->data;
    free(ptr);
}

return val;
}

int main()
{
    int n,i,t;
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        scanf("%d",&t);
        enqueue(t);
    }
    linkedListTraversal(front);
    printf("\nDeleted value = %d\n",dequeue());
    printf("Deleted value = %d",dequeue());
    linkedListTraversal(front);
    return 0;
}

printf("void enQueue(Queue* q,int value) int deQueue(Queue* q) void displayQueue(struct
Queue* q)");
}

```

question

<p>Question description</p><p>You are given a string. You can remove any number of characters from it, but you cannot change the order of the remaining characters.

How many different strings can you generate?</p><p>Constraints</p>1≤n≤5·10^5<p>
Input
The first input line contains a string of size n. Each character is one of a-z.

Output
Print one integer: the number of strings modulo 10^9+7.

 </p>

answer

```
#include <stdio.h>

#define N 500000
#define MD 1000000007

int main() {
    static char cc[N + 1];
    static int kk[26];
    int i, k, c, kc;

    scanf("%s", cc);
    k = 0;
    for(i=0;cc[i];i++) {
        c = cc[i] - 'a';
        kc = kk[c];
        kk[c] = k + 1;
        k = (k + (kk[c] - kc) % MD) % MD;
    }
    printf("%d\n", (k + MD) % MD);
    return 0;
}
```

question

<p>Question description</p><p>Selvan is very interested in surfing the contents from google. He searches for various coding test on Google. </p><p>One day he searched about online coding competitions, in the retrieval links, he received many links for coding competition. he chooses first link from the goole suggestion list.</p><p>first question for the coding competition is LRU cache implementation using queue concepts. </p><p>Function Description</p>Queue which is implemented using a doubly linked list. The maximum size of the queue will be equal to the total number of frames available

(cache size). The most recently used pages will be near front end and least recently pages will be near the rear end.A Hash with page number as key and address of the corresponding queue node as value.<figure class="image"></figure><p>Constraints</p><p>For this experiment, cache can hold 4 pages.</p><p>Let 10 different pages can be requested (pages to be referenced are numbered from 0 to 9).</p><p>Input Format:</p><p>First line represents n and m, where n is the page number with in the range (0-9) & m is cache size (must be 4 for this problem).</p><p>Next line represents the reference pages.</p><p>Output Format:</p><p>Single line represents the cache frames after the above referenced pages.</p>

answer

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
typedef struct QNode
```

```
{
```

```
    struct QNode *prev, *next;
```

```
    unsigned pageNumber;
```

```
} QNode;
```

```
typedef struct Queue
```

```
{
```

```
    unsigned count;
```

```
    unsigned numberOfFrames;
```

```
    QNode *front, *rear;
```

```
} Queue;
```

```
typedef struct Hash
```

```
{
```

```
    int capacity;
```

```
    QNode* *array;
```

```
} Hash;

QNode* newQNode(unsigned pageNumber)
{
    QNode* temp = (QNode *)malloc( sizeof( QNode ) );
    temp->pageNumber = pageNumber;
    temp->prev = temp->next = NULL;
    return temp;
}
```

```
Queue* createQueue(int numberOfFrames)
{
    Queue* queue = (Queue *)malloc( sizeof( Queue ) );
    queue->count = 0;
    queue->front = queue->rear = NULL;
    queue->numberOfFrames = numberOfFrames;

    return queue;
}
```

```
Hash* createHash( int capacity )
{
    Hash* hash = (Hash *) malloc( sizeof( Hash ) );
    hash->capacity = capacity;
    hash->array = (QNode **) malloc( hash->capacity * sizeof( QNode* ) );
    int i;
    for( i = 0; i < hash->capacity; ++i )
        hash->array[i] = NULL;

    return hash;
}
```

```
}
```

```
int AreAllFramesFull( Queue* queue )
{
    return queue->count == queue->numberOfFrames;
}
```

```
int isQueueEmpty( Queue* queue )
{
    return queue->rear == NULL;
}
```

```
void deQueue( Queue* queue )
{
    if( isQueueEmpty( queue ) )
        return;
```

```
    if (queue->front == queue->rear)
        queue->front = NULL;
```

```
    QNode* temp = queue->rear;
    queue->rear = queue->rear->prev;
```

```
    if (queue->rear)
        queue->rear->next = NULL;
```

```
    free( temp );
```

```
    queue->count--;
}
```

```

void Enqueue( Queue* queue, Hash* hash, unsigned pageNumber )
{
    if ( AreAllFramesFull ( queue ) )
    {
        hash->array[ queue->rear->pageNumber ] = NULL;
        deQueue( queue );
    }

    QNode* temp = newQNode( pageNumber );
    temp->next = queue->front;

    if ( isEmpty( queue ) )
        queue->rear = queue->front = temp;
    else
    {
        queue->front->prev = temp;
        queue->front = temp;
    }

    hash->array[ pageNumber ] = temp;
    queue->count++;

}

void ReferencePage( Queue* queue, Hash* hash, unsigned pageNumber )
{
    QNode* reqPage = hash->array[ pageNumber ];

```

```

if ( reqPage == NULL )

    Enqueue( queue, hash, pageNumber );

else if (reqPage != queue->front)

{

    reqPage->prev->next = reqPage->next;

    if (reqPage->next)

        reqPage->next->prev = reqPage->prev;

    if (reqPage == queue->rear)

    {

        queue->rear = reqPage->prev;

        queue->rear->next = NULL;

    }

    reqPage->next = queue->front;

    reqPage->prev = NULL;

    reqPage->next->prev = reqPage;

}

queue->front = reqPage;

}
}

```

```

int main()

{

    int i,n,m,x;

    scanf("%d %d",&n,&m);

    Queue* q = createQueue( m );

    Hash* hash = createHash( m+n );

    for(i=0;i<n;i++){

        scanf("%d",&x);

```

```
    ReferencePage( q, hash, x);  
}  
  
/*ReferencePage( q, hash, 1);  
ReferencePage( q, hash, 2);  
ReferencePage( q, hash, 3);  
ReferencePage( q, hash, 1);  
ReferencePage( q, hash, 4);  
ReferencePage( q, hash, 5);*/  
printf ("%d ", q->front->pageNumber);  
printf ("%d ", q->front->next->pageNumber);  
printf ("%d ", q->front->next->next->pageNumber);  
printf ("%d ", q->front->next->next->next->pageNumber);  
  
return 0;  
}
```

question

JLk1FRk//Or87/////////////////////////////AABEIAoYEMgMBIgA
CEQEDEQH/xAAZAAEBAQEBAQAAAAAAAAABAMCAQX/xAA+EAACAgIBAgMEBwgCAGEEAwAA
AQIDBBEhEjFE0EiMFxFDRhcoGxwRUzNUKRodHwl1IkgmJDRVNg6Lx/8QAFABAAAAAAAAAAAA
AAAAAAAP/EABQRQAaaaaaaaaaaaaAAAD/2gAMAwEAAhEDEQA/APqAAA2orbaS+LEWpJN
PafZo+dIy01TO221zkuy9FyWYv1Wv7qA1AABtJbb0vizNZFMnpWwb+ZLkbycxUbahFblo1ng0yrcYw6Xr
hgUgmxZWV4sndFpw339UcrxCuTiowk2/7AVgxvyoUyUWnKb/AJYnNOXC2fQ4yhP4SAoBjfkl0aUtuT7
RXc4rzYSsUJwlXJ9upAUg+f4he9dEVOLUve7J8FlNytg5OMojf9uANASPxCG301zlFd5JFFVsLoKcHtAdgBv
S2wAJZZ8OpqEJzS7tLg2ovhfDqh6d0/QDQAk/aFb2ICblvskBWCWPiFTJyUouP8AK1yxHPg5qM4Thvs51
CoHFtsKYdc3pfmT/tCHDIXYov8AmaAqIKMlUUmkl6s9TukmntPs0Y5E63iyINddbS7ep551dOJCxRfRpAq
G4JJelQ7xhOSXdpcloqtjdWpwfDA7MLcymmfTKT6vVJdjW2fl1ufS5a9F3PnYtqWRZKVUpOT9F25A+mn
tJr1Bkr4VJdGn1Jb36DlyI48YuSb29cAagxvyYY7gpp+16r0Mn4hBPflz6H/NoCsHLsgq/Mcl0a3sm/aEO6qs
cF/NoCsE92THyOqClJTT04rsZeH3uUFXKM29v2n2ArjZCUnGMk3Huk+x0T40qnfcq4OMk/ab9RdmQrn0
KMPz+EQKAYUZdd0nDTjNfyF2XXTb0TT7b2BuCSOfBzSICe+zaKwPJSUYuUnpLuzGrLqus6INT/ly8Rt1U
6+mXtJPq1x3OsGyMq4wVco9MVy13AqBNPOhGbhCErGu/SjSjJrvT6dqS7p9wNQTW51dVkoSjLcfh6m1
NvnVqfS479GB2eSkoRcn2S2yPxC9xg64xmnw+pdjqN3m4VicZLpr7y9eAKar12wU4PaZ0fOxcuNWPGCh
KcltJdzHyIZEW47TXdMDUE92ZCufRGMpz9VH0OqMfq7cUnGS7xYGwBNPOrhZKDjJyJxwu4FJqs2FlnI
yjKEn2Uju/KhROMZp+16oDYGFVG5Tl0uMY87kZvxCG241zlFd5JAVg4puhdDqg9r8jG3NhCxwjGU5Lv
0gUgmrza7LY1xjlcviuwtza6bZVyt+HqBSDBZcPI82acVvWn3Zn+0lbTIXOMX2k0BWDO26NVPmv2o8
djB+IQ1uNc5L1euwFYJnn1dMXBSnJ/ypco6x8qF8nFJxkvRgbgAAAAAAAAAAAAAAAAACfP8Aqdn4fmjvF+q1/dR5
lwIJzThBbk9aX4nWPFwohGS00tMDQGSD/wBJacV5OuH6moEVxs+K2J+seP7FpNIy0pjbs9WR/uZylm2
RcPLjDfDIlsDey2FuLa4SUkovt8jPw6KWkml29nUMfyosDUEzZOL/ABejrDrIVjhNakt8ARVTsWzdOFXmS
Ta79juxZF19U3R0uD7pmItF1WQ7sfT6veizqDy7LYuaVUF3S52B5fZCGSuirzL9fHsYZkrpeW7a4w1Lhp7N
r6boZPn0JSbWmmZX1Zd/TKUIrT4in/cDTxT9xD736Gue2sSwvXSGdTK+hKC9pPehGNmRjyhfBQb44/M
DCi2+FEIwxtx1333NMCuvt29cHBSe0jm6ZRhy1XGyK917KMaNy13fJOTe9L0A1ObOny5dfu6e/kdHNsP
MqlDeupaAkpuscNY20lXvhnn+/Pv2tPfKXpyzypZIMPkjXFpdpbO8Ki2m6x2LfV/NvuBYReGJateueruW
k2DTOqNisjrctrkDKcU/FYcem/7M68UX/jxfr1fozqVM34hG1R9hR1vf2M9z6p3UKNcdvq33AzC6o0wds
Ot/yr4sxyLMiePPqojGGvV8o1y8edtchD34cpGdizb63CUIwWuee4Hkv4R+C/MZH8Lh8onbptfhvldPt/D
a+j7dRZLAhWo7mktrYG9MYrGhFja6VwT+F/V5fe/RFVacaoxdRSMMCmduo2R03LfcCki8P/AH2R97/
JaS4dM6rLnOOIKW1z37gcR/isvu/oPE/3df3jrlpujkq+lKT1ppmN9WXkKLICKSfEU/7gaelJO3HT5Tk/0Nsx
L6JYtcJHGXT022hwjtRlt89uxtkwIZjzjFbk1wglb5P9mUr4vn+59CElxrUEl061owWM54MaZ+zJL+jM4vN
hDy1XFtLSnsDZURox7lwbaab5+Rx4b9Vxz3RjuvHcJS3KW9sywoZFL8qda6Nt9WwPMP61f/e/VjwxJws
sfMnLTZi0zryL5SjqMpbT335Zx5N+NbKVEVOEuXFvsAzkoZNE48Sb0/t7HlqUvFa01tdP8Ak6rouuyI25C
UVH3Yo6nTN+IQtUfYUdN7+YHPii/8eL/+RXD3I/lwzqp3UKNa2+rfc3itRSfogJ/Efkvmjrbj4emu6q/QZtcr
ceUYLcm1waVw1RCE1/Kk1+AGHh0UsVSXeTezixKHilbjx1Ln+55CvJxXKNUVZW3tbFy7x6LHe779dXpFeg
HFUU/FLdreo7X9i4lqpnHPsscfYIHSe/kVAS+i/VH80dy+oP/APV+h7l1O7HICPvd0Z1K+WLZXZWotQ6Y89
+AHhySxU0uW3s4xeM7ISN8OuVWPGE1qS3wcUuzhmXWSjqMuz2Bjh2Wx65wp8xyfMtmsl3zzYWp6
F2fj15N+NbKVCU4S56W+xpT9jnB126hBL3V6gUkWOk/Erm1ylwWnzV5yz7nSk2u6fqgNPE0IGuxE8pc
MZiUszHUltPuvxHk35N0ZxxUIR51vua5FM55VM4x3GL5e+wDxFtYj16tJmuPCMMMeYrjpR7dUrqpQfr6/
Alh9Mph5arjNLiMtgeY3/Hm5Ele7rej3wuK8qc/wCzy02a4mPKnqnY92T7mKqyMWyXkRU65PtvB7akv
FatLvHb/ueRin4tPa3qO1/RGcfNfiVbu11Nb0vRaZRCma8Qna4+w46T39iA6zZ1QjF2Q65b9mJhkzyJ48u
umMYfHfKNs2idvR0r34PaXxMbY5mRW4ShGC9dPuAt/hUfkvzKsaK+i1rS04L8jCdNsvDIV0+2tcbXxKaYu
NNcZLTUUn/AEak8LilCx653o9fHiy16x5/od4FM6YTvkDny2uT10zfiEben2FHW9gUgyTv+ktOK8nXD9TU
AAAAAAAP/EABQRQAaaaaaaaaaaaaAAAD/2gAMAwEAAhEDEQA/APqAAA2orbaS+LEWpJN
PafZo+dIy01TO221zkuy9FyWYv1Wv7qA1AABtJbb0vizNZFMnpWwb+ZLkbycxUbahFblo1ng0yrcYw6Xr
hgUgmxZWV4sndFpw339UcrxCuTiowk2/7AVgxvyoUyUWnKb/AJYnNOXC2fQ4yhP4SAoBjfkl0aUtuT7
RXc4rzYSsUJwlXJ9upAUg+f4he9dEVOLUve7J8FlNytg5OMojf9uANASPxCG301zlFd5JFFVsLoKcHtAdgBv
S2wAJZZ8OpqEJzS7tLg2ovhfDqh6d0/QDQAk/aFb2ICblvskBWCWPiFTJyUouP8AK1yxHPg5qM4Thvs51
CoHFtsKYdc3pfmT/tCHDIXYov8AmaAqIKMlUUmkl6s9TukmntPs0Y5E63iyINddbS7ep551dOJCxRfRpAq
G4JJelQ7xhOSXdpcloqtjdWpwfDA7MLcymmfTKT6vVJdjW2fl1ufS5a9F3PnYtqWRZKVUpOT9F25A+mn
tJr1Bkr4VJdGn1Jb36DlyI48YuSb29cAagxvyYY7gpp+16r0Mn4hBPflz6H/NoCsHLsgq/Mcl0a3sm/aEO6qs
cF/NoCsE92THyOqClJTT04rsZeH3uUFXKM29v2n2ArjZCUnGMk3Huk+x0T40qnfcq4OMk/ab9RdmQrn0
KMPz+EQKAYUZdd0nDTjNfyF2XXTb0TT7b2BuCSOfBzSICe+zaKwPJSUYuUnpLuzGrLqus6INT/ly8Rt1U
6+mXtJPq1x3OsGyMq4wVco9MVy13AqBNPOhGbhCErGu/SjSjJrvT6dqS7p9wNQTW51dVkoSjLcfh6m1
NvnVqfS479GB2eSkoRcn2S2yPxC9xg64xmnw+pdjqN3m4VicZLpr7y9eAKar12wU4PaZ0fOxcuNWPGCh
KcltJdzHyIZEW47TXdMDUE92ZCufRGMpz9VH0OqMfq7cUnGS7xYGwBNPOrhZKDjJyJxwu4FJqs2FlnI
yjKEn2Uju/KhROMZp+16oDYGFVG5Tl0uMY87kZvxCG241zlFd5JAVg4puhdDqg9r8jG3NhCxwjGU5Lv
0gUgmrza7LY1xjlcviuwtza6bZVyt+HqBSDBZcPI82acVvWn3Zn+0lbTIXOMX2k0BWDO26NVPmv2o8
djB+IQ1uNc5L1euwFYJnn1dMXBSnJ/ypco6x8qF8nFJxkvRgbgAAAAAAAAAAAAAAAAACfP8Aqdn4fmjvF+q1/dR5
lwIJzThBbk9aX4nWPFwohGS00tMDQGSD/wBJacV5OuH6moEVxs+K2J+seP7FpNIy0pjbs9WR/uZylm2
RcPLjDfDIlsDey2FuLa4SUkovt8jPw6KWkml29nUMfyosDUEzZOL/ABejrDrIVjhNakt8ARVTsWzdOFXmS
Ta79juxZF19U3R0uD7pmItF1WQ7sfT6veizqDy7LYuaVUF3S52B5fZCGSuirzL9fHsYZkrpeW7a4w1Lhp7N
r6boZPn0JSbWmmZX1Zd/TKUIrT4in/cDTxT9xD736Gue2sSwvXSGdTK+hKC9pPehGNmRjyhfBQb44/M
DCi2+FEIwxtx1333NMCuvt29cHBSe0jm6ZRhy1XGyK917KMaNy13fJOTe9L0A1ObOny5dfu6e/kdHNsP
MqlDeupaAkpuscNY20lXvhnn+/Pv2tPfKXpyzypZIMPkjXFpdpbO8Ki2m6x2LfV/NvuBYReGJateueruW
k2DTOqNisjrctrkDKcU/FYcem/7M68UX/jxfr1fozqVM34hG1R9hR1vf2M9z6p3UKNcdvq33AzC6o0wds
Ot/yr4sxyLMiePPqojGGvV8o1y8edtchD34cpGdizb63CUIwWuee4Hkv4R+C/MZH8Lh8onbptfhvldPt/D
a+j7dRZLAhWo7mktrYG9MYrGhFja6VwT+F/V5fe/RFVacaoxdRSMMCmduo2R03LfcCki8P/AH2R97/
JaS4dM6rLnOOIKW1z37gcR/isvu/oPE/3df3jrlpujkq+lKT1ppmN9WXkKLICKSfEU/7gaelJO3HT5Tk/0Nsx
L6JYtcJHGXT022hwjtRlt89uxtkwIZjzjFbk1wglb5P9mUr4vn+59CElxrUEl061owWM54MaZ+zJL+jM4vN
hDy1XFtLSnsDZURox7lwbaab5+Rx4b9Vxz3RjuvHcJS3KW9sywoZFL8qda6Nt9WwPMP61f/e/VjwxJws
sfMnLTZi0zryL5SjqMpbT335Zx5N+NbKVEVOEuXFvsAzkoZNE48Sb0/t7HlqUvFa01tdP8Ak6rouuyI25C
UVH3Yo6nTN+IQtUfYUdN7+YHPii/8eL/+RXD3I/lwzqp3UKNa2+rfc3itRSfogJ/Efkvmjrbj4emu6q/QZtcr
ceUYLcm1waVw1RCE1/Kk1+AGHh0UsVSXeTezixKHilbjx1Ln+55CvJxXKNUVZW3tbFy7x6LHe779dXpFeg
HFUU/FLdreo7X9i4lqpnHPsscfYIHSe/kVAS+i/VH80dy+oP/APV+h7l1O7HICPvd0Z1K+WLZXZWotQ6Y89
+AHhySxU0uW3s4xeM7ISN8OuVWPGE1qS3wcUuzhmXWSjqMuz2Bjh2Wx65wp8xyfMtmsl3zzYWp6
F2fj15N+NbKVCU4S56W+xpT9jnB126hBL3V6gUkWOk/Erm1ylwWnzV5yz7nSk2u6fqgNPE0IGuxE8pc
MZiUszHUltPuvxHk35N0ZxxUIR51vua5FM55VM4x3GL5e+wDxFtYj16tJmuPCMMMeYrjpR7dUrqpQfr6/
Alh9Mph5arjNLiMtgeY3/Hm5Ele7rej3wuK8qc/wCzy02a4mPKnqnY92T7mKqyMWyXkRU65PtvB7akv
FatLvHb/ueRin4tPa3qO1/RGcfNfiVbu11Nb0vRaZRCma8Qna4+w46T39iA6zZ1QjF2Q65b9mJhkzyJ48u
umMYfHfKNs2idvR0r34PaXxMbY5mRW4ShGC9dPuAt/hUfkvzKsaK+i1rS04L8jCdNsvDIV0+2tcbXxKaYu
NNcZLTUUn/AEak8LilCx653o9fHiy16x5/od4FM6YTvkDny2uT10zfiEben2FHW9gUgyTv+ktOK8nXD9TU
AAAAAAAP/EABQRQAaaaaaaaaaaaaAAAD/2gAMAwEAAhEDEQA/APqAAA2orbaS+LEWpJN
PafZo+dIy01TO221zkuy9FyWYv1Wv7qA1AABtJbb0vizNZFMnpWwb+ZLkbycxUbahFblo1ng0yrcYw6Xr
hgUgmxZWV4sndFpw339UcrxCuTiowk2/7AVgxvyoUyUWnKb/AJYnNOXC2fQ4yhP4SAoBjfkl0aUtuT7
RXc4rzYSsUJwlXJ9upAUg+f4he9dEVOLUve7J8FlNytg5OMojf9uANASPxCG301zlFd5JFFVsLoKcHtAdgBv
S2wAJZZ8OpqEJzS7tLg2ovhfDqh6d0/QDQAk/aFb2ICblvskBWCWPiFTJyUouP8AK1yxHPg5qM4Thvs51
CoHFtsKYdc3pfmT/tCHDIXYov8AmaAqIKMlUUmkl6s9TukmntPs0Y5E63iyINddbS7ep551dOJCxRfRpAq
G4JJelQ7xhOSXdpcloqtjdWpwfDA7MLcymmfTKT6vVJdjW2fl1ufS5a9F3PnYtqWRZKVUpOT9F25A+mn
tJr1Bkr4VJdGn1Jb36DlyI48YuSb29cAagxvyYY7gpp+16r0Mn4hBPflz6H/NoCsHLsgq/Mcl0a3sm/aEO6qs
cF/NoCsE92THyOqClJTT04rsZeH3uUFXKM29v2n2ArjZCUnGMk3Huk+x0T40qnfcq4OMk/ab9RdmQrn0
KMPz+EQKAYUZdd0nDTjNfyF2XXTb0TT7b2BuCSOfBzSICe+zaKwPJSUYuUnpLuzGrLqus6INT/ly8Rt1U
6+mXtJPq1x3OsGyMq4wVco9MVy13AqBNPOhGbhCErGu/SjSjJrvT6dqS7p9wNQTW51dVkoSjLcfh6m1
NvnVqfS479GB2eSkoRcn2S2yPxC9xg64xmnw+pdjqN3m4VicZLpr7y9eAKar12wU4PaZ0fOxcuNWPGCh
KcltJdzHyIZEW47TXdMDUE92ZCufRGMpz9VH0OqMfq7cUnGS7xYGwBNPOrhZKDjJyJxwu4FJqs2FlnI
yjKEn2Uju/KhROMZp+16oDYGFVG5Tl0uMY87kZvxCG241zlFd5JAVg4puhdDqg9r8jG3NhCxwjGU5Lv
0gUgmrza7LY1xjlcviuwtza6bZVyt+HqBSDBZcPI82acVvWn3Zn+0lbTIXOMX2k0BWDO26NVPmv2o8
djB+IQ1uNc5L1euwFYJnn1dMXBSnJ/ypco6x8qF8nFJxkvRgbgAAAAAAAAAAAAAAAAACfP8Aqdn4fmjvF+q1/dR5
lwIJzThBbk9aX4nWPFwohGS00tMDQGSD/wBJacV5OuH6moEVxs+K2J+seP7FpNIy0pjbs9WR/uZylm2
RcPLjDfDIlsDey2FuLa4SUkovt8jPw6KWkml29nUMfyosDUEzZOL/ABejrDrIVjhNakt8ARVTsWzdOFXmS
Ta79juxZF19U3R0uD7pmItF1WQ7sfT6veizqDy7LYuaVUF3S52B5fZCGSuirzL9fHsYZkrpeW7a4w1Lhp7N
r6boZPn0JSbWmmZX1Zd/TKUIrT4in/cDTxT9xD736Gue2sSwvXSGdTK+hKC9pPehGNmRjyhfBQb44/M
DCi2+FEIwxtx1333NMCuvt29cHBSe0jm6ZRhy1XGyK917KMaNy13fJOTe9L0A1ObOny5dfu6e/kdHNsP
MqlDeupaAkpuscNY20lXvhnn+/Pv2tPfKXpyzypZIMPkjXFpdpbO8Ki2m6x2LfV/NvuBYReGJateueruW
k2DTOqNisjrctrkDKcU/FYcem/7M68UX/jxfr1fozqVM34hG1R9hR1vf2M9z6p3UKNcdvq33AzC6o0wds
Ot/yr4sxyLMiePPqojGGvV8o1y8edtchD34cpGdizb63CUIwWuee4Hkv4R+C/MZH8Lh8onbptfhvldPt/D
a+j7dRZLAhWo7mktrYG9MYrGhFja6VwT+F/V5fe/RFVacaoxdRSMMCmduo2R03LfcCki8P/AH2R97/
JaS4dM6rLnOOIKW1z37gcR/isvu/oPE/3df3jrlpujkq+lKT1ppmN9WXkKLICKSfEU/7gaelJO3HT5Tk/0Nsx
L6JYtcJHGXT022hwjtRlt89uxtkwIZjzjFbk1wglb5P9mUr4vn+59CElxrUEl061owWM54MaZ+zJL+jM4vN
hDy1XFtLSnsDZURox7lwbaab5+Rx4b9Vxz3RjuvHcJS3KW9sywoZFL8qda6Nt9WwPMP61f/e/VjwxJws
sfMnLTZi0zryL5SjqMpbT335Zx5N+NbKVEVOEuXFvsAzkoZNE48Sb0/t7HlqUvFa01tdP8Ak6rouuyI25C
UVH3Yo6nTN+IQtUfYUdN7+YHPii/8eL/+RXD3I/lwzqp3UKNa2+rfc3itRSfogJ/Efkvmjrbj4emu6q/QZtcr
ceUYLcm1waVw1RCE1/Kk1+AGHh0UsVSXeTezixKHilbjx1Ln+55CvJxXKNUVZW3tbFy7x6LHe779dXpFeg
HFUU/FLdreo7X9i4lqpnHPsscfYIHSe/kVAS+i/VH80dy+oP/APV+h7l1O7HICPvd0Z1K+WLZXZWotQ6Y89
+AHhySxU0uW3s4xeM7ISN8OuVWPGE1qS3wcUuzhmXWSjqMuz2Bjh2Wx65wp8xyfMtmsl3zzYWp6
F2fj15N+NbKVCU4S56W+xpT9jnB126hBL3V6gUkWOk/Erm1ylwWnzV5yz7nSk2u6fqgNPE0IGuxE8pc
MZiUszHUltPuvxHk35N0ZxxUIR51vua5FM55VM4x3GL5e+wDxFtYj16tJmuPCMMMeYrjpR7dUrqpQfr6/
Alh9Mph5arjNLiMtgeY3/Hm5Ele7rej3wuK8qc/wCzy02a4mPKnqnY92T7mKqyMWyXkRU65PtvB7akv
FatLvHb/ueRin4tPa3qO1/RGcfNfiVbu11Nb0vRaZRCma8Qna4+w46T39iA6zZ1QjF2Q65b9mJhkzyJ48u
umMYfHfKNs2idvR0r34PaXxMbY5mRW4ShGC9dPuAt/hUfkvzKsaK+i1rS04L8jCdNsvDIV0+2tcbXxKaYu
NNcZLTUUn/AEak8LilCx653o9fHiy16x5/od4FM6YTvkDny2uT10zfiEben2FHW9gUgyTv+ktOK8nXD9TU
AAAAAAAP/EABQRQAaaaaaaaaaaaaAAAD/2gAMAwEAAhEDEQA/APqAAA2orbaS+LEWpJN
PafZo+dIy01TO221zkuy9FyWYv1Wv7qA1AABtJbb0vizNZFMnpWwb+ZLkbycxUbahFblo1ng0yrcYw6Xr
hgUgmxZWV4sndFpw339UcrxCuTiowk2/7AVgxvyoUyUWnKb/AJYnNOXC2fQ4yhP4SAoBjfkl0aUtuT7
RXc4rzYSsUJwlXJ9upAUg+f4he9dEVOLUve7J8FlNytg5OMojf9uANASPxCG301zlFd5JFFVsLoKcHtAdgBv
S2wAJZZ8OpqEJzS7tLg2ovhfDqh6d0/QDQAk/aFb2ICblvskBWCWPiFTJyUouP8AK1yxHPg5qM4Thvs51
CoHFtsKYdc3pfmT/tCHDIXYov8AmaAqIKMlUUmkl6s9TukmntPs0Y5E63iyINddbS7ep551dOJCxRfRpAq
G4JJelQ7xhOSXdpcloqtjdWpwfDA7MLcymmfTKT6vVJdjW2fl1ufS5a9F3PnYtqWRZKVUpOT9F25A+mn
tJr1Bkr4VJdGn1Jb36DlyI48YuSb29cAagxvyYY7gpp+16r0Mn4hBPflz6H/NoCsHLsgq/Mcl0a3sm/aEO6qs
cF/NoCsE92THyOqClJTT04rsZeH3uUFXKM29v2n2ArjZCUnGMk3Huk+x0T40qnfcq4OMk/ab9RdmQrn0
KMPz+EQKAYUZdd0nDTjNfyF2XXTb0TT7b2BuCSOfBzSICe+zaKwPJSUYuUnpLuzGrLqus6INT/ly8Rt1U
6+mXtJPq1x3OsGyMq4wVco9MVy13AqBNPOhGbhCErGu/SjSjJrvT6dqS7p9wNQTW51dVkoSjLcfh6m1
NvnVqfS479GB2eSkoRcn2S2yPxC9xg64xmnw+pdjqN3m4VicZLpr7y9eAKar12wU4PaZ0fOxcuNWPGCh
KcltJdzHyIZEW47TXdMDUE92ZCufRGMpz9VH0OqMfq7cUnGS7xYGwBNPOrhZKDjJyJxwu4FJqs2FlnI
yjKEn2Uju/KhROMZp+16oDYGFVG5Tl0uMY87kZvxCG241zlFd5JAVg4puhdDqg9r8jG3NhCxwjGU5Lv
0gUgmrza7LY1xjlcviuwtza6bZVyt+HqBSDBZcPI82acVvWn3Zn+0lbTIXOMX2k0BWDO26NVPmv2o8
djB+IQ1uNc5L1euwFYJnn1dMXBSnJ/ypco6x8qF8nFJxkvRgbgAAAAAAAAAAAAAAAAACfP8Aqdn4fmjvF+q1/dR5
lwIJzThBbk9aX4nWPFwohGS00tMDQGSD/wBJacV5OuH6moEVxs+K2J+seP7FpNIy0pjbs9WR/uZylm2
RcPLjDfDIlsDey2FuLa4SUkovt8jPw6KWkml29nUMfyosDUEzZOL/ABejrDrIVjhNakt8ARVTsWzdOFXmS
Ta79juxZF19U3R0uD7pmItF1WQ7sfT6veizqDy7LYuaVUF3S52B5fZCGSuirzL9fHsYZkrpeW7a4w1Lhp7N
r6boZPn0JSbWmmZX1Zd/TKUIrT4in/cDTxT9xD736Gue2sSwvXSGdTK+hKC9pPehGNmRjyhfBQb44/M
DCi2+FEIwxtx1333NMCuvt29cHBSe0jm6ZRhy1XGyK917KMaNy13fJOTe9L0A1ObOny5dfu6e/kdHNsP
MqlDeupaAkpuscNY20lXvhnn+/Pv2tPfKXpyzypZIMPkjXFpdpbO8Ki2m6x2LfV/NvuBYReGJateueruW
k2DTOqNisjrctrkDKcU/FYcem/7M68UX/jxfr1fozqVM34hG1R9hR1vf2M9z6p3UKNcdvq33AzC6o0wds
Ot/yr4sxyLMiePPqojGGvV8o1y8edtchD34cpGdizb63CUIwWuee4Hkv4R+C/MZH8Lh8onbptfhvldPt/D
a+j7dRZLAhWo7mktrYG9MYrGhFja6VwT+F/V5fe/RFVacaoxdRSMMCmduo2R03LfcCki8P/AH2R97/
JaS4dM6rLnOOIKW1z37gcR/isvu/oPE/3df3jrlpujkq+lKT1ppmN9WXkKLICKSfEU/7gaelJO3HT5Tk/0Nsx
L6JYtcJHGXT022hwjtRlt89uxtkwIZjzjFbk1wglb5P9mUr4vn+59CElxrUEl061owWM54MaZ+zJL+jM4vN
hDy1XFtLSnsDZURox7lwbaab5+Rx4b9Vxz3RjuvHcJS3KW9sywoZFL8qda6Nt9WwPMP61f/e/VjwxJws
sfMnLTZi0zryL5SjqMpbT335Zx5N+NbKVEVOEuXFvsAzkoZNE48Sb0/t7HlqUvFa01tdP8Ak6rouuyI25C
UVH3Yo6nTN+IQtUfYUdN7+YHPii/8eL/+RXD3I/lwzqp3UKNa2+rfc3itRSfogJ/Efkvmjrbj4emu6q/QZtcr
ceUYLcm1waVw1RCE1/Kk1+AGHh0UsVSXeTezixKHilbjx1Ln+55CvJxXKNUVZW3tbFy7x6LHe779dXpFeg
HFUU/FLdreo7X9i4lqpnHPsscfYIHSe/kVAS+i/VH80dy+oP/APV+h7l1O7HICPvd0Z1K+WLZXZWotQ6Y89
+AHhySxU0uW3s4xeM7ISN8OuVWPGE1qS3wcUuzhmXWSjqMuz2Bjh2Wx65wp8xyfMtmsl3zzYWp6
F2fj15N+NbKVCU4S56W+xpT9jnB126hBL3V6gUkWOk/Erm1ylwWnzV5yz7nSk2u6fqgNPE0IGuxE8pc
MZiUszHUltPuvxHk35N0ZxxUIR51vua5FM55VM4x3GL5e+wDxFtYj16tJmuPCMMMeYrjpR7dUrqpQfr6/
Alh9Mph5arjNLiMtgeY3/Hm5Ele7rej3wuK8qc/wCzy02a4mPKnqnY92T7mKqyMWyXkRU65PtvB7akv
FatLvHb/ueRin4tPa3qO1/RGcfNfiVbu11Nb0vRaZRCma8Qna4+w46T39iA6zZ1QjF2Q65b9mJhkzyJ48u
umMYfHfKNs2idvR0r34PaXxMbY5mRW4ShGC9dPuAt/hUfkvzKsaK+i1rS04L8jCdNsvDIV0+2tcbXxKaYu
NNcZLTUUn/AEak8LilCx653o9fHiy16x5/od4FM6YTvkDny2uT10zfiEben2FHW9gUgyTv+ktOK8nXD9TU
AAAAAAAP/EABQRQAaaaaaaaaaaaaAAAD/2gAMAwEAAhEDEQA/APqAAA2orbaS+LEWpJN
PafZo+dIy01TO221zkuy9FyWYv1Wv7qA1AABtJbb0vizNZFMnpWwb+ZLkbycxUbahFblo1ng0yrcYw6Xr
hgUgmxZWV4sndFpw339UcrxCuTiowk2/7AVgxvyoUyUWnKb/AJYnNOXC2fQ4yhP4SAoBjfkl0aUtuT7
RXc4rzYSsUJwlXJ9upAUg+f4he9dEVOLUve7J8FlNytg5OMojf9uANASPxCG301zlFd5JFFVsLoKcHtAdgBv
S2wAJZZ8OpqEJzS7tLg2ovhfDqh6d0/QDQAk/aFb2ICblvskBWCWPiFTJyUouP8AK1yxHPg5qM4Thvs51
CoHFtsKYdc3pfmT/tCHDIXYov8AmaAqIKMlUUmkl6s9TukmntPs0Y5E63iyINddbS7ep551dOJCxRfRpAq
G4JJelQ7xhOSXdpcloqtjdWpwfDA7MLcymmfTKT6vVJdjW2fl1ufS5a9F3PnYtqWRZKVUpOT9F25A+mn
tJr1Bkr4VJdGn1Jb36DlyI48YuSb29cAagxvyYY7gpp+16r0Mn4hBPflz6H/NoCsHLsgq/Mcl0a3sm/aEO6qs
cF/NoCsE92THyOqClJTT04rsZeH3uUFXKM29v2n2ArjZCUnGMk3Huk+x0T40qnfcq4OMk/ab9RdmQrn0
KMPz+EQKAYUZdd0nDTjNfyF2XXTb0TT7b2BuCSOfBzSICe+zaKwPJSUYuUnpLuzGrLqus6INT/ly8Rt1U
6+mXtJPq1x3OsGyMq4wVco9MVy13AqBNPOhGbhCErGu/SjSjJrvT6dqS7p9wNQTW51dVkoSjLcfh6m1
NvnVqfS479GB2eSkoRcn2S2yPxC9xg64xmnw+pdjqN3m4VicZLpr7y9eAKar12wU4PaZ0fOxcuNWPGCh
KcltJdzHyIZEW47TXdMDUE92ZCufRGMpz9VH0OqMfq7cUnGS7xYGwBNPOrhZKDjJyJxwu4FJqs2FlnI
yjKEn2Uju/KhROMZp+16oDYGFVG5Tl0uMY87kZvxCG241zlFd5JAVg4puhdDqg9r8jG3NhCxwjGU5Lv
0gUgmrza7LY1xjlcviuwtza6bZVyt+HqBSDBZcPI82acVvWn3Zn+0lbTIXOMX2k0BWDO26NVPmv2o8
djB+IQ1uNc5L1euwFYJnn1dMXBSnJ/ypco6x8qF8nFJxkvRgbgAAAAAAAAAAAAAAAAACfP8Aqdn4fmjvF+q1/dR5
lwIJzThBbk9aX4nWPFwohGS00tMDQGSD/wBJacV5OuH6moEVxs+K2J+seP7FpNIy0pjbs9WR/uZylm2
RcPLjDfDIlsDey2FuLa4SUkovt8jPw6KWkml29nUMfyosDUEzZOL/ABejrDrIVjhNakt8ARVTsWzdOFXmS
Ta79juxZF19U3R0uD7pmItF1WQ7sfT6veizqDy7LYuaVUF3S52B5fZCGSuirzL9fHsYZkrpeW7a4w1Lhp7N
r6boZPn0JSbWmmZX1Zd/TKUIrT4in/cDTxT9xD736Gue2sSwvXSGdTK+hKC9pPehGNmRjyhfBQb44/M
DCi2+FEIwxtx1333NMCuvt29cHBSe0jm6ZRhy1XGyK917KMaNy13fJOTe9L0A1ObOny5dfu6e/kdHNsP
MqlDeupaAkpuscNY20lXvhnn+/Pv2tPfKXpyzypZIMPkjXFpdpbO8Ki2m6x2LfV/NvuBYReGJateueruW
k2DTOqNisjrctrkDKcU/FYcem/7M68UX/jxfr1fozqVM34hG1R9hR1vf2M9z6p3UKNcdvq33AzC6o0wds
Ot/yr4sxyLMiePPqojGGvV8o1y8edtchD34cpGdizb63CUIwWuee4Hkv4R+C/MZH8Lh8onbptfhvldPt/D
a+j7dRZLAhWo7mktrYG9MYrGhFja6VwT+F/V5fe/RFVacaoxdRSMMCmduo2R03LfcCki8P/AH2R97/
JaS4dM6rLnOOIKW1z37gcR/isvu/oPE/3df3jrlpujkq+lKT1ppmN9WXkKLICKSfEU/7gaelJO3HT5Tk/0Nsx
L6JYtcJHGXT022hwjtRlt89uxtkwIZjzjFbk1wglb5P9mUr4vn+59CElxrUEl061owWM54MaZ+zJL+jM4vN
hDy1XFtLSnsDZURox7lwbaab5+Rx4b9Vxz3RjuvHcJS3KW9sywoZFL8qda6Nt9WwPMP61f/e/VjwxJws
sfMnLTZi0zryL5SjqMpbT335Zx5N+NbKVEVOEuXFvsAzkoZNE48Sb0/t7HlqUvFa01tdP8Ak6rouuyI25C
UVH3Yo6nTN+IQtUfYUdN7+YHPii/8eL/+RXD3I/lwzqp3UKNa2+rfc3itRSfogJ/Efkvmjrbj4emu6q/QZtcr
ceUYLcm1waVw1RCE1/Kk1+AGHh0UsVSXeTezixKHilbjx1Ln+55CvJxXKNUVZW3tbFy7x6LHe779dXpFeg
HFUU/FLdreo7X9i4lqpnHPsscfYIHSe/kVAS+i/VH80dy+oP/APV+h7l1O7HICPvd0Z1K+WLZXZWotQ6Y89
+AHhySxU0uW3s4xeM7ISN8OuVWPGE1qS3wcUuzhmXWSjqMuz2Bjh2Wx65wp8xyfMtmsl3zzYWp6
F2fj15N+NbKVCU4S56W+xpT9jnB126hBL3V6gUkWOk/Erm1ylwWnzV5yz7nSk2u6fqgNPE0IGuxE8pc
MZiUszHUltPuvxHk35N0ZxxUIR51vua5FM55VM4x3GL5e+wDxFtYj16tJmuPCMMMeYrjpR7dUrqpQfr6/
Alh9Mph5arjNLiMtgeY3/Hm5Ele7rej3wuK8qc/wCzy02a4mPKnqnY92T7mKqyMWyXkRU65PtvB7akv
FatLvHb/ueRin4tPa3qO1/RGcfNfiVbu11Nb0vRaZRCma8Qna4+w46T39iA6zZ1QjF2Q65b9mJhkzyJ48u
umMYfHfKNs2idvR0r34PaXxMbY5mRW4ShGC9dPuAt/hUfkvzKsaK+i1rS04L8jCdNsvDIV0+2tcbXxKaYu
NNcZLTUUn/AEak8LilCx653o9fHiy16x5/od4FM6YTvkDny2uT10zfiEben2FHW9gUgyTv+ktOK8nXD9TU
AAAAAAAP/EABQRQAaaaaaaaaaaaaAAAD/2gAMAwEAAhEDEQA/APqAAA2orbaS+LEWpJN
PafZo+dIy01TO221zkuy9FyWYv1Wv7qA1AABtJbb0vizNZFMnpWwb+ZLkbycxUbahFblo1ng0yrcYw6Xr
hgUgmxZWV4sndFpw339UcrxCuTiowk2/7AVgxvyoUyUWnKb/AJYnNOXC2fQ4yhP4SAoBjfkl0aUtuT7
RXc4rzYSsUJwlXJ9upAUg+f4he9dEVOLUve7J8FlNytg5OMojf9uANASPxCG301zlFd5JFFVsLoKcHtAdgBv
S2wAJZZ8OpqEJzS7tLg2ovhfDqh6d0/QDQAk/aFb2ICblvskBWCWPiFTJyUouP8AK1yxHPg5qM4Thvs51
CoHFtsKYdc3pfmT/tCHDIXYov8AmaAqIKMlUUmkl6s9TukmntPs0Y5E63iyINddbS7ep551dOJCxRfRpAq
G4JJelQ7xhOSXdpcloqtjdWpwfDA7MLcymmfTKT6vVJdjW2fl1ufS5a9F3PnYtqWRZKVUpOT9F25A+mn
tJr1Bkr4VJdGn1

AAADaSbfZA8kIKLi+z
WmBHHlyMht0QioJ63L1NMbJnO2VV0VGyPw9TaEK6K9R9mC57kuLu/Lnka1BcR+OC0AATyvms6FKS6
Wt/b6lBFZ/Fq/u/oznJds+Fdc3Hcfj8wLwfOyKrMRRthdOXOmm+59FPaT+IGeRcqK+uSbW9cC6bWNOc
eH07RJ4IVqDs65ctLp3waeV5WFZ7cpdUN8vtwBrhzlZiwLN7k98/ibHzsXGndjxcrRhz0qPzNcGc+u2mcn
LofDYG9MrnOzzYpRT9nXqakeDKUrshSk2ILjb7dzKrqy7Z+Zc4NPSgnoD6IJsWF9VkoWNyr/lk2UgAfNx4
W5E7U7pxgpej5DV9OT9Hha2prhv0QH0gfPyKrMVRthdOXOmpMqvdsqV5C9qXr8EBscWuaqk6knPOTI
raHW5rJk7I8+93NZ2ys8Ndm9Sa7r5gU1Obri7EIPXKR0RWTkvC4y6n1aXO+e55Xi23UxnO+Sk0nFLsgLg
S+H2ysoam9uL1tlfkPMrcepx36ruBG535OTZCuzy4Q44LYpqkTe2ly/ifNxaOu+6PmTj0S7p9+X3KHX7T
Uep9PT23x2ArBJ4IKUaluMmn1ej+w58QlOKp6JNNv0YFoiLsa2FTt8+bnHl88GksqUcCnv87WvxArDeot/
AhhiWWVqcr5qxrrfsexjbjfjNXOcjQ3yuOoCnHvV9fXFNLLeuTyuVzumrlpVr3WvUI8Nq3BW9cuG10747G
mLKTzL05NpPhbArBBHzM26f8AyOFUXpa9T3dmHkQi5udU+OfQC4EOW7XnVwrm4tx+PHqc5FNmLF
WwunJp87YH0AeRfVFS+K2ZZXnOvVGupvl77IDYhz7qXTW7I5MnZhmXc3IkuOCrv5mv7gUghrxLla1Z
O+anJbWn206rra8ax3RFVx2bXcCsEFOPZkV+bZdNOXZL0Ncad tcbI39TVfk13QFQPn1Rnl7ssucFv2Ypm
mJZON88ec+vW4yAsB83HhbkTtTuGEZej5O4OzEy4Vym51z7bAqtyI12wrabc+zNT5uTR05VUfMm+t9
2+3PofQqh5dah1OWvv9wPL7VTU5tNpfA8dnVjuyPG4dS/oTel1bqlZ1y40unfHc6oq8vDILrILqr3pvhcAd
4NkrMdSm9vb5KD5uHjzup5tlGcfCibYUpwvsonJyUe2BYCBYll5E4+Y66occPue1ynjZUA/Mdlc+232Au
AIlpSuzXTK11wiuNPWwLzJyu+kpKK8rXL9TGqq+jlsjJ2Uvvt9g5S/aaJ1Pp6e2+OwFYJrcXqlKbvsinzpPhE+
HGdm5Rsm6oPu33A+iCG2c8jKITGzy64d2n3PG5YI9erXOub003vQF4BNkQvssUYy8ur1knyBSD51m8S
2uVdznFvUot7NPEJTjOhVyabb9fkBaCC/Gsqqdvnzc48vk0syPwY2/wA8kl+IFZLgWzthNzl1alpHEMS1w
U3fNWNb78Dwv93Zv/sBaDyclCEpPtFbIKarctO2dsorfsqIG0bZ/tGVfV7CjvRUfPx4zh4i4zl1NR7/ABNbq7
rbn1TddK7afcCsuyil0zri031vS0SwlHy4Qja7K5+je9HGbT031+3J9cn3fbldgPpAwcLKsdxbnPOcmTyxp
RrcpZMlalvXV/YC8GGFbK7HUpcyT02aXWKqqU3/KgOwQU0WZMPNtumnLsl6HeLZZG6eNbJyaW1IBk2
2yyY0VS6drbZTTCUK1Gc+uXxPn/AEf/AM/y/Mn231b5PpQj0QUdt6WtsDPlvj1qck2m9cDlm4405xemlt
EniVWo+Z1ye5JdO+Fwazq8rCsXXKW1v2n2A2xJynjQlj7bXLNT5+NjWXY8XK6UY/yqJrgWTfm12S6vLf dg
Vg+fDqzZylK1wrT0kno7x5zpy/IIPzlyW4tgWgEC8zMyJrzJQqhxx6gXg+dZCjyKph5spQclrb+02vJelUxUm
k1yt/MCsE+fJxxZOlae1yjO/IlVh1dL9uaST/AAAsBBLHnCp2Ljk7Etv2uGe23ys8O8zbUtpNr5gXA4obdFbf
Lcv+RLgWPylZzk2ovfL+wC0Hz6ozy92WXOC37MUzTEsnG+ePOfX0rcZAWEuBbO2NnXlepaRUfMxKZ3
eYIY4QT516sD6Yla3Zi5capTc659t+h7jylXm2Uzk2nzHbAtBG5St8R6VJqFa5SfcrkuqLW2trW0Bn9lj9J8nT
3rezLJtnDLphGWoyfK+JMsF/AM91+ZPtqv3yd5sZLlojB+1rSba+gD5+RRZj1+bG+bknztm9+S68ONi96aW
vsApBB9Gn5fX9Jl5ut+9x8ijCud9CIL3k9MDcGGc3HEm4tp8cr5neM28ett7bigNAR4E5SV3U3LUuNsxo18
xyIzfKMt8RT0B9IE+LG+tzhc+qK92TZPDqzZylK1wrT0knoD6Alsec6cvyJT8yMluLZmo235d1cbZRinzyB9E
HNUPLrjdqcterOgAAAAAAAGk00+zA+fbasy7y1NQpj3betllcqoxUISjpcJjm0HG/wDx/wD9mdQw6ISUo
w01yuWBp5kPM8vqXWvQ6OPJrVrt6fbfqdgRWfxav7v6MWfxWv7v6MqdNbtVrj7a4T2HTB2q1x9tLSe
wJ/E/qy+8v1Koe5H5HntULo9Ni2t77naWlpAS+Jfvf/ZHVk4zwZdMk/Y9PkbzhGyDjNbi+6M4Y1VcJQjh2
Zd+e4HGB9Tr/H82ZYf1zJ+f6liddcaoKEFqk7I8hTCE5TjHUpd3vuBLgfvsn73+TxfRszqc4+XNPnnTK66YVyl
KEdOT2+e5xzi02y6pw5+KegJ8Oco5M6Y2OytLaZccVU10rVcVE7Ai8O96/73+Rb/Favu/5Kq6YVOXRHXU
9vkOmErVa4+2lPYGHix1X/2RxmzlHFQsbSlpNr5FdtULY9Ni2u/cSrHOHRKKcfgwlq80vHbi1KTXD6tvZ
7H+Efg/wAyIOFjx3qvvx2ztUVqp1KPsP02wJlf4TH5L8yvH+rVfcX5HrprdXIOPsfDZ1GKjFRitJLSAj8M/d2f
eLTiqmFKarjpN7fj2BFg/Wcn736s8skq/Flym9Jx7srhTCuUpQjpye3z3FtFdySsjvXYCPxO2Eq4wjJOXVvhn
XiHej73+Db6Fj9PT5f292aWU129PXHfTyuQOcr6rZ91kcoOfhUGv5Xv+7PoTipxcZLafDPIVxrgoRWor0Az
qyKpUKbnFaXO32PKr/pFM5KDiUUt+oeFjuXV5f93o3iIFRSSXZICPwycflcepdxU+DzE+vZh/AFKIYtMLf
MjDUvmdQphCcpxjqUu733AjwZxosspai98b9RmTV99VVb6mnttehXbj1XfvIJv4irHqp/dwSb9QJrv4pT9
3/J34l9Vf3kbyphK1WOPtxWk9ntlcLYdM1tfDYCr91D7qJfEpyjCuKbjGT9posSUUkuy4ObK42R6ZxTXwAg
yK8SvHfQ1KbXGpbZ7KDn4VHXPTz/cpjhY8d6r78ctmtdcaoKEFqk9AMlqMiuWPGTnFaXO32M52fS8S3o
g0I2+31NJYWPXKXU61v7GzaMVGKjFJJdkgJ8O+t40U5JOK002FcsgFOIREkmIL4nU8OicuqVfL76ajaEl1xUY
JJL0QHzcSrGsg1bxYnym9FGNDGje1S9zS+O0a24tNsuqcOfiuDuqmulariogR4FkIW3RIJJuXG/UXzV+dTCt
9XQ9to8w6a7ncrlp6lx/csqrp35cdN+oE2a1HLx5N6W+/4lqaa2mmvsOLaa7opWR3rse11xqgoQWorsB

j4h9Tn+H5iuUZYOoyTar50/sN5RU4uMltPujOvGqqUICOIJafIGXhv1VfNnFH8Tu+7/grrrhVDpgtL4HkaYRtlYo6nLu9gfOppp+kWV5HD37Lb0bxqw4XwjF7nva09INtFV37yCbXqeVY1VL3CGn8e4GpHZKi++VV0OmUe0m9bLDO3Hqu/eQTa9QIof8Aj5kK6bHOEu8d70aP+Kx+7+hRVjVUvcIJP49zryYeb5vT7etb2BJmXebYsaEktv2pNINPk1wjXCePt7nMsOicnKVe23t8sRw6lyUo16ae1ywI3VUs6yN/Ck9xe9GsqcKE4x3uTfGpbK7aa7lqyKlo5qxaapdUlC/F8ganz8hxszlXdLpqS4W9Jn0DO2iu7XmR3rswPn5cceEoRp1vfLT2b5/7/G+9+qNvoeP09PlrW992aTprscXOO3Dt2A4zPqlnyJLYOXhdbX8r2/7n0JwjZBxktxfdHkIRhBQivZXGgM4ZVTpU3OK45W+TDwvmuz7xssLHUury18t8GIVMKU1XHSb2+QF0XOmcV3cWkTeH3QVHlykoi3w3osMbMSmyXVKHPrp62BNVONnicpQe1062c+xdm2RyJaUelxb0i2GPVXPrhBKWtcHluNVc9zht/HsBDLyI51Ko1pNbae0belvVuO32Unt/0N1iULp1WIova5Z3bVC6PTZHaAwzrXHF6q5d3ptP0MfLw4Y/U2pS1/wBud/lthTXCry1H2Pg+TOOFjxitVrf2vYGfhn1Z/eZtlwdmNOK5ejqqmFMXGuOk3vudgS4V9bxopyScVppszpl5/iM7le5Fa2UWYdFkuqUOX309GldcKo9MIqK+wCR/xVfd/QtOPJh5vm9Pt61vZ2BH4p9Wj99fkzS+UZ4U3GSa6fQ2shGyDjNbi/Q4hjVQrlCMfZl3WwOcL6pX8v1MMNbyMpfGX6sthCncFCC1FdckwphXKUoR05Pb57gfOxaaJOdd/Fifq9FFNeJDJu3uxcrT2je3Gque5wTfx7HtVfdP7uKW/UDQgw5xpyLqptRbe1v1LzO3Hque7Iba9ewEmVZCebQoyT6ZLevmdZbUM6icul9tnF9NdOVjKuKiur9UW21Quj02RukBL4hdW8foU05NrhpZnmQ3iY89bUUt/ikVLCx0mvL3v7WbKEVDo17OtawfAiVGE6+vqXT94XRr/ZsnRvo3vn5m30HH6t+X/dm/THp6dLp1rWuAMKciqOLBucVqKTW+SfbI7MS+K7y2l/QpWFjqXV5f93o0qphSmq49Kb2+QPh4IWNZBq3ixPIN6KMaGNG9qj7ml8do1txabZdU4c/FcHdVNdk1XFRA7Pn+H2wg7ISkotava36n0DB4dDi15a5e+7An tmr/ABCqNb6IDltHWenXZVkrXuvT/wB/qU1UV0p+XHW+7Jc2+NqePWnKblp8dgO/DotwndL3rJbkzmuCrrjBdorR0BC5KHiu5NJOPr8j3K5zcfXx/UouxqrmnZhbXrs9dFbICTjzD3eeewGfiH1Of4fmT5UHLw6mSXupb/oXWVxtg4TW4v0PYwjGCgl7KWtARQowpVqe9L13Lsb4ipVTdG+lv1PHg47lv/7s3jFRioxSSXogMc5N4lmvsf8Ac5x76o4kG5xXTHT5KWk1p8pmH0LH6ury/wANvQGHh0vYukk3zvQjHFy05v8A45evOmV1U1078uPTvlnFmJRZLqlDI+qeqj8KcpTtp63OCXEjHFpok5138WJ+r0fSrqhVHprico4txqrnucE38ewGFNeJDJUa3uxcrT2jzE+vZH/AFKaqlK6f3cUt+p7CmEJynGOpS7vfcD3zleZ5fUuvvo6OPJr87zen2/idgAAAAAAAAAAABpNptJtAAAACAAAAAAAHihFScIJFju9HoAAAAAAAAAAAAAAAAAAAAAA4ndXW9TnGL+DZ1GUZrcZKS+KYHoAAAHMblTbUZJtd9PsB0AYW5INU+mUm2u+l2A3AXK2AAAAAAAzu6CuVTfttbS0BoAAAAAAAAAAAAAAAHFt0KUnN629LgDsAAAAAAAAAAOI3QnbKtP2o9+AOwAAAAAAAcuyEZKLklJ9lvudAAAAAPJSUluT7JbYHoOarl2wU4PaZ0AAAAAAANqKbb0l3ZN9nj5fmeXLy+rp6v1ApATUkmntPswAAAAAAAAAAAAAAEt9L6e+uAAJmfCilKV6U5t9+5xjVZ9Idfua7fAOvpd+nG2UNf9X3J8PePkyx5JPfKku7AuAAGeRZ5NE5+qXHzPn4vVj31Sm/ZuX+/p/U18StXVXU3xvqlozy8mm6qMYKSIF7XAHOlbPKrculy16LufPw7UrZt1yk5y7pdi/Hs82mE/iufmTeHd7vvAZWZTeZXNQsSS9z1f f0LndGNKtnuK12fcmv/AlpT93/J54m25UxS2m3x8QO14hDhyrmoPtLRtbfGqnzeZRetaj7J5FlUq3i6Twu/YztjOHhijYmmpevzAoqzYXWqEly59Teyfl1uXS5a9F3OceKjRWktLpRoB86jKaybG42SUnxH4HWRZGrxKE59IH/J1i/Xsj/fU8tSl4rWmtrp/wAgdxz4OaUoTgn2bRRddCmHVn6X5k3ii/8AHi//AJHWVOqNVbth1y/lQHn7QhtOVc4xfaTR1I39FLUYyfXF6IHsjHJnkTx5ddMYw+O+Uax/hn/8bA8wL3OuNbjNtJvqfZ8lZPgfU4fj+zvY3GuTXdJgT25sIWOEYysku/T6GmpkvvT6dqS7p9yHCndCuTrp69vmWzWmFzzlbKroTWnyBpLPjro cXGXVF60l3OqcyFtnluMoS+DMcNJ5mQ9cqT1/VnuTx4hQ13YFGRkwx0urbb7JGL8QhFe3XOL9E13OWIPxXuu0Y8f0OvEknjb1yplDW7jjTVGcotqXbRrOShCUn2itkOf9Up/D8iy/6vZ91/kB5TdTg6rzEmI9phLPht9EJziu8kuDGLcfCza+Ov7nWPZfXRGMMFcdb3vuBXXfc2p2Qe0u69UKlo319cU0t65JcWuyCvc4dEZLaX9Tvw36r/wCzA2jkRlkSpSfVFb36HN+XCmfRpzn8lmNX8Vt+7/gxxp2q62cKvMk3y99gLKMqF0nDTjNfyyNyBxvsy6rXR0aem0/QvA+bZIN5lc1CxJL3PV9vx CfVTTPTjt70/Q6v/iIP3f8jxPmFa/+QHUVeIJ7Vc3D/topjZGVasT9nW9nGRFLFsSS0oPSInJrwM0v6/uBu/EIbjjXOUV3kkUV2wtr64v2fyJKbL4UxjHG3HXx7nuHTYqrq7IuCn2/EDp+IQ2+iuc4ru0iim6F8OqD49V8C0t5GHDodPXBPvE7qnT9Fusx04vTbXwYHdmdCM3CEJWN d+IGIGRDIi3De13T7oiw53V0/8AHR1Jv3t9zXGhb9MIZKrojJcrYFpPdmQrn0RjKc/VR9Ch9j5eHzbHrnCnzHJy82bdRIQvbik4yXeLMMF+i3/I5hG+ebC2VPQuz5Osf+i3/IDA/LhTPo05z+ERRIQuk4acZr+WRHjTtV1s4VeZJvI77GjjfZl1Wujo09Np+gG9mbXVbKuSluPwXc5hn1ufTOMq/vHEEn4tZtdo8f0R54nFPytr+bQHb8Q

gufLs6P8AtopU4uvrTXTrezjKS+i2LXCiyOUmvCY69Xp/1A2fiENtxrnKK7ySN1fCVDt19xSbJabL4UxjHG3HX
x7nlfdleNkqcHFOLaX4MDT9oVuK6ITk/vVdjSrKhbVKcVLce8fU48OSWKmly29nGLxn5CXYDKvKazJzcLG
mvc9V2PpEVp8AFLvu/wCC0Ce6VsY6lODdj91/A9uy66beiafbezHj/iNHyPLUpeK1pra6f8gdxz4OaUoTgn
2bRWR+KL/x4v8A+RXD3I/ICPxC9xg64xmnw+pdjqN3m4VicZLpr7y9eD3xH6o/mjuX1B//AKv0AlxcuNW
PGChKclttJdizHyIZEW47TxMy8OSWKmly29nGLxnZCQGt2ZCufRGMpz9VH0OqMqF7cUnGS7xZdh2W
x65wp8xyfMtmsl3zzYWyp6F2fIF5hfwlwpkoacpv0RuQ4aUs3InL3k9L+oGiz69qLhNSb1po1svjXdcptz7M
n8QS82h656v8DK+vY/++oFV1qpqdkk2I8ArY+SrW+mLW+TLxD6nP8PzJstteH0Jdmlv+gG37Qh38uzo/w
C2jeV8IOO5e1HW+CbzMjy/L+i+zrWtnEYWV+G2xsi098b/AAAuqmrK4zS0pLfJxRkRulNRTXQ9PYxPqtf3S
XEbisuS7rbX9wNrM2EbHCEZWNd+ljdSjlhk90013T7oiwp3V1Py6OtN8y2aY8LvprtIV0KS55AuMqciN0
pximuh6ezUi8P/fZH3v8AIFE8iMMiFLT3jbTOb8qFDUWnKT9EY3/xOn5f5Or6bYZKyKUpPWnFgd05kLLP
LIGUJvspCyVX02uMoN2a4l6LuZRuruvgr6pV2p+zsXfxSn7v+QNbcylum1wmn23s5jnwdijKE4dXZtGc4qXi
sdreo7/se+KL/AIIP16v0AquthTBzm9L8ydelQ1uVc4r0eu5xne3djwl7rfP9inJini2JpaUWB1RarqlOKaT+j1
KShFyk9Jd2T+H/AFSHzf5muS61RJ2rcPVfEDB+IQ5ca7JRXeWeiF0LKvMg9xJq7b5VpVYyVeuNy9Dnw/6p
b83+QHa8QrcV0wnKX/VI1x8mGRvpTUI3TMfDlpY7euXLlnIXHilqX/X/AABaYX5cKZ9GnOfwibnzMadqut
nCrzJN8vfYCjKhdJw04zX8sjyzNrqtIXS3FenqYON9mXVa6OjT02n6HsEn4tPa3qO1/RAa150J2KEoSg32
6jW+6FEOqb+SXqTeJd6X69Qy1151Eje73A6/aEEtyrnFeja7mryYrG8/T6fh69zzNSeJZtdkTz/hC+S/MC2u
asrjNcKS2cUXxv6ulNd1ye4v1ar7q18vNU13WwN7M6EZuElSsa79KNKMiGRFuG9run3RFhzurp/46O
pN+9vua40LfprkrJVdEZLlbAoovjf19Ka6Xp7EsiMcINon1SW9+hP4d3u+8Lfl4rV93/IFF+TCjXVtyfaK7mdW
bCdihKMoSfbqXcmIKf7RnKNfmSiuvFvse5Kyb1H/x+lx00wNs+Tl5dEe9j5+RT5Ufk8rXs61oku58Tp3/ANf
8IoEf82lZTLvW+Cwix+PErku2v8ABAaaaaaaaaaaaaAPJRUouL7NaPQBDXDKxdwhBWQ3tPZpj
0WefK+/Sk1pRXoVADJO/wCktOK8nXD9TUACWimz6XZdbHW+i87KXFSTTXD4PQBLhVWU9cJr2d7i99z
3CpnU7euOuqW1yUgCTLqt+kV3UxUnFa0dX0Syal9WoWrlfyUgCPrzuno8uO+3Vs6yKbZ4ar312cbfYqA
HNScaoJ90kmddAARou+nLnOuCnGf2nc6ZvxCFqj7CjpvfzKgBPnVTuoUa1t9W+5zIY87IVyr111+hUAILE5
mRW4ShGC9dPuUV1SeF5U10y6WjcASYUb615VlaUFvUtlfcACGNWRizl5MVZXJ7032NqFkysc7mox1pQ
RQAjCwmddeRfKUdRIlae+/LPb6Zzy6ZxjuMe72UgCXKosdsb6NdceGn6k+ZLjnTu2EYQT7b5Z9lyyqXfT0j
pc72wMr6HfhwjH3kk1/Qzl9NtrdbjHjTlvuWwj0wjH4LR6BNj47+h+TatN7Mq1l48fLjCNkV7r2XADHHh
d0yd8k3L+VehNVDKxuquuuM4t7TbLwBHjU3xy52Wpe1Hun8hKi6i+VmOIKM+8WWACar6VZcpWJVw
X8q52UgASZdVv0iu6mKk4rWhlVXX1Vex7Se5LfYrAHF8XOiyMVTuLSMKsZvB8mxL5/DkqAEUPpIEPLVc
ZpcKWzauq2VE43T9qXw/INwBFD6ZTHy1XGxLtLzpiY0q4WO3TlZ3SKQBDCvJxW41RVlbe1t9jfHWQ5yn
e0k+0F6G4AEXk341spUJThLnpb7FoAmp+kzt67dQgl7q9RTTOObbZKOoyXD2UgCOVF1F8rMdKUZ94s7
q+IWXKVlVcF/KudlIAhTNeIttcfYcdJ7+R7m0zt8vojvplt8IIA4vi50TjFbbi0jGrHbwVTYul8/hyUgCKH0yiHI
quMOuFLZrGu54tkbZdU5J6Xw47FAAxw65VY8YTwpLfBxRTOGzdZKOoy7PZSAIrK76syV1UFNSWu5au3
IAE11M551Vky7jHu9nk6ZvxCFqj7CjpvfzKgBPnVTuoUa1t9W+5vFaik/RHoAxy6ndjyhH3u6M6lfLfsrsrU
WodMee/BUAMcOuVWPGE1qS3wcUUzhmXWSjqMuz2UgCLyb8a2UqEpwlz0t9jsn6TO3rt1CCXur1KQ
AI7abqsh3Y6UIL3ossAhzMI3ytpldFRXVxFWZRY4WVa64Pt8TrJx3fKtqSXQ9m4EF0czlrcXXGK+G+Wby
xvNw4VS4IFL8GUACKMs2EOjy4ya4Utmypsnirtn1Tku/wNwBDV9MqrVua4vXck2d4VFIxmq1e8+++/c
rAEMKsnFIJVRVlbe0m+xtjrldjnc1GLWIBFAAEPIZGPkWTpgpxm962XACGNOTPLrttitL4Psd21X15Dup9p
PvFsrAEXIZGTfCd0FXGD2tPlndtM5Z9dij7EY6b38yoAT0Mf7QVvT7HTrexn1TupjGuO2pb7lIAny8d3Vx6
XqcOuYXTy5USU4RhFL2pb7lWRQr4KPU4tPaaMJYl1i6bchuHwS7gaeH/U4fj+Z3k1efRKcem+xpCCRgox
WklpHoEMPPqgqlCMUlr32R3h02VU2QnHTfbnvwVgCfBqnTR02LT3vucwpmvELLXH2HHSe/kVAARyo
uovlZjpSjPvFlgAmq+IWXKVlVcF/KudnkZrxCdrj7DjpPf2lqAE2bTO1V9Ed6lt8jMx5W9M63qyD4+0pAHz
8ieXPHkp1xhFL2nvubVV+d4dGvetx/U3vrdtMoJ66l3FFbqjBvbiu4Elf02FaqvceOFjs7wqj1wsjbH3n8e5
WAIYV5OK3GqKsrB2tvsb46yHOU72kn2gvQ3AEKqyMa6x0wU4Te+/YV05EsyF1sVrXOn278FwAkyKLfer
6NdfrF+oTzLzxtiqop8vvsrAEenXbTelxF6Zx1Lp6t8a3s8shGyDhJbTJvo13keT5i6Orw/XpA8wE52XXvtJ
6RYc1wjXBQitJHQAA
AA
AAAAAAABlm3WRnXVU9Sm+5tRXOuLVljm99wNAAAAAAAZfSI/Sfj0963sXSuUoeVFNN+1v0A1AA

8IWvW6/udUW3wohGGNuOu++5tTjt4Pk2rTe/w5M6/pIEfLVcbIr3XsDnHrsrqyOuDgpJtL+pt4d9UXzZ3RC11yV8tuXovQmqjl4ydcK4zjvaewOqpP4ld8v8GGJZapWThT5kpPl77FGJRdDJnZal7S7pnjpvx7pTx0pwI3i2Byo32ZldsqejXD5Lyan6VO5Ts1XBfyrr1FF07Mq609whwuAKWtrT7HzK7Po+PkVN8xel+PB9MivxJWzsZpew9OT2BxD6Jk1p8KyHPzO8Lbrvv9ZN6NM6iV1S6FucXwa49fIY8INcpc/MCDCndCuTrp69vmWzWmFzzlbKroTWnyl1ZGLOXkxVlcnvTfY2oWTKxzuaJHWIBAUeeJ/Rcnli+zXUv9/EvlS7FndZCUFv0lz2Awqg6JY97/AJ21L8SjGXm5t1vovZX+/ga5VHmYzrguVrpGFS6aEprUm9sDHE+vZHz/AFGF9byfvfqzTHpnDKunKOoy7Pfc8xaZ15F8pR1GUtp778sDjA/f5P3v1ZaS4IM67b5TjpSltc9+5UBF4al/yvXPVrYyUo+IUSj3lwzimrLx3KUIJpvmlf8Ac0pousyVfkJR6fdigMZSn+0ZyjX5korhb7HuSsm9R/8AH6XF7TTNsiiXq+jXX6xfqE8y2cU4q qKfL77ArXYiu/iIP3f8lpLI02SthdTpzjxpgPEvqv/ALIZFDyMSEYvUkk1v14Mb4ZmRDuq4xSfup9ze+idIEoiX TOC+PcDKeRJQ6MvHfS+G12OsxQWAvgL9zjRzZ9MurdTqjFPhy2d3Y8/oMaYe1JaAzzvqFPzX5FcoxWM4pLXR2/AwyqLLMSuEI7IFra39hTJN1OK79OgPn1ya8Knr46/uizDio4tevBbZnjY8lhyqtXS5NmDsKl+XGEZ xXuy2B7hpLMMyElpj/qWkHh6l9lv6nuS7v7dl4EULdWTWJR1c+1JvucVOb8STSioyceUn9h7CrJxZyjVCM4Se1s9qpyFmRttSaa5afb7AFST8Ut2t6jf2PfEP3uO/Xq/wd1Uzjn2WOPsSjpPfyPcymds6XCO1GW3z8gKSP xT6vH7/wCJLCbPqndTGNcdtS33A6y/qc/unuF9Ur+R3ZX5IMoPja0SVLMph5Ua4tLtJsD3B+sZP3v1Y8NS/wCWWuerWzrBotqna7V7zXO+/c6waZ1Rs8yOty2uQM5ceLR16x5/oxirqz75S7xekaSpm/EIWqPsK0m9/M5totryHfjpPq96LA2ux4XShKW04PaaJ8v67j/P9T1V5GRbGVyVci86T7nWbTZKddtS6pQfYDzxP6r/AO yM87mGPD+WT5/saZMLr8SKcNWb247NL8fz8eMN6lHWmBs4RcOjS6da0fPxX/4eThe0k9f0NHLNcPL8 uO+3Xs0rxnVhzrjzOUxv7XoDzw5JYqaXLb2cYvGfkJdfDrIVjhNakt8HFFM4Zl1ko6jLs9gUkVP8Uu+7/gtr K76syV1UFNSWu4HmQILxOInbXT/k78SSeL8pl9spslnVWqPsqOm99u53m1Ttx+mC29rjYHGRKqOLW7 o9fC0vi9Gd1mRPHn1URjDXq+uazOPO3HrUeJw1wZzWbdW4ShGC1y99/sA6o/hb+5L9Tvw76pH5v8z3 HpmsLyprpemv6nGFG+r/inWIBb9rYFZ86uz6K8qvtrmP+/ij6JFl4krsmEor2XpS5Ayog8a6iT7WR0/8Af6G 2F/yXXX/F6Xy/3Rp0u6jUF7UXtHeJU6ceMGtS7sCfA/f5P3v1Y8096/73+TvEpnXbfKcdKUtrnv3PcKmd UrXOOuqW1yBn4b7tv3iyXuv5E+DTOqNnmR1uW1yUvIMCLwuK8iUtc9Wt/gZSIP9ozIGvzJRXC32KcCqd NMo2R03Lff7Ec5FFivV9Guvi/UDHJWTeo/+P0uL2mm5KUs+hNenY7TzLzxTiqop8vvs6upnLNqsjHcYrl 7A88RSeK/saNseKWNWtcOKOM2uVuO4wW5bXBrTFxphF8NRSYEGJJxwL2vTf5Fhh8VHei13ltv+pzhUS rpnC2Oup9t+mjiEMrF3CuKshva57AJJV+Kx6eOuPP9/8AB7H/AMfxGSfEbVv8f92d42PZ5zv11vsl6DPol dCLrW5xf1wBzgp2WXZDXvPS/3+hPh2Wx65wp8xyfMtn0MevyqlQ9UufmTeTfjWylQIOEuelvsBzCN88 2FsqehdnyXk1P0mdvXbqEEvdXqUgAAAAAAAAAAAAAAAAAAAAAAKqoVJqC1t7Z2AAAAAAAAAAAAAAA AAAAAMcuh5FsgmlzvK2AhkVqKXwR6AAAAAAAAAAOKqoVJqC1t7Z2AAAAAAAAAAAAAAA AAAA8IfSi4vemtcEixL4LoryGofBrsWADLGx448Gk9t938TUAAAAAAAAAAAAAAA AAAA8IfSi4vemtcEixL4LoryGofBrsWADLGx448Gk9t938TUAAAAAAAAAAAAAAA AAAA8IfSi4vemtcEixL4LoryGofBrsWADLGx448Gk9t938TUAAAAAAAAAAAAAAA AAAA8IfSi4vemtcEixL4LoryGofBrsWADLGx448Gk9t938TUAAAAAAAAAAAAAAA AAAA8IfSi4vemtcEixL4LoryGofBrsWADLGx448Gk9t938TUAAAAAAAAAAAAAAA AlrrJ35TojZ5clr2mvU4l1YdsHG1zrk9NN70BtfbOOBtBS1GS5RURZP8Ro+r7IWznkRx659C7ykBYD59sZYftZxc5x3qu WzrOnPzaPKk4uXbn5AXAgyMeyip2xvm5R77Z1CizJrVtlouXKUeyAtBjg3SdVite3W+/2GVSnmOVllzhHeoxTA+gCPFsnDIIjzn1pLcZGVMbb77oq2UYKXOnz3YH0T3lJxBcptz7MlrU8bNjV5kpQmvU4yaOnKqj5 k31vu3259APpAjy5TxsSMYSk3vXU+5x9Gn5fmUZEpz+fDAvObZ+XVKffpWxU5uuLsWpa5RjnV9dDI1Sxm9J9/mBliLlu6bp26jv3ddy0j8Pp1VC3rlzv2d8dzCiu3Issj50owUueQPpgivlZ5leLVNp69qXqc3UW40PNq um+nupeoF4lsu5zwYWQbj1NdmePFtIT5kr5+Zra54AuB8/HjbmV7sulGMfzsJ6/M7xJTrtryMeU3NJbTYF pksiLyZUae4rezSceqDjtra1teh82OPvPnV5k+Fvq3z2QFOTbOGXTCMtRk+v8SogzIL6TQoP2uybPMiizHr8 2N83JPnbA+gCa/JdeHGxe9NLX2GP0af19f0mXm6373HyAvBhhXO+hSl7yemdZVvk0Smu/ZAaghqxbbK1 ZK+anLlafY98PIOUrIzJyaa9fmBaCTw6UpQs6pN+16s8IKX7ThHqfT09t8dmBYCK6yd+U612eXCK9pr1Ojd WHbBxtc65PTTe9AfQBHOuv2nCPU+np7b47M68SIKOOnFtPqXZ/MCoEuXfKqmEYe/PhMysx511OyOTJ 2RW3zwwLwZY1vnURm+77nOZe6KNx95vSA3BEsO2UoqV81Y1vvwvjFvnBzGfvw4bAqB87Eqtyatzum oJ6ST5ZpiSnXIWy85uSS2m/8AftAtBBKUsrJnBWOUqHH7iMpYuVCHmOyufHL3oC8IVs/2i6+r2Oneio+ ddCc/EnGEuhctv7NAfRB8+6uzD6bYwylHepKTOsiyVWTTCpPy591vgC4EmfOW66q21Kb9CuK1FL4AAR 5Vlk74Y9UulymTOLgLcaHm1XtfT3UvUC8EWXc54MLinx6muZPhj2zp8vd8uvW0l2QFwJcXlbwnZPlw2v

mY1VzyY+bbe477JPWgPoAkwrZ9dINkupw7S+KKwAAAAAAAAAAAAAAAAAAAAAD5s661nzjfxGXMX
vRpOjChKKb25PSSlsrtpruWrlqSOK8WmqXVCHPx2Bhk/xGj5GeVCCz07k/Lmu5dKmErI2SjuUez2e2Vw
tj0zipICOyjCrScn3+EjtLSWRipdk1r+qN4YdFcugMOV229mk6YWTjKUdyg9p77AZ531Oz8PzOsT6rX907s
hGyDhNbi+6PYRUIqMVpLhICHDj1rKivV6/MyxKseyLjdxZF9m9H0a6YVOThHXU9vk5txabZdU4c/FcAY48
MWORqp7ml8do5wP3+T979VVVU10rVcUtiumFcpShHTk9vnUBLkfxKj5f5Ga1HLx5N6W+/wCJVkmErI
2005x7PYtpruILHeuwGWVfXCuPXHrhN6bXYmvpoqrdfvTL0SI3Lo01xq8pR9j4PkzWFjqW/L/q2B3jTIZ
jwnL3muTzK+q2fdZr2PJxU4uMltPhgYYH1Ov8fzI4d71/3v8ldcl1wUILUV2R5XTCPycl66nt8gSXNUelwsI
xCS1s0zb61jSSkm5LSSZRZXCYPTOKvtMoYdFcugMOV229g5EHX4ZXGXfe/zLI+4X3f0PbaoXR6bFtd+5
1pdPT6a0BJ4X9Wf3n+SOYfxWz7v6lqqqhTHprjpB33CpgrXao+21pvYHZCpKHs3Jpjx438kXGVuNVdJSn
HbXrvQE+V9dx/n+pr4h9Tn+H5msqa5ShJx5h7vPY9srjbBwmtxfoBDIQcvDqZJe6lv+h1CjClWp70vXcuxb
GEYwUEvZS1oweDjuW/L/uwPcRUqpujfS36nniEHPFlr05N4xUYqMUKl6l9Awx8ip48ZOcVpaab7GHhzU
p3tdnJP8zd4WO5dXlr+r0Y+HJKzIS4XV/kDnw+ca/NhOSi1L1egri2eKRCtHJa2vkyrmzEptl1Thz6tPWz2GN
TXNTbBKSWh0wlP11rPnG/iMuYvejSdGFCUU3tyekILZXbTXctWRUkcV4tNUuqEOfi3sCe+Sr8TrIJ609/6j
xK2uVMYRknLq3wyu2mu5JWR3rsZrCx1HXI/3YE/iEP+OmbTcY8SPflwvL6+pdP3i1xTj0tJrtpmHOHH6t+X
/dgd4yrVEfk30PlbMfEoOVCKlppe2VpjLSWkg1taYGUcmqVXmdaS1zziXCTIHlt1pT3r+5Q8LHct+X/AHZ
soxUelJKotaQEvhn1Z/eZzV/Fbfu/4K6qoUx6a1pd+54qYK12qPttab2B86FVX0u2vl429xe9G6pw4XQinu
bfGnsptoru15kE9ep5VjUVvcIafx7gakE7I1+KNzek1rf4F5nPHqnNzIBOTWnsCbxC6EqVXCSIKTCezTjp6s
Hp/mhFNfgd14INUuqEOfi3vR5k5UKE0+ZtbS0BLht5GSrJf8A04Jfj/uz6JPgVOrHXUtSlyygCG5qjxGFkuIS
WtmmbfWsaSuk3JaSTKLK4WR6ZxUl9plDDorl1Rhyu23sCXlg6/DK4y77T/MuX7hfd/Q9tqhbHpsW137n
WI069NaA+fiwdnh1sV3bev6I5xKcW2r23qa77lovqqhTHprjpB33OLMSiyXVKHL9U9AZ4sMaNs/IbcktN7
4KIWQsTcJKWu+hXVCqPTCKijyqmulNVx6d8sDsAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAA
AAAAAAAAAAAAAAAAAAAAAAkI2SwwAAAAAAAAAAAAAB44RbTcU2
uyaPQAAA
AAAAAAAAAAAAAAAAAAAAAAAcSuqhJxIZFNejYjdVOXTGyLb9EwOwAABz1wu1B
yXU/Tfj0AAAAA8nJQg5S7JbYHoOa7I21qcHuLOgAOY2Qm2oyTce+n2OgAAAAHEroQtjW37Uu3AHYAA
AAAAAAAAAAAAAA5ssjVBym9JHFGTx5KG+O+0BqAAAAAAACzshWtzkortywOgAAAAAAAY3
ZVVUMmcva+CR1TfXem623rvtaAaaaaaaaaaaaaAcxshNtRkm499PsB0Diq6FrkoPfs9PgVX
QuTcHvT0+AOwAAAOKroWuSg99L0+AOwAAAAAAzsuhXOEJpmb0uANAAABzOyEndUIHfc2z2cowi5S
eku7A9BjTIVxzYN7S3yjYAAcQuhOyUl2o9+AOwAAAAAAAAAAAAAHFtsKYdU3pb0dp7Sa
7MAAAAAAAAAAAAAAAAAAAAAAPnRjTzmXu9rpT0tvRVtj46krKkm16q
WzIYVErjTe59T3y+xioLF8QhGtvomuwF4AAis/i1f3f0Ztdl1029E0+29mNn8Wr+7+jPLupeK1pra6f8gdxz
4OaUoTgn2bRWR+KL/x4v8A+RXD3I/ICLxC9qlrippr2I2O3d5uFZ7Eo6h/Mu/A8T+q/8AsjS76jL7n6ATY
2XGrGhBQlOS3vS7clePkQyI0UNrXdP0M/D0liQaXL3v+pnh8ZeQl23+oGuLKp23eXBxkn7Tfr3PLM6EbH
CEJWNd+IGeB/+yfvfqzyEL8Oc+itWQk98dwKcfJhk90013TNSxFnTbdOcYOFv8yZUBNdwmwrs8uMZTku
6iTu6N+fTKKa1w0/TuaeGJOFlj5k5csXpLxKlpd1z/AHAovylY8U572+yXdmdebCU1CcJvt9upE98p/tLcYd
biuF+B7k/SMiCi8ftT2nsCy++FEOqb79ku7MP2hBLC65x+G13M7k552PGxfyptP48m+ek8Se121+YHUsm
McZXPpuePU1hJThGS7SWyG7+FR+SK8f6vV9xfkB5j5EcilCu1p65M7M6EZuElSsa79KMMNuOHe13W
/yOcOd1dP8Ax0dSb97fcC2jlhkRbhva7p90KL439fSmul6eyfGhb9MlZKrojJcrY8O73feAolkRjkRp0+qS3v
0PL8mFGILbk+0V3MLf4rV93/JI12ftCyUK/MkuEt9gKqsyFlnlyjKen2U13KD52Qsi9wf0fpchtnM+iB87xK3
qca+mSUZd9cP5FMciEceVnlShGOlprRI4l2q+8a+ifU5/h+YG1c1ZXGa4Uls4pyl3Kbimul6ez3F+rVfdRL4f7
t/3gO14hXKPswm5f9UjXHyozDaScZLumYeFpeRKWuerW/wEOPFZ69Y/ogNr8uFM+jTnP4RFOXC2fQ1K
E/hlxwV1ZGROXvdWvI3KLMeFlsLhtSh216gRWZTeZXNQsSS9z1fcoyrKnRCd1cmm+F2aOL/4pT93/I8U/
cQ+9+gFVtkKa+qb0kTrxCHDIXYov+bRznrryMeD91vn+pXZCMq5Qkl0taA8nbGNltXtRS3wT/tCDS6ITk9
bel2MceTfhlfqfo3opwIpYkNLW+WBPfDIh1Q3xw0/Q6sn5cHLTlrx0XckwOMjJS7dX6stA+XRdH6VbOU
pOT4430l0La1kypjDUtbbS4ZhhfWsn736sQ/i0/u/ogNLc6uqyUJRIuPw9tTzUPo/nTTiu2n3J6op+KW7W
9R2v7G+bQ8inpi9ST2t+oGa8Qhw5VzjF9pNDNyOmlqCk+qO1OPZHE8iSh0ZeO+l8Nrsa5CgvD5eX7nTw

AwbnZXGDjPaXvPsze2xVVym02o86Rnh/Va/kayipRcX2a0wOa7Y2UqxcRa3yc4+RHli5RTWnrTII2uvCtpf vKXT/X/Wa0r6HkuEn7Mob/Ff6wKq8iNI064p+x3foZ2Z0IzclQIY136UZYqIHCUu/mnt7M8Od1dP/HR1Jv 3t9wLaMiGRFuG9run3RpJ9MW9b0t6l8aFv0yVkquiMlytloGdF0b6+uKaW9aZzTkunOMd7j8fUkrs+iV Kr7a5j/v4o8og8a6iT7WR0/8Af6AW/SI/SfJSbkltv0Rxlyqdt3lwczJ+0369zPC/5Lrr/i9L5f7o8wP3+T979W A8O96/73+R4b7tv3h4d71/3v8AI8N9237wFpNZnQjNwhCVjXfpRtc3GmbXdRbRP4bFLG2u7b2Btj5ML0 +naa7p9yLFyl0SuTUplSllhJfMtWPCN7uW1JrTXoTeHRXmXy1z1a2BvRlwum4dLjNejPb8mFGILbk+OV3 MLePFatesf8mXXZ+0LJQr8yS4S32AqqzlWWWeXKM0sFZSXc7vyIY8U572+yXdkQsi9wf0fpcHtNM0muvx SKI2jHgDuOdBzUZwnXvs5Lg4zfrWN979UU30wvh0T3re9oj8Qbrsx3Hlx7b9ewFV+TChxi05SI2jHubJ7Sb Wn8D5+BqWRN3b8/wC0+gBPlyqTr82Dlt8a9DHxK32PK6Zd0+r0PfEfeo+9/g68T+rL7yA0w7lyr6Y1yh0p lldzu+3yob6JS3x7KO4e5H5I9I7r+QEHH1715clOTcve7pcHnd8aMy9y223pJLvyA+f/V5ff/RHmKk8+9tcp 8Aa05kLbPLcZQI6Jnd+RDHinPe32S7snzOM3Ha77/UzvIP8AaW4w63FcL8AKK82EpqE4Srb7dSO8jJhjyip p+16r0Jcn6RkQUXj6ae09nuYuq7FU13aTT/ACijKjf1NRcYxW9szfiENvprnKK7ySO89uOJLXrpGFFt8KIRhj bjrvvuBZVbC6CnB7R2SYFdltb64Ock9pFU+nol1e7rn5ATSz4dTVcj2a7uK4Nql4ZEeqG+O6foTU3TcWsX HXI77t9zzA6vpV/UIF+qXzA7fiFabXRJyT1o1sy0VUxsntdS2o+pP4dFeZfLXPVrZxkyn+0Y9MOtxXEQN4Z0 HNRnCde+zkio+fk/SMivpePrnaey6rfIq6ve0t/MDy22FMOub0vzJ14hDa6q5xi+0muDPxByeRTFR6I36fjy e3TyLqnW8bSf29gO/EnvFTXbqRv1xroU5vSSRHkRIHw2uM1qSetf1HiDfk0RS2n6Aa/tCHd1zUP+2je+2+N dHm+9HjsTTsyJ1Ov6LqlLUw5nOE6/C3GxNNS7fiBs/Ela3Gucl6vXY6efV0xcFKcn/KlyjTGivota0tOC/lm8Li ICx653oDfHyoXycUnGS9Ge35UKGovcpPtFGD48WWvWPP8AQyjOxZ104VeZJNrv2AqpzlW2eW4yhP4S Xc9uy66beiafbeya5ZF1lcnR0uD7pnVqUvFa01tdP+QO458HNKUJwT7No48Qvai64qaaa9pdjrxRf+PF/w DyHi1KPzQG2PerKtuMoqK5cvUyfiENtxrnKK7ySN+jzMbo3rqhrf4ElcsjEh0Sp64L1iBZTdc6HXB7X5Hzh hui1uVCcU3yn6M3AAAAAAAAAAAAAAiVV+LZLyYqyuT3rfY6potsyfPvSi0tRiisAZJ3/SW nFeTrh+pqABLOmx+IQtUfYS03v7GJ0zfiELVH2FHTe/mVACf0qndQo1rb6t9zeK1Fj+iPQBhmUyux3GPvb 2jiCvsxbIWVqMunUee5UAMcOuVWNCE1qS3tficY1M4ZN05R1GT4e+/JSAJcSmddl7nHSm+Oe/c4hDKx nKMIq2De02+S0ASY1Fvnyvu1GUlrpRWABD5N+NdKVEVOEudP0EacmeXXbbFaXon7pcAJcnHsdbsqGv Mj3T9TlyzbWo9Mal6y3ssAE2XjzscLKn/yQ/uYZMsqePlzK4wgu/Pc+gZ5FbuplWnpv1AxjV53h0Yj6bitGc Ppsa1Uq4rS11t9kV0w8umMG99K1s7AlwaJ11ThbHXU/j3RnCvJxW41Rvlbe1t9i4AYY6yHOU72kn2gvQ wVWRjXWOMCnCb337FwAhrpyJzkLrYrWudPt34O76LY5Hn4+nJ+9F+pWAJlvMtsjuKqint+uysACbNpna q+iO9S2+TW+vzqZV71tGgAhr+mwrVSrjxwpNneFRZVcxWLTb457IYAmwKp00yjZHTct9/sQjTNelSt6fYc db2UgCOdn1ORK3HSkp+9FnsKr7r42XahGHKin3KwBJl1W/SK7qYqTitaPMuq6/HgID297a32LABhl47vg ul6nF7izGTzbIeW64x3w5bLQBN9GdeFkqHtSa/qzTFhKvHhCa1JLIGoAlxaZ133ynHSILa578sqAAIxza15F 8px0pS2nvvyxGma8Qlb0+w463v7EVACWqmcc+yxx9iUdj7+R3l0ztgnXLpnF7X2m4Ahs+mXvup1Rinw5 bN3jv6F5Ce30639puAJsLz4R8u2tRjFcPfcAAiniSlinqzX/Hvqb36nefjzujF1rck/jrgqAHFdajRGt9IHTJIV5O K3GqKsrB2tvxXADDHWQ5yne0k+0F6G4AEWXiSuyYSivZeIlk1zaXdRqc9ql2igAZYITpx4wa1LuzLEpnXbf KcdKUtrnv3KgBNhUzqla5x11S2uRg0zqjZ5kdbltclIANJpp9mRQrycVjyVFWVt7W32LQBNj02+bK696k+F FdkMKmdUrXOOuqW1yUgCWymcs+u1R9hR03v5nl9Fscjz8fTk/ei/UrAEkXmW2R3FVRT2/XZ1lUTlZG6lrl zl+j9SkARTjIZOoTiqob5afc7yKJyux3CO4wfL327FQAl8eU5Rup4tj/corcpQTnHpl6o6AE2bTO2VXRHftL b5Pc6qd1CjWtvq33KAB5Fagk/RHr5QAEWJXkY83W4J1t76tmmPTOGVdOUdRI2e+5SAjsmmdmTROM dxi+XvtyeZOPY7Y3UNEzHun6lQAJcs21qPTGest7Osmmyy+iUvtQe5Pf2oqAHF1atqlCXZolr+mUR8tVxs ivdey0AZY0blFu+Scm96Xod2Q8yuUhx1LR0AIko5IEfKhXGSXaWzvDouqvsatqS97fdlgAmwqZ1Stc466p bXlysec7I3UtKyPo/UpAEblm26j0Rq+Mtli7cvYAE+Xju6MZQerlPaZm55so9Hlxi/WWywAS5VNtmLGC9u aa2+2zrlx/Px4x3qcdaZQAI1POUejy477dWzq6m2WF5bfXz9nqVADimLjTXGS01Fj/wBDDApnTCasjpu W1yVACZ0zfiEben2FHW9nN1FteR5+Ppt+9F+pWAJlvLtsi5JVQT2/XZ7Omb8Qhao+wo6b38yoAT51U7q FGtbFvVuMqmVuL0R95aeigATVRutx5V3R8t6STTM4PMph5arjPXaWy0AT4VEqISc2uqt20vQoAAAAAA AA AA AAAAAAAAAAAAAADK3ljXbCptz7M1AAzyHaq90xUpb7M7jvpXV31yB6AAAPJS6YuT9Fs4ouV9fXFNLLeu QNAS4ds7Lb1OW1GWl9ncqAAAAAAAAAAAAAAyMiNCi5JvqeuDUAAAA

AAA+
fYrbPELK4WygtLfPppC6FmFKNkbJTg
3ppnvmRr8Vm5tJNJbfyR74jbGdcaoNSIKXZAe+ITmpUeXJpyb1p/I6UJYIVls7HY9evxOM1dNmKvg9fkUZ
sHPFmo8vubLVTO+Hm2ZEoylykn2NcO6co2Qm+qVfr8THGpxLaU5PUI725aN8f6PBWyx05OK5+0cfHi8
vqnZfJS3xFPsVYqurU438xj7sm+6MYwxcqLn+7I686GHKU/Opc3OK4jIDmvqzzSnO1wgnqMU9HeNOd
WU8ec+uLW4tmGJTRPqh7NkX6vRTRXixyEqnuaw+HtAY1xtvyb4K2UYKT3z9p1FTxxyFfmSICfxOsL63k
/e/Vjk+v44GWTR05VUfMm+t92+3PofQqh5dah1OWvV9yTNajl48m9lf8AEttW0019gEviMpRxtxbT6
I2ZxnzlHFqcZNPa5T+w68T+q/+yM/EPqlXzX5Ae24tvIStlfPzEt6T4KMOx240ZS5I2Z3f+4s+6/yMfdvqi+bA
7y6vMqb65R6U3w+5P4dV/wAas65d2unfbzb+5n91k3hs4/R+nqXvt8AT49U7b74xsc19Xta7vlmlanjZsa
vMIKE16nWB+/yfvfqxkfxkj5f5AZNk7cIY8J9CS3KRxapYcoWQuC4N6IfvZzkV1rxB+fvomuGaWUYVeup9/
hLYFy5RHmWWSuhj1S6XLltFiWkkuyIcl+T4hVbL3GtbA5ycezHoc4Xza9U2d5E5Lw2uSk09R52dZ91bxXF
TTctaSe/U4yE34XDx0ogV1vdMX69KjcG1xxLLJty6W+7+w1hkVLGjJzj7vbFjNiQdnh90Vy23r+iA9qrnkx8
229x32SetGuHbPrspsl1OHaXxRhiU4ttXtvU133LRRiwxlPyG3JLTe9oDzw2UpUycpNvq9X9g6pftPp6n0
9PbfHYz8OsjXCyE5KMILfL0K7I2eKOuHtdOt/gB7b0eRISpjZ5dc07T7njcsS+vVrnXN6ab3ozdVSzrl38KT3
F70aypwoTjHe5N8alsC4jyLLLsIY9cnBJbk0WEMpKjxJznxGxcP8A35AeXVW4kVbXbKST9pSOs+1vFrnBuP
U0+H9h1n3Q+juEZKUpa0lyY5sHDBpi+6a3/QCnHx51z8yy1yk1yvQ3n+7I8menNn7uXyYEGJVbk1bndN
QT0knzTEIOvKsx5zckltN/w/C/adeGfVn95nNX8Vt+7/gDxuzMyZwjNwqhw9ep5PzMGyD8yU6pPTUvQY
sIRIXV2Pp6ntNjPnG5101tSk5b49AOfeaulxn1yfVLs3wi6mryoOPXKf09yZL4nxXX8Flr8yHQ59S6V67A6B
5GUzxUotNP0egAA
AAAB5cLPE7YzipLp9fki
mrFpq1Qhz8W9m2lvelv4gDiymFkoucdul2uemx2ABhPDOnLqdfL+D0a11wqj0wior7DoAYTw6LJdUocvv
p6Na64VR6YRUUdADK3Gpue5wTfxXB7VRXsv+OKW/U0AHEKYVzIKMdSm9t77iVMJ2RnKO5R7PZ2AOL
aa7opWR3rse11xqgoQWorsdADm2qFsemxbXfueWU12wUz3Fdls7AHkkpRcXy whole string continues

ku7M682EpqE4Srb7dSApBjfkwoIBTT9r1XoZftCCabrmov+bQFYCakk09p9jK/IjR09SbUnra9ANQZ33Ror
c5ba+CPY2xIT5v8ALrYHYMqsiNIl04xXxMf2hDuq7HBfzaArBzXONkFOD3FnOREqK+uUW1vXAGgOLLY
10ux+6lsU2xuqVkeE/iB1KSjFuTSS7tnMrYRq8zq3D4rkx+kV341knCThHhr4nNjg/DG649MdcJ/MCjzoKn
zd+xre9HUJqyCnHs+xHL+Ff8AqvzN8P6rX8gNgZ33Kivrkm1vXBg/Eld41zlH1euAKwTPOq0uhSm2t6S7f
M7x8qF7aScZLumBsDC/KhRJRacpv+VHIWZCyfRKMoSfZSXcCgGTyIrJVLTTa2n6DlvjjwUpJvb1pAagyuyl
UVqU9pvtH1MoZ0HNRnCde+zkgKgD5/iF710RU4tS97snwBdZONcHOXZdz2E1ZBTj2fYlvt87BtfRKOUoS
RnTmwrohFQnNxXOlwgLwZ0Xwvh1Q38Gn6GVubCFjhGMrJLv0+gFIMsfJhen07UI3T7mrek38AMrsiuJ
XLI9kjuqyNtanHen8T5rvUs7zJVTaS4jrkssy4UqvqjJKa327AUAwx8qN/VqLio87Zm8+HU1Cuc0u7SArBnR
fC+HVB/NPujF59aIKPTJyT1pLuBUCeJLhdY4dMoT+DJbspvKrkoWJLvh4gfSBzXPzIKWnHfo+50AAAAAAA
AAPJb6Xrv09AHzvD8au2uU7F1c6S+B1CP0bxFVw9ya3r/AH5HSpxxbJ0iKnXLnpb7HWPrbLId9+ILWIFe
gFYMK7/pLTivJ1w/U1Ais/i1f3f0Y8Q9+j73+DudNj8Qhao+wlpvf2M9zKZ2zpcI76ZbfIFJlfVrfus1ObYeZV
OHbqTQGOB9Th+P5mXhqX/ACy1z1a2dYUb615VlaUFvUtnWDTOqNmR1uW1yBnLjxaOvWPP9GZdd
n7QsICvzJLhLfYpITN+IQtUfYUDn7+ZzfRbHI8/H05P3ov1AxyFkXuD+j9Lg9ppndqUvFa01tdP+TuLzLbI7iq
op7frs9nTN+IQtUfYUDn7+YHPijaoil2cuTzzMjy/L+i+zrWtIGTQsilw3p90/tJ4yzYQ6PljJrhS2BpgQsrocblt
Pq42a3uCpk7eYa5QojZGpK2XVL4jq86iUE9N9gJ6rbpVpUYyVfpvRz4b+5tX2iH02MFUoRSXClvsjvcpsp
hZGyOtvh77gc+FxxkSlrnq1v8DKUp/tGco1+ZKK4W+xTgVTppIgyOm5b7/YjnlosV6vo11+sX6gY5Kyb1H/
x+lx00z6K7EieZbOKcVVFPi99IYHzsmU/wBox6YdbiuInuT9lyK+l4+udp7N8rHnOyN1LSsj6P1OHLnt1Ho
jv8ZbA4zN9GN1e9tb+fB5nuTy6oqPXpbUfybZINlrp6F1dL232OsvHld0zrerldgMb5ZF1Tg8bW/XfY4y1K
ODSprUk9NGrnmj0eXGD9ZbPc2i23HhGPtyT5fC9AN74r6NZHS0oPj8DHw1JYqaXLb2UWxcqZxXLcWkZ
YVc6sdRmtS2+AMcbjxC9LtwxLLVKycKfMIJ8vfYrppnHntsIHUZLh7M3Tfj3SnjpThLvFsDIRvszK7ZU9Gu
Hye1JPxS3a3qO1/Y1p+ITuU7NVwX8q9Tyqmcc+yxx9iUdJ7+QHHiH73Hfr1f4HifuV/eNMymds6XCO1G
W3z8hnUzuqXI8yi96A0yavOolBPTfYIV1tFXI30brS1uPwNnC7JxmrF5c98aZm5Zvl+W6ovjXVsCjG8ryU6e
IM5zfqlny/U9xKXRSoN7e9s8zfqlny/UD3DSWLXpd0YYiUM6+Efd7nNEsuGPFQrjOLXsvfY2w8eVSIoX7s
m+fsAy8Niv+WWuerWzi+U/2luMOtxXC/AowaZ1Rs8yOty2uTzJx7HbG6hrzI90/UDHJ+kZEFF4+mntPZrk
0WWxqshrzlc6fqeOWba1HpjUvWW9neTTbKULKZe3H0fZgZWZCkIDMocVvv6HPiHv0QiuqPovidWxysl
KudcYR3tvZtk4vm1QUHqdfusDG2eRbVkt43DXx7HF8Zw8MhGa1JS/ya9ec49HlxT/7bOsqm2zEjBe3NN
bfYDaMYrGUUlrp7fgT+G/VJfef5FST8pR9enRhg0zqocbl6bk33A48L+ry+/8AoiwiXK8jHm63BOtvfVstAix/
4jf8jzDSeZkPXKk9f1ZrTTOObbZKOoyXD2eYtM68i+Uo6jKW099+WBxk8eIUND2Wk19M55dM4x3GPd7
KQIsV67j/AD/U98U+rR++vyZ1m02SnXbUuqUH2Ocqu7IxIly9WdW3HfbuBRKMVjOKS10dvwl5NeFT1
8df3PoSTdTiu/TonxseSw5VWrW2/UDLHsvrojGGPuOt733O8GuyFlrnDojLII5rWXJx8uMI2RXuvZRjRuSk
75Jtvik9AO7240WNd1F/kYeHRSxU13k3sqkIKLT7PhkNdeViuUK4KyDe1tgewSXis9Lxs/ohR/E7vu/4OMf
zP2jLzddfTzo0uqurynfTFS6lpoDzx97j/e/wAHGW5PPPrSh19K2o/EWVZV1lc5wSUX7qfY3y8edko21PVk
P7gY5DyL6uh42udp77Hmv1KGKpe8u/wDY0cs2xKPRGv4y2d5mPO6qPS9zhz8NgaZf1Wz7pHNteEx16v
T/AKndn026twdcYrXL33Na8dywVTyul6/pyBITZfCmMY4246+Pc78PrsrVinBxTe0jmH0yiHlquM0uFLRj
xtjb+dLqk3vXwA1I7r+RH4X9XI9/wDRFj5RFiV5GPN1uCdbe+rYDFSeFe2uU+BmcZuO133+ppj0zhlXTIH
ZdnvuMmmmdmTROMdxj+XvtyBPfkf7S3GHW4rhgef5P0JgovH009p7NsnsdbsqBvMj3T9TlyzbWo9Ma
I6y3sDjMXVdiqa7Jp/gb56X0Of2a/M5yabLL6JRW1B7k9/ajXLhKzGnCC3J9l+IDE+q1/dRxn1+Ziy+MfaRp
jxcKIRktNLk0aTTT7MD585/Svo1W+GuqX+/10Fa68G2lv2oy6f6/6zbBxZ02zlnfZhnuLsSU82M0v+NtOXI
HOZF04Fda45Sf5nUbMhVKCxV0613KciX0uDevVP4MmhLNrgq/LjLXClxDvw+uyqqUbluPtBWzXKr8zHn
H11tHuPGyNerpdUt7+RoB812O7FopT5ILT+SPYT+j15NW/d5j+PH+DTGxJV5cpyWoLfTyMzFndfCUFw+J
c9gCr8vwtr1cdv8Tz/7T+H6lWRBzx5wgtrSRj5Nn7O8rp9vXbf2gcS/hX/AKr8zfD+q1/l4dM/2f5XT7fTrW
zXGhKGPCMlppcoDHxL6q/mjaqK+jQja6Vx+Bxm1ztx3GC29rjZrWnGqMX3UUgJPC0vJm9c9WhHjxW
WvWP6GmBTommUbI6blvuPJn+0Hb0+x063sDLXXmZE5e8npf1PfE4pVQmuJKWkzq2i2rlD2Pp9XvRZ
y6b8qyLvioVx50n3AzyahVkJe1BrYtayc2qC5hFdT/AN/oV21qyqUH6rRngY86VOVi1J8LnfAGOTKf7Rj0w
63FcRPcn6Rkv9Lx9c7T2b5WP0dkbqWIHZ0fqcOWbbqPRGr4y2BVVvyodXvaW/mSeKfuf/e/QtXbI7J86
mV9KUOZJ718QPc36pZ8hhRSxa9LW1tnElfdhzj0tKb4ST7m2NCUMeEZLTS5QEmE+n6V0/yvhf1M8Kd0
K5Ounr2+ZbKcSmdc7nOOIJ8c9+5nGrIxZy8mKsrk96b7AKYXPOVsquhNafJcT0LJY53NRjrSgigCKP8Vl939
EM5KWTjpractNfijRUzXiEren2HHW9nmvTOzlolCO1GW299uUB1nNwx56+xf3OsOKji169VtmlsFbXKEu

neXj/SKUlPUk9pmwAjU85R6PLjvt1bOrqbZYXlt9dnH2epUAOKYuNNcZLTUUn/QwwKZ0wmrl6bltclQA
 mdM34hG3p9hR1vZzdRbDI8/H02/ei/UrAEMqsjKnFXRVdcXvSfcotrVIUoejWjsASYGPOITIYtSfc53wdZW
 POdkbqWlZH0fqUgCNyzbdR6I1fGWzTLx5XQjKD/5Icp/EoAENn026pwdcYrXL33KMSEq8aEJrUlva/E2AH
 k03CSXdpmGDVOqjpsWntvWygATY9M4ZV05R1GXZ77nuXjyuUZ1vVkJOUUACKf0y6HluuME+HLZpbju
 OC6a11Pj8eSkAcURcKIRktNRSZjhUzq8zrjrqltclIAlnTN+IV2qPsKOm9/MqAAjyqrIkwpipNLWj3JqtujQ1
 D2k9yW+xWAM8iLnROMVttaRlxjuWCqbF0vT/AA5KQBFD6ZRDy1XGaXClsox42xg/OI1Sb3r4GoA4ui5U
 2RittxaX9DCqmyOBKpx9tp8bKgBLXjzeCqZNwnz69uTiMs2EFX5UZa4UtloAww8d0UuMnty3oxK8jh
 m63BOtvfVstAGUhF9ImpxXla9lmoAA
 AAA
 AAA
 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAH//2Q=="></figure><p>Constrain
 ts:</p><p>0<size <100</p><p>0<data<1000</p><p>Input
 format:</p><p>First line indicates the size of the queue</p><p>Second line indicates the
 elements of the queue.</p><p>Output Format:</p><p>every lines indicates the
 enqueue of each elements</p><p>last line indicates the fine queued elements.</p>

answer

```

#include <stdio.h>
#include <stdlib.h>

struct node
{
    int data;
    struct node* next;
}*front=NULL,*rear=NULL;

void linkedListTraversal(struct node *ptr)
{
    while (ptr != NULL)
    {
        printf("%d ", ptr->data);
        ptr = ptr->next;
    }
}

void enqueue(int data)
{
    struct node *n;
  
```

```

n = (struct node*)malloc(sizeof(struct node));

n->data = data;

n->next = NULL;

if(front==NULL){

    front=rear=n;

}

else{

    rear->next = n;

    rear=n;

}

}

```

```

void dequeue()

{

struct node* t;

t = front;

front = front->next;

free(t);

}

```

```

int main()

{

int n,i,data;

scanf("%d",&n);

for(i=0;i<n;i++)

{

scanf("%d",&data);

enqueue(data);

}

printf("Dequeueing elements:");

for(i=0;i<n;i++){

dequeue();

```

```

    printf("\n");
    linkedListTraversal(front);
}

return 0;
printf("for(i=front;i<=rear;i++");

}

```

question

<p>Question description</p><p>A
positive integer <math
xmlns="http://www.w3.org/1998/Math/MathML"><mi>X</mi></math> has been stolen. But
luckily, <math
xmlns="http://www.w3.org/1998/Math/MathML"><mi>N</mi></math> hints are available,
each described by two integers <math
xmlns="http://www.w3.org/1998/Math/MathML"><msub><mi>a</mi><mi>i</mi></msub></math>
 and <math
xmlns="http://www.w3.org/1998/Math/MathML"><msub><mi>d</mi><mi>i</mi></msub></math>
, meaning that <math xmlns="http://www.w3.org/1998/Math/MathML"><mrow class="MJX-
TeXAtom-ORD"><mo
stretchy="false">|</mo></mrow><mi>X</mi><mo>-</mo><msub><mi>a</mi><mi>i</mi></msub>
<mrow class="MJX-TeXAtom-ORD"><mo
stretchy="false">|</mo></mrow><mo>=</mo><msub><mi>d</mi><mi>i</mi></msub></math>. The hints are numbered <math
xmlns="http://www.w3.org/1998/Math/MathML"><mn>1</mn></math> through <math
xmlns="http://www.w3.org/1998/Math/MathML"><mi>N</mi></math>. While some of those
hints are helpful, some might be just a lie. Therefore, we are going to investigate the
number <math
xmlns="http://www.w3.org/1998/Math/MathML"><mi>X</mi></math> under different
possible scenarios.</p><p>Initially, we neither trust nor distrust any hint. That is, each hint may be
either true or false. Then, in each of the <math
xmlns="http://www.w3.org/1998/Math/MathML"><mi>Q</mi></math> stages, we will
either:</p><p> 1 id</p>Entrust the <math
xmlns="http://www.w3.org/1998/Math/MathML"><mi>i</mi><mi>d</mi></math>-th hint (<math
xmlns="http://www.w3.org/1998/Math/MathML"><mn>1</mn><mo>≤</mo><mi>i</mi><mi>d</mi><mo>≤</mo><mi>N</mi></math>). That is, from now on, the <math
xmlns="http://www.w3.org/1998/Math/MathML"><mi>i</mi><mi>d</mi></math>-th hint must be
true unless declared otherwise in the future<p>2 id</p>Distrust the <math
xmlns="http://www.w3.org/1998/Math/MathML"><mi>i</mi><mi>d</mi></math>-th hint (<math
xmlns="http://www.w3.org/1998/Math/MathML"><mn>1</mn><mo>≤</mo><mi>i</mi><mi>d</mi><mo>≤</mo><mi>N</mi></math>). That is, from now on, the <math
xmlns="http://www.w3.org/1998/Math/MathML"><mi>i</mi><mi>d</mi></math>-th hint must be
false unless declared otherwise in the future.<p>3 id</p>Neutralize

the <math
 xmlns="http://www.w3.org/1998/Math/MathML"><mi>i</mi><mi>d</mi></math>-th hint (<math
 xmlns="http://www.w3.org/1998/Math/MathML"><mn>1</mn><mo>≤</mo><mi>i</mi><mi>d</mi><mo>≤</mo><mi>N</mi></math>). That is, from now on, the <math
 xmlns="http://www.w3.org/1998/Math/MathML"><mi>i</mi><mi>d</mi></math>-th hint may be
 either true or false unless declared otherwise in the future.<p>After each stage, you
 should determine the number of possible positive values <math
 xmlns="http://www.w3.org/1998/Math/MathML"><mi>X</mi></math> and report such
 values in increasing order. If there are infinitely many such values, print <math
 xmlns="http://www.w3.org/1998/Math/MathML"><mo>-</mo><mn>1</mn></math> instea
 d.</p><p>Constraints</p><p><math
 xmlns="http://www.w3.org/1998/Math/MathML"><mn>1</mn><mo>≤</mo><mi>N</mi><mo>,</mo><mi>Q</mi><mo>≤</mo><mn>200</mn><mspace
 width="thinmathspace"> </mspace><mn>000</mn></math></p><p><math
 xmlns="http://www.w3.org/1998/Math/MathML"><mn>0</mn><mo>≤</mo><msub><mi>a</mi></msub><mi>i</mi></mi></msub><mo>≤</mo><msup><mn>10</mn><mn>9</mn></msup></math></p><p><math
 xmlns="http://www.w3.org/1998/Math/MathML"><mn>1</mn><mo>≤</mo><mi>t</mi><mo>≤</mo><mn>3</mn></math> for every stage (update).</p><p><math
 xmlns="http://www.w3.org/1998/Math/MathML"><mn>1</mn><mo>≤</mo><mi>i</mi><mi>d</mi><mo>≤</mo><mi>N</mi></math> for every stage.</p><p>In tests worth 74 points in
 total, <math
 xmlns="http://www.w3.org/1998/Math/MathML"><msub><mi>a</mi></msub><mi>i</mi></mi></msub><mo>,</mo><msub><mi>d</mi></mi></msub><mo>≤</mo><mn>500</mn><mspace
 width="thinmathspace"> </mspace><mn>000</mn></math>.</p><p>Input</p><p>The first line contains two space-separated integers <math
 xmlns="http://www.w3.org/1998/Math/MathML"><mi>N</mi></math> and <math
 xmlns="http://www.w3.org/1998/Math/MathML"><mi>Q</mi></math>.</p><p>The <math
 xmlns="http://www.w3.org/1998/Math/MathML"><mi>i</mi></math>-th of the
 following <math
 xmlns="http://www.w3.org/1998/Math/MathML"><mi>N</mi></math> lines contains two
 space-separated integers <math
 xmlns="http://www.w3.org/1998/Math/MathML"><msub><mi>a</mi></msub><mi>i</mi></mi></msub></math>
 and <math
 xmlns="http://www.w3.org/1998/Math/MathML"><msub><mi>d</mi></msub><mi>i</mi></mi></msub></math>,
 describing the <math
 xmlns="http://www.w3.org/1998/Math/MathML"><mi>i</mi></math>-th hint. It is guaranteed that
 no two hints are identical. That is, for every two different <math
 xmlns="http://www.w3.org/1998/Math/MathML"><mi>i</mi></math>, <math
 xmlns="http://www.w3.org/1998/Math/MathML"><mi>j</mi></math>, it is guaranteed
 that <math
 xmlns="http://www.w3.org/1998/Math/MathML"><msub><mi>a</mi></msub><mi>i</mi></mi></msub><mo>≠</mo><msub><mi>a</mi></msub><mi>j</mi></mi></msub></math> or <math
 xmlns="http://www.w3.org/1998/Math/MathML"><msub><mi>d</mi></msub><mi>i</mi></mi></msub><mo>≠</mo><msub><mi>d</mi></msub><mi>j</mi></mi></msub></math>.</p><p>Then, <math
 xmlns="http://www.w3.org/1998/Math/MathML"><mi>Q</mi></math> lines follow, each
 containing two integers <math
 xmlns="http://www.w3.org/1998/Math/MathML"><mi>t</mi></math> and <math

$\text{xmlns="http://www.w3.org/1998/Math/MathML"}><\text{mi}>i</\text{mi}><\text{mi}>d</\text{mi}></\text{math}>$ — the type of an update and the index of an affected hint.
 $</\text{p}><\text{p}><\text{strong}>\text{Output}</\text{strong}></\text{p}><\text{p}>$ After each stage, print the number of possible values of
 $\text{xmlns="http://www.w3.org/1998/Math/MathML"}><\text{mi}>X</\text{mi}></\text{math}>$ (in case there are infinitely many of them, print
 $\text{xmlns="http://www.w3.org/1998/Math/MathML"}><\text{mo}>-</\text{mo}><\text{mn}>1</\text{mn}></\text{math}>$). If the number of possible values is finite and non-zero, in the same line, continue to print those values in increasing order.
 $</\text{p}><\text{p}>$ Explanation for sample test case 1:
 $</\text{p}><\text{p}>$ In the sample test, we are given
 $\text{xmlns="http://www.w3.org/1998/Math/MathML"}><\text{mi}>N</\text{mi}><\text{mo}>=</\text{mo}><\text{mn}>3</\text{mn}></\text{math}>$
 & hints and
 $\text{xmlns="http://www.w3.org/1998/Math/MathML"}><\text{mi}>Q</\text{mi}><\text{mo}>=</\text{mo}><\text{mn}>10</\text{mn}></\text{math}>$
 & stages.
 $<\text{br}>$ The first stage is described by a pair "1 1", which represents entrusting hint
 $\text{xmlns="http://www.w3.org/1998/Math/MathML"}><\text{mn}>1</\text{mn}></\text{math}>$.
 $<\text{br}>$ After this stage,
 $\text{xmlns="http://www.w3.org/1998/Math/MathML"}><\text{mrow class="MJX-TeXAtom-ORD"}><\text{mo stretchy="false"}>|</\text{mo}><\text{mrow}><\text{mi}>X</\text{mi}><\text{mo}>-</\text{mo}><\text{mn}>3</\text{mn}><\text{mrow class="MJX-TeXAtom-ORD"}><\text{mo stretchy="false"}>|</\text{mo}><\text{mrow}><\text{mo}>=</\text{mo}><\text{mn}>0</\text{mn}></\text{math}>$ must be true, so
 $\text{xmlns="http://www.w3.org/1998/Math/MathML"}><\text{mi}>X</\text{mi}></\text{math}>$ must be equal to
 $\text{xmlns="http://www.w3.org/1998/Math/MathML"}><\text{mn}>3</\text{mn}></\text{math}>$. We report
 $\text{xmlns="http://www.w3.org/1998/Math/MathML"}><\text{mn}>1</\text{mn}></\text{math}>$ possible value
 $\text{xmlns="http://www.w3.org/1998/Math/MathML"}><\text{mn}>3</\text{mn}></\text{math}>$.
 $</\text{p}><\text{p}>$ Then, the information that
 $\text{xmlns="http://www.w3.org/1998/Math/MathML"}><\text{mrow class="MJX-TeXAtom-ORD"}><\text{mo stretchy="false"}>|</\text{mo}><\text{mrow}><\text{mi}>X</\text{mi}><\text{mo}>-</\text{mo}><\text{mn}>3</\text{mn}><\text{mrow class="MJX-TeXAtom-ORD"}><\text{mo stretchy="false"}>|</\text{mo}><\text{mrow}><\text{mo}>=</\text{mo}><\text{mn}>0</\text{mn}></\text{math}>$ is neutralized at stage
 $\text{xmlns="http://www.w3.org/1998/Math/MathML"}><\text{mn}>2</\text{mn}></\text{math}>$. At this point,
 $\text{xmlns="http://www.w3.org/1998/Math/MathML"}><\text{mi}>X</\text{mi}></\text{math}>$ could be any positive integer, so we print
 $\text{xmlns="http://www.w3.org/1998/Math/MathML"}><\text{mo}>-</\text{mo}><\text{mn}>1</\text{mn}></\text{math}>$ in the second line.
 $</\text{p}><\text{p}>&$

answer

```

#include <stdio.h>
#include <stdlib.h>
int compare(const void *a, const void *b);

```

```

int search(int* arr,int value,int start,int end);

int main()
{
    int i,n, q, t, id;

    int one = 0, len = 0, qu_rear = 0, front = 0, b;

    int idx[3][200002], sol[400001] = {0}, queue[200002], answer[2], **sol2;

    scanf("%d %d", &n, &q);

    for (i = 0; i < n; i++){
        idx[1][i] = 0;
        idx[2][i] = 3;
    }

    for(i = 0;i < n;i++){
        int a, d;
        scanf("%d %d", &a, &d);
        idx[0][i] = a + d;
        sol[len] = a + d;
        len++;
        if (d != 0 && a - d > 0){
            idx[1][i] = a - d;
            sol[len] = a - d;
            len++;
        }
    }

    qsort(sol, len, sizeof(int), compare);

    for (i = 0; i < 2 * n; i++){
        if (sol[i] != 0 && sol[i] == sol[i + 1])
            sol[i] = 0;
    }

    len = 0;
    for (i = 0; i < 2 * n; i++) {

```

```

if (sol[i] != 0) {
    sol[len] = sol[i];
    len++;
}
sol[len] = 0;

sol2 = malloc(2, sizeof(int *));
for (i = 0; i < 2; i++) {
    sol2[i] = malloc(1 + len, sizeof(int));
}

for (i = 0; i < n; i++) {
    if (idx[0][i] == 0){
        idx[0][i] = len;
    }
    else {
        idx[0][i] = search(sol, 0, len - 1, idx[0][i]);
    }
}

for (i = 0; i < n; i++) {
    if (idx[1][i] == 0){
        idx[1][i] = len;
    }
    else{
        idx[1][i] = search(sol, 0, len - 1, idx[1][i]);
    }
}

for (i = 0; i < q; i++){
    scanf("%d %d", &t, &id);
}

```

```

id = id - 1;

if (t == 1 && idx[2][id] != t){
    if (idx[2][id] == 2){
        sol2[1][idx[0][id]]--;
        sol2[1][idx[1][id]]--;
    }
    one++;
    queue[qu_rear++] = id;
    sol2[0][idx[0][id]]++;
    sol2[0][idx[1][id]]++;
}
else if (t == 2 && idx[2][id] != t) {
    if (idx[2][id] == 1){
        sol2[0][idx[0][id]]--;
        sol2[0][idx[1][id]]--;
        one--;
    }
    sol2[1][idx[0][id]]++;
    sol2[1][idx[1][id]]++;
}
else if (t == 3){
    if (idx[2][id] == 1){
        one--;
        sol2[0][idx[0][id]]--;
        sol2[0][idx[1][id]]--;
    }
    else if (idx[2][id] == 2){
        sol2[1][idx[0][id]]--;
        sol2[1][idx[1][id]]--;
    }
}

```

```

}

idx[2][id] = t;
if (one == 0){
    printf("-1\n");
}
else{
    answer[0] = -1;
    answer[1] = -1;
    b = 0;
    while (idx[2][queue[front]] != 1) {
        front++;
    }

    if (idx[1][queue[front]] != len){
        if (sol2[0][idx[1][queue[front]]] == one && sol2[1][idx[1][queue[front]]] == 0){
            answer[b++] = sol[idx[1][queue[front]]];
        }
    }

    if (idx[0][queue[front]] != len){
        if (sol2[0][idx[0][queue[front]]] == one && sol2[1][idx[0][queue[front]]] == 0)
        {
            answer[b++] = sol[idx[0][queue[front]]];
        }
    }

    printf("%d ", b);
}

if (b > 0){
    printf("%d ", answer[0]);
    if (b == 2)
        printf("%d", answer[1]);
}

```

```

    }

    printf("\n");

}

}

return 0;
}

int compare(const void *a, const void *b){

    return *(int *)a - *(int *)b;
}

int search(int *arr, int value, int start, int end){

    int mid = (start + value) / 2;

    while (arr[mid] != end){

        mid = (start + value) / 2;

        if (end < arr[mid])

            start = mid - 1;

        if (end > arr[mid])

            value = mid + 1;

        if (value > start)

            return -1;

    }

    return mid;
}

```

question

<p>Question description</p><p>Anderson is a Placement trainer. he is working as CDC trainer in reputed institution that during training the youngsters are struggling in queue concept. </p><p>Anderson usually gives interesting problems to the students to make them love the DS. </p><p>One such day Joe Anderson provided to the final year students to solve the task such that, Circular Queue using Linked List. there is no memory waste while using Circular Queue, it is preferable than using a regular queue. </p><p>Because linked lists allow for dynamic memory allocation, they are simple to build. </p><p>Circular Queue implementation using linked list is identical to circular linked list except that circular Queue has two pointers front and back whereas circular linked list only has one pointer head.</p><p>Final Year students were lacking the idea to solve the problem.</p><p>Being an exciting youngster can you

A43x3rknJj2nftiX2s5bWjekVj4uoDImpabVvTtr8GMkZstJjkRWPr2qAHK+Kb4qx2WrEPnKzbbc3G/z3dgH
PJWbYjie3b4PlZtbBWce2/5urOPHGOJiN9pnchOb5pbm4ifnuVx3x44ikxM77zv8XYBwtbNaOTyIrv8d3
28Ri00x+WzszfHGtbffaJ32B8w15OKsfk2AAAAAAAIcv84w/on/K5DI/nGH9E/wCVwAAAAAAAAADob
3Vvo0zm91b6Anw0wTirNuTv8d5dbWrhwTbHEbbs4MWO2GszSJnb5PuprFdNMVjaOr/ALB9pktMTe1
YrTbf82Yy5bRyq445P5z1y3krN8E1jtmGMeekY4i08mYjrgGue3w2vWOuO2JzjLltWLvxxt+csRE9Hy3mN
uXMzEO+L3VfoDF8lr6ebUj4Tvv8DTTkmlEVcnbqn4sY3014jtndrTZKzjrTf8W3YDsAAAAAAAAAAAAA
AAAAAAAAAAAAA+X9i30fXy/sW+gOOlrWcETMRPa+Y9q6m0U9nbr2Zwael8MTaJ3n828ERXIYpjafn84
Ai+TLvNLvrX4b/FrHktaLVmi5deDhFMWPeuak7/AdjfrdK1jmb2xUmsz2bz2g+2tmpWbTas7fDZ9y5zJBG
Svx2cJjFOOYilpvt19reWJ6FX+gO+PnJ677bT2RHwbAAAAAAAAAAAAABDoP4vFqXIdB/F6r9QLgA
AAAAAAAAAAAAAAA+AAAAAAAAAAAAAAA+AAAAAAAAAAAAAAA+AAAAAAAAAAAAAAA+AAAAAAAQ5f5xh/RP+Vzz
9ZN8Ovx5ox2vWKzH4WvWNVlZeAlhD6xt5bLwPWNVlZeAlhD6xt5bLwPWNVlZeAlhD6xt5bLwPWNV
lZeAlhD6xt5bLwPWNVlZeAlhD6xt5bLwPWNVlZeAlievTQ+sbeWY8D1jby2XgC6Ii2iNoJiJjaY3hD6xt5b
LwPWNVlZeAlnya1md5rEz9EXrG3lsvA9Y28tI4AumImNpjedS7EPrG3lsvA9Y28tI4AuijsjZ8itYneljf57lv
WNvLZeB6xt5bLwBclfWNvLZeB6xt5bLwBclfWNvLZeB6xt5bLwBclfWNvLZeB6xt5bLwBclfWNvLZeB6xt
5bLwBclfWNvLZeB6xt5bLwBclfWNvLZeB6xt5bLwBclfWNvLZeB6xt5bLwBclfWNvLZeB6xt5bLwBclfWN
vLZeB6xt5bLwBclfWNvLZeB6xt5bLwBclfWNvLZeB6xt5bLwBclfWNvLZeB6xt5bLwBclfWNvLZeB6xt5bL
wBclfWNvLZeB6xt5bLwBclfWNvLZeB6xt5bLwBclfWNvLZeB6xt5bLwBclfWNvLZeB6xt5bLwBclfWNvL
zeB6xt5bLwBclfWNvLZeB6xt5bLwBclfWNvLZeB6xt5bLwBclfWNvLZeB6xt5bLwBclfWNvLZeB6xt5bLwB
clfWNvLZeB6xt5bLwBclfWNvLZeB6xt5bLwBclfWNvLZeB6xt5bLwBclfWNvLZeB6xt5bLwBch0H8Xqv1H
G3lsvBn0Zbl59Rbaa7zE7T8AegAAAAAAAAAAAAAAA+AAAAAAAAAAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA
AAAAAAAAAAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA
AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA
AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA
AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA
AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA
AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA
eANDO2T514G2T514A0M7ZPnXgbZPnXgDQztk+deBtk+d
eANDO2T514G2T514A0M7ZPnXgbZPnXgDQInU5ItMbV6p2fOIX+VRzNUQrEnSr/ACqdKv8AKoZwrEnSr
/Kp0q/yqGcKxJ0q/wAqnSr/ACqGcKxJ0q/yqdKv8qhnCsSdKv8AKquJ3iJFiYnQAKAAAAAAAdnWb23mj
rERMX2NbZPnXgDQztk+deBtk+deANDO2T514G2T514A0M7ZPnXgbZPnXgDQztk+deBtk+deANDnktp
Sbb1nb8nDpV/IUSZiNqxJ0q/wAqnSr/ACqJnCsSdKv8qnsr/KomcKxJ0q/yqdKv8qhnCsSdKv8AKp0q/wAq
hnCsSdKv8quuDnbJmxaljb5CxVEuwA6AAAAAAABnJaa13jbffbrNsNzrwBoZ2yfOvA2yfOvAGhnbJ868
DbJ868AaGdsnzrwNsNzrwBoZ2yfOvA2yfOvAGhnbJ868HDJqMIMk12rO35CTNllk6Vf5VOIX+VRM4ViTp
V/IU6Vf5VDOFYk6Vf5VOIX+VQzhWJOIX+VTpV/IUM4ViTpWT5VOIX+VQzhWEdgOgAAAAAAAAAAAAAAA
AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA
AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA
AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA
PNVsAHIAAAAABPZL0K+zH0efPZL0K+zH0Vtx6fQBqAAAAAAAx19mf1T/22xi9mf1T/wBtgAAAAAAA5
6j3F/oijsw6j3F/oijsrlyAAyAAAAAFGk9qydRpPasO+PakBW4AAAAAAADGX2Y+sf9tsZfJ6x/22AAAAA
AAAAi1H8Rb6QtRaj+lt9IHnenMBHnABAAAAAWnvRjsgl7IfeoAAAAAAAAAAAAAAA+AAAAAAA
AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA
AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA+AAAAAAA
ygHQQ29JcmYi2DJEz2brYnesTPV1bg+jhg1Mz8lq0rM1r/r+Eu4AAPPt7dvrL4+29u31i8R5qtgPtY5VoiPil
+Dt0a/zgpgtN9rfAXGXEdsuGaz1bbTO0PItPasTMzHUGMuQAhPZL0K+zH0efPZL0K+zH0Vrx6fQBqAAAA
G8b7DjqsM5sW1Z5N69dz5g7DzdN6Smtua1MbWjq5X/AjejW0WiJid4kGcXsz+qf+22MXsz+qf+2wAAB
Jk1tsd7V6Pkml+Mdj5j1/OY73rhvPJ+HzBYILekuRtytPkjfs3XunlVidtt47AfQAc9R7i/0RQt1HuL/RFHYjlKa
BkDrXBa1YmNtpLYL127JfxlVFFtP+Cnpif4sVwWtWLRMrbIlkNWrvbTWfgvIKNJ7Vk6iSe1Yd8e1ICtwAA

AAADeBFr6ZMU9IwTMTHtR84fdJ6Qx59q2/Bf5T8QU5fZj6x/22xl9mPrH/bYAAA56jPXBJ5Vt5+UQnj0hEWiMuK+OsztFpBYJc2tjHkmIMdslo7dm9Nqq6iZrtNb17ayDuAAi1H8Rb6QtRaj+It9IHNeMBHmB9rWbWi+LrOa/zgdREy4jtjwTN9rdkPmTDNbRtMbTO0BjO3IdbYLvrmzMdTkJMTAAEbejHZAR2QK9QAAAAAA
AA
AAAAAAAAAAAAAAAAAAAAADnk1GLFaK5LxWZ69pY6Zp/Fql1WKub0pipeN6zTs4u/q7TeH+4N9M0/i1OmafxasertN4f7nq7TeH+4N9M0/i1OmafxasertN4f7nq7TeH+4N9M0/i1OmafxasertN4f7nq7TeH+4N9M0/i1OmafxasertN4f7nq7TeH+4N9M0/i1Qek5wZYjLiyVm8dsR8Vnq7TeH+7hrNNpNNgm3N/inqrG/xBPpPSV8W1cv4q/P4w9OmWma1LY7RMdbwMWK+a8Vx1mZl62i0U6W9Ztbe1oneI7IBeAADlqMU5sfJreaTvvEwCHU1z4tROPyUrelZ6o37la1uti1a4qW5PLijtPyhudHqMv4c2o5VPlt2qowY4il5FZ2jbrgEGPW4sGbHix2jmOT+Kduvfr+z0q2i1YtHZMbwmto4nWVv7V5ERtNdu1VEbRtAAAPPt7dvrL4+29u31I8R5qtjeL3lfqw1jmIvEz2RIkb d89ck3ia7zH5NXna2KJn8Xxcsme3LnkW6nKbTM7zPWNJqiHbJW3Pb9e28Gpmec236tnOct5jabPIrTed7Tvl5mqP8ZAHPZJGo123Vp42J7HoV9mPorXj0g6Rr/LwdI1/I4egDV5/Snf5eDpGv8vD0AHn9I1/I4Okay8PQAef0jX+Xg6Rr/Lw9AtMVizmdogHga2uab85lxciz6vq+aXWzdNP4Z3r3ZdNrfLr9TtjrM1jqj8Iuk9G0xbWy7Xv8vhAKdjk53DF9pjITM7T9XZf7M/qn/ALbAABw1tM2TDyMO289u8/Bx9H5OTytPakVtj+Xxfb6bURe1sWomltO+0vldBPIvM5rc7ad5tAOwqpnpqj1N6Vvjp7Mb/B6GK8ZMdbx2WjdHbR6jJHly6jfH8tu1bSkY6RWvZEbQD6ADnqPcX+iKOxbqPcX+iKOxGXIADJVtadNEV7XzHFq4bcvfb4MzI2w1itvxR2uV8lr+1O40mqldrxa2GnJ3k3mukjbqnf8Ay5Vy3rG0T1Pk3tNeTM9QmUMgDgbpbJSI5xV5V+raGFGk9qw749uH Snf5eDpGv8vD0BW7z+ka/wAvB0jX+Xh6ADz+ka/y8HSnf5eHoAPP6Rr/AC8HSnf5eHoAPOnPrpiYnTxMS8vNS+PLMXrybdu3ye/qs9dPhm89vwj5y8fDps2tyTfsiZ67SDppNdkia4r73jeNvm9pj0eLTUjkxvbeN7T9VYAAEx1xG35vP1Vum5K4MUb1rbe1vkq1WLJmxTTHfkTPbKFotVhryceesR+kFeXj0+Ob22j4fnKfRY73zX1N45PL7I/JjPoc+bJW1s0fhiNomHfBh1NMkTkzRavy2BSAAi1H8Rb6QtRaj+It9IHFnMBGDeH3tXXNXJOxeu+3wcsUxXEz2Q3fPblzyLdQ7iYt+utp/+bHG/X8XoA2jPvMTtynHITyt9+v5tTlvO289gZRLeptPObb9Ti+2tNp3ntfBzM3kJ/LtASNt8n0h8L49jk+ke/jXR2QK9SHk+ke/jOT6R7+NcAh5PpHv4zk+ke/jXAleT6R7+M5PpHv41wCHk+ke/jOT6R7+NcAh5PpHv43ya+kNvbxxryeyQR+jc2TNhvOW3KmLbLEPoj3GT/wCyf+oXX
XX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXAh/zjD+if8AK5DI/nGL9E/5XAAAAAAATMREzPZDzL6fLr8/Lv vTFHVX5vTAc8GDHgpycddo/wC323vaf1bYt72n9QbAAAAAAAAB59vbt9ZfH23t2+sviPNVsAEAAAAAA J7HoV9mPo8+ex6FfZj6K1430AagAAADlqcM56RTlcmzs+LbtI1AYw4aYacjHWlhsAYxezb9U/9tsYvZt+qf+2wAAAAAAAAC9R7i/0RR2LdR7i/0RR2Iy5AAZAAAAACjSe1Zo0ntWHfHtSArcAAAABPI0sZ80WyzvS vZVRWIrEREbRHwAGMvsx9Y/wC22Mvsx9Y/7bAAAAAAAARaj+It9IWotR/EW+kDivTmAjAAAAAAAAC NvRjsgl7IFeoAAAAAAA7JCeyQQ+iPcZP8A7J/6hch9Ee4yf/ZP/ULgXXXXXXXXXXXXXXXXXXXX cu0dUz1RAOo4zmvEbzhink/Vqc1Yxxelmd+zYHQCzXr12xTffnu1kyTGOL02mPj9AdAiYmlmOyQAAAAA AEOX+cYv0T/lchy/zjF+if8rgAAAAAAAAGLe9p/Vti8Ty6zEb7bg2M8q3hzxg5VvDnjANDPKt4c8YOVbw54 wDQzyreHPGDIW8OeMA0M8q3hzxg5VvDnjANDPKt4c8YOVbw54wDQzyreHPGDIW8OeMAzqPcX+iKOxZl5d8dqxsD5 /OE/MZe7+4zriZ05jpzGXu/ucxl7v7ozwlzHTmMvd/c5jL3f3DCXMdOYy939zmMvd/cMJcx05jL3f3OYy93 9wwlzuat2rOfMZe7+7tpsd6TabRtuOqKZiXcBWWAAAAAAADGX2Y+sf9ts5lma9UbzvEnKt4c8YBoZ5Vv DnjByreHPGAaGeVbw54wcq3hzxgGhnIW8OeMHkt4c8YBoZ5VvDnjByreHPGAaRaj+It9IV8q3hzxhPlxZL 5zTfeqfzgc1ReHAdOYy939zmMvd/dGOEuY6cxl7v7nMZe7+4YS5jpzGXu/ucxl7v7hhLmOnMZe7+5zGX u/uGEuY6cxl7v7nMZd/Z/clpm62OyAjsFegAAAAAAA7JCeyQQ+iPcZP/sn/AKhch9Ee4yf/AGT/ANQuA AAAA
AMkU5XJjbrlQ45aWjGStEV1bTAOU3rj2tTLNvnEt6ibTOPkztu+xabTEVwzHzm0PuaszfHtE7RPwgHy+G a0m0ZLcqOt9nNPR4v/AKp6nS/XjtEfKXKuObaWK7bWj5g+xgtMbzktnzTTblZltO8xJGXJEcmcVuV+z5pt +Xk5Xbv1g3gmZnJvP8AqfMkz0jHG/UzHLw5Lfgm1bTv1Pn/AMI9RS00mIB9yZlilmk35FY/dnlxjtHlyTaJn riW70tTLN615cT2w+1tNrRth2J4zMA+3mek0jfq2fdRMxhtMM5q2i9cll327YYy3yZccxXHal+O4NZMk0w

AAADN/
bp9Wmb+3T6g0AAAAAAAACDJ7y31Zaye8t9WUeerYAOQAAAAACex6FPYr9Hnz2PQp7FfoNeN9AVq
AAAAAAAazj/wBX6paZx/6v1S0AAAAAAADGX3VvohtsXZfdW+iGOxGXIADIAAAAUAaT2rfROo0ntW+
g7o2pAVuAAAAAAAzl9hpnL7DQAAAAAAACPU++/osR6n339BzXpyAR5wAAAAAAI29GOyAjsgV6g
AAAAAAABm/t0+rTN/bp9QaAAAAAAABFqPfS5umo99Lmjz17AByAAAAAAT2PQp7Ffo8+ex6FPYr9B
rxvoCtQAAAAAAADsItFo3id4fXm6ucuh87i68dp66/CJB6TNvbp9XHS63Fql2idrfKXa3t0+oNAAG+
3aMZsUZsc0tMxE/IGuVHzh9eNlw4+l1w48lo29qbWenmzU0uDeevaNoj5g7bxultJHJy8vPf8A+bL1xX5Q
tAABBk95b6stZPeW+rKPNXsB9p7dfql+PsRMztHapy5K47bcijNq1vW9Y6rDvD1Nas1nae18U5rxOSK8m
O2Ot9z2rTesVjeY7RJphKAOSex6FPYr9Hnz2Kq6vBFYictd4j5jXjdxx6Zp/FrxOmafxa8Vauw49M0/i14nTN
P4telOw49M0/i14nTNP4telOznnnyWxY+XWvK27Y/Jnpmn8WvEnV6eY97XiDWHPjz05WO2/5fJ0eFqLR
p9Ty9NkjaeaqvldpPSVMm1cu1bfP4SC3H/q/VLTGPr5X6mwAAN4+cG8fOEgtwYaVtmva+8/CJ+LOg01c
mntNsksN/IPYDOOVHzh9eNnw441VcGPJa3/ABTa3Y9i1YpSKx1xEA+gAxI91b6IYXZfdW+iGOxGXIADIFO
Oa108WmsSRyc1Jnk7TUdYuE0tFYtMdUsq7ZljFW3Jjafg+UmtdPyprEi4x9SjV7cq0zEbbssjgUaT2rfRO7aa
9acu1piliO2R3RtWOPTNP4teJ0z+LXird2HHpmn8WvE6Zp/FrxB2HHpmn8WvE6Zp/FrxB2HHpmn8Wv
E6Zp/FrxB8z6jmLxy4/BP+r5O1bVvWLVmJjfjCbNn0ubFNLza7T+bysWpyaTLMUtyq79nwkh5fYaSYtbj
1GPa5Nu7KsAAAiYnsmHPUYrZcfJreaT84efq9PXSY65MeS/Ocrsme0HqTMR2zsRO/YkvpJz35eXJbaYj8
MdWzlo98etyYqXm2OI+M9gPQAAR6n339FiPU++/oOa9OQCPMDWP3lfqoyZKY77ciJ+Y6im8XTVibTtH
aWiaztPaqiK0yVmsdV/2ZyXi2WK8m0q3aOsfxMKc9613rFY3m01MOZi0gHYJG3ox2QOEavT7R/8teL70
zT+LXir1Ow49M0/i14nTNP4telOw49M0/i14nTNP4telOw49M0/i14nTNP4telPuoyzhrF+TvWO3b4NYst
M1OVjtEw521WmtExOWkxLx73nS6mZ0+SJr2xsD32b+3T6pNJ6Rpm2rf8ADf8AaVV/ap9QbAAActRqKaa
kXvvtM7dTq558FNRSK5N9onfqBPpPDEbzW8f0db6vFTHW9p25XZhT63bNqMemrEdu9vo45bWj0jy
aY+XNa7VrPZALcGsx578isWifzhQjxaq9dRXFqMUUtzb2ZhYAACLUE+IzdNR76XNHnr/oBvD15a/Ucx+sPt
azado7VOTJSI+TyIn5vtYrjyxtHVf9h3h+7SzExMxPbD4otaLZoryY6pM96xvSK9fzExTgDknsehT2K/R589i2
mXHFI/8Akr2fMa8boMc9j8SnGDnsfiU4wrVsY57H4IOMHPY/EpxgGxjnsfiU4wc9j8SnGAbGOex+JTjBz2
PxKcYBsY57H4IOMPtcuO07VvWZ+USDQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAINZl1EazHiwWiOVXsns+JyfSXxf8Av9H3UfzbT/pn/K4EHJ9Jd/F/7/Q5
PpLv4v8A3+i8BBYfSXxf+/0OT6S7+L/AN/ovAQcn0l38X/v9Dk+ku/i/wDf6LwEHJ9Jd/F/7/RjNi1+TFauS+
KazHX/AO7PSed6Sz2vMabDEzafa2B5MTNbb1nrj4w9b0fqM+aaxkrvWOyz5pPRkV2vn657q+YitqREbR
ANgAOeozcxi5fJm23wh0AeVqc2mz45jHinnbf7dp3dLaLPljHac0RNax1TG+0vQitYneKxv9H0Hk3xaiNfp
Ofe/J3i23Y9WkTFl07zEdcvu0b77RuAAAgte8t9WWsnvLfVIHnq2NU9uv1ZfYnad4HKrLOLlbX33cr5om1
eTH4auVrTad7Tvl4Opqd73xWmLdfK3hjPeL33r2bOYJNVwAQ+CiNFp7REzirMz1p3oU9iv0GvG4dB03h
VOg6bwqqBWqfoOm8Kp0HTeFVQAn6DpvCqdB03hVUAJ+g6bwqnQdN4VVdnnyTjxzNY3tPVEA8r0lGD
FMY8W0lt2zMfBx0uh6iYnbk070vQ0/o6JvzuonlXnr2XxERG0RtAOIxixzSJmYie2XZjH/q/VLYAAIs2s
wcu2PPittWereu8SnwZObzZc+LFbmdtuT85epNYt2xE/V9iNuwHl6nPpc+O3IxTOW3+3ad1+lreumx1ye
1EdbpFaxO8Vjf6PoAAMZfdW+iGOxdI91b6IYRlyAAyVY5rGmj9jE5KUpNccdvxcuXbk8nfq+Tl6mp3rkx2x
xW+/U+TkpzHljt3/y4gZSADkdtPSuTl1vG9ZjrhxUaT2rfQd0bfeg6bwqnQdN4VVArDp0HTeFU6DpvCqoA
T9B03hVOg6bwqqAE/QdN4VToOm8KqgBPOh00RvOKrxtXOO2eYw0itY6o2+L2NZzmSOZxR127Z+Upm
IOOPT/AltuVf5yCDTejsm0ZMu9I+Ef7MdUbM5fYaAABy1Oeunwze0TPyiHm4NTivm5/VWtN49mu3VV
68xE9sbvnIr3Y4A87WekImlx4ZmOVHXbbsh00ObTV2xYptNrdszHat5Fe7HAitYneljgD6AAj1Pvv6LEep9
9/Qc16cgEedrH7yv1UZZxcv8cdclomYneO2C1pt09p3kdRVaHW2bfJWYj8Nfg+2vim0WjffeJcAMPdM1
ovk3r2OYDmZv+hPX1AEbUxodNMe6qdb03hVUR2QK9SfoOm8Kp0HTeFVQAn6DpvCqdB03hVUAJ+g6
bwqnQdN4VVACfoOm8Kry/SPMUyxjw0iOT7Uw9bU3vXHtje9uqE+l9HVpPOZvx3nr/KAQaT0fkzzFrfgp
8/m9iKRjjHSJmYj5urN/ap9QaAAABLpdPembJmzbcbu89W077QajTZJzRnwWiMkRtMT2SqARY9Pnyaiub
UzX8Hs1qtAAAEWo99Lm6aj30uaPPX/Q3h97X6sPsTNZ3jtgcxtTknDy/wAcTvDnOffLftuqPg5WtNp3md
5fB1NSib4uXfo33363LNal5JmOxgEmq4AIT2Oftr2ntETNZ3nr9qXKex6FPYr9Brxpfvmm7tv7pPVmm7t
v7pVitUnqzTd2390nqzTd2390qwEnqzTd2390nqzTd2390qwEnqzTd2390nqzTd2390qwEnqzTd2390uG
mw0w+ILUpExEV6ut6SGn83v+gFwAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAIdR/NtP+mf8AK5DqP5tp/wBM/wCVwAAAAAAE9jniw0xTM1j8U9sz2y6ADN/bo0zf26A0AAAAAAAACDJ7y31Zaye8t9WUeavYAIAAAAAAPQp7Ffo896FPYr9BrxvoCtQAAAAAAAGcf8Aq/VLTOP/AFFql0AAAAAAAGMvurfRDC7L7q30Qx2ly5AAZAAAAACjSe1b6J1Gk9q30HdG1CtwA
AAAAAAAGcvsNM5fYaAAAAAAAAR6n339FiPU++/oOa90QCPOAAAAAAABG3ox2QEkdCvAAAAAAAAAM39qn1aZv7VPqDQAAAAAAAItR76XN01Hvpc0eev+gAcgAAAAAE9j0KexX6PPnsehT2K/RWvG+gDU
AAAAAAQ0/m9/wBC5DT+b3/QC4AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAE2a+++bkW35MR2R8QUxMT2TEkzEdspLTWvXireLR+Teo3tOP4bg78qN9t4fXDJgpGOZjeLRG++75bjbo1bfGerchflRvtvD
7vEuMaenJ695n57vmmiYvkiz32khfeJ7DeN9t3HT+1k/VL5I/icYO8zt2kTE9k7pcl4tmtF4tNa/CHzetbvnFW8dfXEWcveN9iZ27XG/8TT6Nan3FgdDeN9t4T5bzXDjiN+ul7GJfYequSJ+ewKrTyazM/Bw53JxDy9+uZ
6t/gWta2kmbdvY7UiObrG3wBz/wDn233pwMOW1sdr5Nto7Nn3U2mKRSvbbqZz15Gm5MdkaA+xfNk
66REV+G7U85OKd+q/w2Yrz1qRNeTWNuqG8OSbxMWja1e0H3FfnMcW+PxbcdL1VvHytlTsAAAAAAACH
UfzbT/pn/ACuQ6j+baf8ATP8AlcAAAAAAA Afz26NP1qxbtjcH0Z5qnyOap8gaGeap8jmqlGhnmqfl5qny
BoZ5qnyOap8gaGeap8jmqlIEWT3lvqyu5nH3IOYx9yEZTx3m6EXcxj7kHMY+5AnX6hF3MY+5BzGPuQH
X6hF3MY+5BzGPuQHx6hF3MY+5BzGPuQHx6hehT2K/RnmMfchvshXdNOIAOwAAAAAAAGcf8Aq/VLT
M46TO8x1yc1Tug0M81TunNU7oNDPNU7pzVO6DQzzVO6c1Tug0M81TunNU7oPmX3VvohjsXc1Tun
MY+5A5qpyQi7mMfcg5jH3Rn1+oRdzGPuQcxj7kB1+oRdzGPuQcxj7kB1+oRdzGPuQcxj7kB1+oVGk9q3
0duYx9yH2tK09msQrqmi03aAGgAAAAAAADOX2GiYi0bTG8M81T5A0M81T5HNU+QNDPNU+RzVPkD
QzzVPkc1T5A0M81T5HNU+QNI9T77+irmqfJ8nDjntrAkxeLIRbzGPuQcxj7kly60Qu5jH3IOYx9yA6/Ulu5j
H3IOYx9yA6/Ulu5jH3IOYx9yA6/Ulu5jH3IOZx9yBetuOyAFagAAAAAAADN/ap9Wny1Yt2xuD6M81T5H
NU+QNDPNU+RzVPkDQzzVPkc1T5A0M81T5HNU+QNDPNU+RzVPkCTUe+lzXThxz20g5jH3IGdVF5uhF
3MY+5BzGPuQjnr9Qi7mMfcg5jH3Dr9Qi7mMfcg5jH3Dr9Qi7mMfcg5jH3Dr9Qz2PQp7FfuzzGPuQ3H
VGyu6acQAdgAAAAACGn83v+hchp/N7/oBcAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA5Zcd
uXGTHMcqPhPxdQHKJzWmN61rHx+Jlpa16TEDUT1uoD5ejmlojtmHKuGZ08Ut1S7AOETniOTyaz/u3fN
NEfxJEzvO/aoZrjlrlThbad5By5GTHktOOltFuvaXyMeW2at7xG0fL4KAHG+O8Z0cx7bz2xPxfazmtaOVWt
Y+Px3dQHlnjtNq3p7VfhLGSubLSYmsRH37VADlbFN8VY7LViNnzfPptyK7/Pd2Ac8IZtgMj652fcNuVirP5
Ns0pFImlK9kzuDEUtbPy7R+GOx0vWL0ms/F9Ac5+kcmK1tEdk7tYqTjra153meuXV8vWL1ms77T8gctLH
/AMW8/wCqd3YililiOyAAAAAAAEo/m2n/TP+VyHufzbT/pn/ACuAAAAAAAABwzWvz1KUtyeVHy3
fYx5t+vNH9sMZ4mdTjis8mdp63SuPJFomcszHy2Bq2alZmLW2mH2cllpFpnajcYrFtZbeN9oh9vETq61t2
RXel/MHSmbHedq26322WILbWnadt3LUxEVraOq0TGz5asW1deVG+1dwdaZsd52rbrc8mprTLFd+r49R
qliLY7RG08qIMkf/04/pIO1bRasTHZL6AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAdhz95vatMXK5P8Aud0uPLX
Hmy8frn4QDrXNveK3pNjs/N1TTbn8tORE7UneZlnJet81oycqa16oiAVm6bDaK5oiKw5E/CY7GbczNp
35Vp+YKxPgtNtNfeZnbel3NNiicdbzvM/D8gUAAAAAAAAlafze/6FyGn83v+gFwAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAdhz95vatMXK5P8Aud0uPLX
Hmy8frn4QDrXNveK3pNjs/N1TTbn8tORE7UneZlnJet81oycqa16oiAVm6bDaK5oiKw5E/CY7GbczNp
4sVqYr1mYmZmZhvDSaYq1nthsAAAAAAAQQ0/m9/0LkNP5vf9ALgAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAdhz95vatMXK5P8Aud0uPLX
4n7O98WPJO96Rafzh86Ph8KvAHh1jpve/Y9Y6bxP2duj4fCrwOj4fCrwBx9Y6bxP2PWOM8T9nbo+Hwq
8Do+Hwq8AcfWOM8T9j1jpve/Z26Ph8KvA6Ph8KvAHh1jpve/Y9Y6bxP2duj4fCrwOj4fCrwBx9Y6bxP2P
WOM8T9nbo+Hwq8Do+Hwq8AcfWOM8T9j1jpve/Z26Ph8KvA6Ph8KvAHh1jpve/Y9Y6bxP2duj4fCrwOj
4fCrwBx9Y6bxP2PWOM8T9nbo+Hwq8Do+Hwq8AcfWOM8T9j1jpve/Z26Ph8KvA6Ph8KvAHh1jpve/Y9
Y6bxP2duj4fCrwOj4fCrwBx9Y6bxP2PWOM8T9nbo+Hwq8Do+Hwq8AcfWOM8T9j1jpve/Z26Ph8KvA6P
h8KvAHh1jpve/Y9Y6bxP2duj4fCrwOj4fCrwBx9Y6bxP2PWOM8T9nbo+Hwq8Do+Hwq8AcfWOM8T9j1
jpve/Z26Ph8KvA6Ph8KvAHh1jpve/Y9Y6bxP2duj4fCrwOj4fCrwBx9Y6bxP2PWOM8T9nbo+Hwq8Do+H
wq8AcfWOM8T9j1jpve/Z26Ph8KvA6Ph8KvAHh1jpve/X+z56y03fn7gxgR2Y68H3mcXh14An9ZabYZ

r/ANoLRF6zwfK/9p6zwfK/9oLRF6zwfK/9p6zwfK/9oLZ7EfTOYzzh1HV3b/CXz1ng+V/7U2u1On1WLaltF
47J2B60TEvxE7xPyZv7dPq8LS63Lpp2ieVTuy9bDqsepmk0nr364BSAAxn5yMc8zETf4btvlrVrG9pil+cg87
LqtZivWlq45tbsiOt6EW5OPIX2jaN5QarBfFe+qpmjlfCjhjVZ8ufHjrGO/lmIm3JjtBXpc+TUZLWiljDE7V+cq
Xk9LtXU4opiyVpWu3N/N6tLcqkW2mN432kH0AEGT3tvqy1k97b6so89ex9rG9oj5y+NU9uv1HLtbBSvtX
2K4YrljlT1fBrNhNJaJiY/q+XtWs46b77T1yrW0R/j5mx05XVPXM7bPmTBSkTM269uqGsuOeci+8bbwxqv
eR9ESYj9mziAMx6FPYr9Hzn2NRi18xvXUUivwjb7DXjXiHmfSPmMfD7HM+kfMY+H2VquEPM+kfMY+H2
OZ9I+Yx8PsC4Q8z6R8xj4fY5n0j5jHw+wLmMuOMuOaW7JScz6R8xj4fY5n0j5jHw+wJ6azLo804c29qx2T
8dnp4s1M1OVjtEw8j0hhz1it9RlpeeyNo6/+kuny5ceSJwzPK+UfEH6LH/r/U046W17YpnJXk2meuHYAAE
me+spa00jHzcfGZY02o1WfDe8VpvHVX83fU441O0cdcvJ2nr260+ky2w5r6fJes0pG8W7NgYzarWYZrF6
497TtER1y9DHypx1m8RFtuvZDrNPaMltVTN11jeI2V6bLoBt0vaNpmOsHUAGcvurfRBHYvy+6t9EEdiM
uTQAMneMFeRFrW23gthrG1otvX4y3bHOTBSInr2fJjmsE1md5kaWj4+5sePkR17dXV+bEYK8iTbbdbvJS
cIKzWY6oZy/w9BZiN2CjilmYjsfAGQ76T27fRwdMUZLvvGK0Vvt1Tl7o2tEPM+kfMY+H2OZ9I+Yx8Psrdcl
eZ9I+Yx8Pscz6R8xj4fYFwh5n0j5jHw+xzPpHzGPh9gXCHmfSPmMfD7HM+kfMY+H2B11uGcmPIY52yV6
4mE2I9j9fl1HVPZynTmfSPmMfD7PL1eO+PNMzl1taeuZqD38kxbFvExMT8YbeDos2es8jHvak9sT2Pejs
6wAAc9Rkvjx746Te3ySX1Or08RkzUpNjnadp7F1rRWs2tO0R8Xn2tb0jk5Nfw6es9c94HbLm1N78nBjjk7
b8q3ZJptTktmthzViLxG8THxdNTqaaakfG09Vax8XPR4LRe2fNP8A8l/h8oBWAAj1Xw6LEeq99/Qc16cgE
ed9rHktEfOxecGOs7TfZxx+8r9VGbDN77xMDumPxmmGK5drT9PzfMuOnORET1zO0x8m7XrGXHxs7Z
ZvjmM0X6tuVA6tFrQ+ZMNKRP4uv4Q4Oup97/AEchnVa/4ACRt6FfZh9TaimqtNZ0+WtK7dcTHx4OXM+
kfMY+H2V6lwh5n0j5jHw+xzPpHzGPh9gXCHmfSPmMfD7HM+kfMY+H2BcleZ9I+Yx8Pscz6R8xj4fYFzzd
Zzmiy89i93aeuvw3dOZ9I+Yx8Ps55tPrbYrRI1GOabde8fYFOi12PURtvyb92Xent3+r8zvMW6p64+MPZ9
GZc+SJ5yu9e9PaC8ABhqtRnpqqYcMVmbRv1rHy20RNp26viCDpOrpqKYr1x72+XydL6rLly2x6asTye20s
aKJ1GfLqbdns0cNLhyZcuanPWxxFuyvbIPQ0855i0Z4rG3ZNfi7lHkyV1OTT3vOSKxvFpWgPI/Yn6Pr5f2J+
gPPgIEeWdjvTBWccWm2zgqik301Yidh1TF3O2CNotW29fi6Xx4+brvb5t83zbmcNotMbz8Cac5hpyZjq
V3aPjNMFZxxabbONoiLTETvDvfq0sR+adHFVoABy6af31VrzorItalw2it/hMt8z6R8xj4fZW/HpcleZ9I+Yx8
Pscz6R8xj4fYdrhDzPpHzGPh9jmfSPmMfD7AuEPM+kfMY+H2OZ9I+Yx8PsC5Jr8N5pz2GZjT5fGGOZ9I+
Yx8Pscz6R8xj4fYGdj6Trfamb8Nvn8HoRO8bw/N6ik481q2tW1t+ua9iv0bn1EXiLYm+P47/AHr4+2/wCpp
nH22/U0AACXL48GPHbLfFFrT/25ej8OG+C9Yv8Aj7W3VMO+XVZcWS0W09rU+E1ccUZlZdVOGaxydo
pHbIPus02mwaafw7X/0z8ZlXpYvGmxxk9rbrebjyZjzzm1GDJe3+mIjqh6eDLOXFppNPYkHQAHPUe5si
W6j3NkSMuQAGLvjx0nFy7k4sd6zOOZ6vg1jrFtNETOxHJwUt+LeZVralNW5vmq778n4MUxY5xcuz7WK
5MEV5URMPk7RpprvE9f+UX1xyxeXPI7GQGQ76T27fRwfazqlt/8AxzWZ+PKHVG3oCHleke7iOV6R7ujX
oXCHleke7iOV6R7ulFwh5XpHu4jleke7iBcleV6R7uI5XpHu4gXCHleke7iOV6R7ulFwh5XpHu4n3Q58+XP
kpm5P4PIhxBaAAAAAAAAAAAAAAAACDURE+ldPExv8Ahn/K7kU7teCLUfzbT/pn
/K4GeRTu14HIp3a8GgGeRTu14HIp3a8GgGeRTu14HIp3a8GgGeRTu14JtdmppsMzFa8u3VWNlc9iLoc6j
Nzuo7P9NAeZp9Jl1Nt4javxtL1sGkx6aacnrtPbKmtYrERWIi+EM39un1BsABjNipmpNLxvDYCOPRuKLbz
a9o+Uz1LliliNogAcraettTXPMzyqxs6gAACDJ7231Zaye9t9WUuevY+1na0T8pfAcumXjy7bxvDmATN33
efnJMzPa+AAAD0KexX6PPehT2K/RWvG+gDUAAAAl2iJtadoh9Yy465axW/XXfs+YPJtizekc83iNscdky9
LTaPFp6/hje3zI3lrlG0RtADGPtt+ptnH/q/U0AACXL0MWS83ibVtPbtLVNDhrjtTaZ5XbMz1qAEfq3Fv12v
Ne7urrWK1itY2iOyH0AAAbnL7q30QR2L8vurfRBHYMuTQAjF1v15W0tY3iY+LnMzPa+Aszd93n5ybztu+A
AADvpPbt9HB30nt2+g7o2qAvuAAAAAA46vPGnwzb/VPVEPN0/o/Jql85nmaxM7/nL1bYaWyRe0bzHZ
v8GwceZphw8nHWlh2Zy+xLQAAOeow1z4+RaZiPyTR6Mx1jaMmSI/KVoCO/o3Fe0Wm994jbtbw6KmLJ
F4yXmY+EypAAAEEq99/RYj1Xvv6DmvTkAjzvtZ5Non5S1kyTe8zG8MAXH3efnL4ATO/aAAAEbehX2YfXY
vsw+q9QAAAAAA830jmvmv0bBEzP8Aq2elPYxjxUxRPljrntn4yCLSejK49rZfxW+Xwhbjja1oj5ts09u/1Bo
ABnLjjLjtSZmlt1Ts0AxhxVw4q469kOWbRY8t+Xvalp7ZrOygBy0+mx6eJ5EtVpbM9suoAPI/Yn6Pr5f2J+g
PPgIEeWdjPOX/AOKKRvEx8XMCJs+zMz2m8x2TL4A+7zttv1PgAAA6af31VqlT++qtVvx6AB2AAAAJtdqe
YxbV68luqsKWOZpzvOTG9vhM/AHmaX0bfJPOZ94ievb4y9THjpjrFaVilaAZx9t/1NM4+2/6mgAAAAAA
Ac9R7myJbqPc2Rly5AAyT85PNcjb+rAC3AAABB30nt2+jg76T27fQd0bVAK9AAAAAAAh0X8bqfqQ6
L+N1P1BcAADN71pG9p2Boc656WnaJmJ/Nq+SuOY5XxBocukY9+2frs67xtvv1ADI0jHv2z9dm6ZK3mYr
O+wNDNLxffb4TtJN4reKz2z2A0M3yVx+1P9Hymal52iev5SDYzN4i8U+Mvt7RSs2nsgH0fOVHJ5W+0bbu

+Uvms9G0phm+HfevXMSCalcnpDWcqY2rv1z8oXel8k009aR1cqet89FaitsfMzERavZ+bXpbFOTTxaI3ms
7gn0Gp0unxfimecntnkqMnpDR5KTW0zMT/ALU/o/ouXHyMtK85Hz+K22m0IY3tSkR9Qeb6OyRTXcmk
70tvDv6a/wD8v6qdLXSZJm+Gld6ym9Nf/wCX9QU+jcNcelrb8VuuZa1+Ot9LfeN5iN4b0n8Lj/TD7q/4XJ
+mQeX6Jw1yZrWtG/JsezMRMbTHU8n0L7zL9lesDxlrzPpSK17OUt9LZJpporH+udpR5v5vH6oXelMU5N
NvWN5rO4MeisFK4lyzETa3x+ULrVi0bWijh53orVU5rmbzFbVnq3+K++WmOs2taliPzB49o6J6Sjk9Ucr9
pUel8PVXNW0OzqIPWZ1vpGLRH4eVv8A0exnxRlw2x8YBy0Obn9NW3xjqI5+uvOq1tcNeys7Oel1M602
Wlvi1fV39E4Zvknt9IB6FqRj000jsiuZfQ3vb/R6mb3N/pLy/Q3vb/QHb017rH9zb9FYK100ZNom1vix6a9
1j+su/oz+Ax/1/7kHfNjrkvWraN94eT6Kmaa21PnEvZnsI43o3+Y2+kg6emckzfHj36tt1uiwUw6evJiNjeZ
S+mME2rXLWPZ6pa0XpDHOKKZbRW1Y23n4guyY65KTW8bxLw8NOb9JVP+h+nJs9TNr8GOkzF4tPwiHk
6W031+O09tr7yD9AADy/TXzi/qt0H8Fi/Sj9NRPJxT8N5VaC9eh4/wAUdUbSDtn9xk/TP/TxvReGuXUTN
o3isb7PYzTE6e8xO8TWf+nl+hff5P0/5B7G0bbbPE9I44w62Jr1RO09T23j+mP4qn6QehrMs00drx2zDzP
R+bT4bWvmmeV8Ord6epxTi0U0jt5PU8vQTgi849RSN57JkHoT6S0sxtNpmP0vNnjprq5ME/h3+T1ei6T
bfm6bOeKmhyZppSIjtUFsdhPVAT1wDwK5cd9bOTPM8nffs3enHplSxG0Wml/S82aV02umM1OVTf4/J
6INPo71i1aUmJB53pDlp801vhn8fx6tt3o4Mk5PR3Knt5EwxkroceStLUpyrKL0rj0160iIrFZ6oB5nob+lv8
ApffTOSZzUx7/AIYjd89DfxF/0temMU85TLHZttIO2n1ukwYq0rMxO3X+Fvn6zSajDau88rbqnk/FrSV0efF
E8inKIOujdcuHRYqTa9KRAJvQ2Sd8mOZ6vahP6RryvSE1+ez1dLTT8mMuCsRyo7YeZrf5pH1gHr4cVcWO
taxttCD0xjrzVckR177bvSjsQ+mP4WP1Az6lw1jBzm29pngsz465cNq2iOuE/on+Cj9Uq7ezP0B5HoiZjU3r
8Nn30xkmc1MfwN2fRP8Zb6S36YxTy65YjeNtpB20+t0mDFWIzmoR/CzqtZpNRhtWZnf4TyWtJXR58U
TzdlvEdcS65cOixUm16UiATehsk//Jj36u2HD0pG+vmPnEPU0tMHJ5zBwli3yeX6U/jp+kA9fT4a4cVa1jb
q603pTFW2lm20b1+K1N6Q/gsn0Bx9D2301q/Kzh6a9vF9JdfQvuMn6nPo1E8rFPw2kHoaT+Fx/phP6W
xWyaajrG/JneYdtHkrOkxzyo6q9beXPjxUi97RFZnaJB5mi9IY8GKMeSkxt8YX01Wm1ExWL1mflaNnydPp
c34uTSfziXm+kNNh0/JnfbrmezfsB7Y4aG9r6PHa/tbff3B53pr3WP6y7+jP4DF/X/uXD017rH9Xf0Z/AYv6
/wDcgqeBpccZddFJ7OVL33iejv5jxB7Va1rERWIiI+TxzDM6j0jzc9kTFY28XX1tp9fGX4TO8A9jHjripFaRE
RCf0hgri01rTH4qxvEu2HPjzUi1bR9Pkm9l6qmPBakWib2jbaAcfQ2SzfHM9UdcJvSMcrXzHz2hX6HwzW
lsk/6uqE2t/mkfqqD18WKmLHFKREREIPTGOObpkil332l6SD0x/DR+oD0ThrXBzlxE2tPaq1OOuTBeto+D
j6M/gqf1U5PdW+kg8n0PaY1F6x2TD2Hjeh/wCJt+I7IM5Pd22+UvH9E9Wsnf5S9p4mqx30Ws5yfhmd4
B7Yn0+sxZ6xMWiLfGJdrZKVje14iPqDh6R/gsn0S+hd+Tl+W8f5cvSGs6TaMWlrrvxl6Gg0/R9NFZ9queuQe
bl/m8fqh7TxNbvh9I8uezeJh7FMtMIltW0TE/mCP0z/AAtP1/4lv0X/AAUfWUvpbU0vFcVJidp3nZV6L/gq
/WQeZzIL66b6ifwcqd/i9OPSWliNotMR+l5tq10+umM1eVTfr/OHqu0+jyVi1aUmJB53pDNp881vhmeV
8erZ6OnyTI9G8qe3kTDGSmhx3rS1KbypjFSuGceOlisxPVAPL9DfxN/0f5h7Dw/RuWMGrMnVvHJ3n4P
anJSK7zalj57g8j0t/G1+kPQ1uScWivaO3bZ5OtRn1c2r7MbRD2NxInNo7Ujt23ghI+j82nwza+aZ5Xw6t1
8+k9LMbTaZj9KD0fOn5dseorG89ky9Pouk235umwPJ5yINfGTTz+Hldj3kWKmhyZppSIjtVaAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAl1+ltqqUitojkzv1qgHLSYpwaeuO0xMx8nUAedk9HXrqedwXivXvES9CI3rtal6463
0B52f0VW9uVityfc09FZZ6r5Y2esA4aXS001ZivXM9suWv0ltVyOTaI5PzWAMYKTjw0pPXNY2M1JyYb0j
qm0bNglvR+ivpbXm1lonlRHYTAEF9Be2t5+LRtvsvmN42kAefqPRdL25WK3Jn50Meistp/HljZ6wDjptLj0
1dqR1/GXYAedrPRts+eclLRXft3W6bDGDDXHHw7XQBnJXIY7Vj4xsk0Givpb2m1onePgtAS6/S21VK1ral
2nfrdNJhnBpqY7TEzXfrj6uwBPYg0mhvg1U5bWiYnfqheAzktStJnJMRX47osnozBlnIY7TXf5dcKNZpuk4u
Ry5r8XndE12CdsVpmP8AbYFFPReDF+PjabRHx19UI9HWMnpKJpH4eVNobnTa/P8Ahvytv91l+i0VdLW
ZmeVee2QVAA5anT11GKaW/pPyed6qyxO1ckcl6wDhjwTj0nM8reeTMbuGg0N9Lkta1oneNupcAlddob
6nNF62iliNutcA+VjasR8oR6r0djzzNqzyLstAeR6qzb7c5HJWaTQU008qZ5V/mrAAACNTpMepr+LqmOyY
QT6Ky1n8GWNrAPOweiq0vystuVt8F968rHasdW8bNAIdBob6XLa1rRMTG3UsyY65KTW8bxLQDzMn
on8W+Ljt+Us09E3tMc5i4PVAYw4aYMUpG0I9RoL5dZGaLREbx1LwBPrbU4YpWYid9+tQA4aLBbTae
MdpiZ3mep3mN4mABB0dTz5va0TExt1LcmOuSk1vG8S0A8zL6J/FviybfLNPRN5mOdyPVAc8GGmDH
FKR1Qj1no++o1HOvtER1dr0ABy1WKc2ntjrO0y6gJdBpbaXHatpid536nTVaeupxc3V8p+TsA8n1Vlidoy
xyVd9BXJp6Yr3mZp2SrAeTPorLX2Msf8ATeL0T+Lflk3/ACh6YD5WsVrFaxtEdUPoAl1+ltqqVrWYjad+t00
mGcGmpjtMTNd+uPq7ADz9L6Pvh1XOzaJjr6noADnnwUz05N46v+nQB5V/RN6zPN5G8Poqltvlyvyh6
QD5WsVrFaxtEfBDqNBfLrlzRaliJidl4Am12mtqcMUrMRMTv1qQHHR4Z0+nrjtMTMfJ1vHKpMfONn0BB

odDfTZZva0TExt1LwAYy4qZqTS8bxLYDy8voj8W+LjtHylBmw5MOTkZd/q/RuebBjz15OSu4ltHXR4a85G
Ss2+c/BTp9bi1GW1Kb7x2T800+iMfK3jJal+WyyTaXFpq7Y4657ZntB81ekpqa/i6rR2S8/1VlidoyRyXrgPM
n0Tth2i+95ntlZo8E6fTxjtMTMT8HcBw1Olx6mu142mOyYQW9FZaz+DJGz1gHnYPRVa25WW3K/J6MRt
GOACLV+jqZ7zek8m09v5pq+issztbJHjesA8zL6KnevNWijt3+L04jalgARar0bjzWm1J5FpTeqs2+3OxyXra
JNJoKaaeVM8q/zVgAAA
AAA
AAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAADE2tNpitYnb5zt/h93yd2v8Ad9int3aBnfJ3a/3fY3yd2v8Ad9mgGd8ndr/d9jfJ3a/3fZoBnfJ3a/3f
Y3yd2v8Ad9mgGd8ndr/d9jfJ3a/3fZoBibZliz5Ner/d9nHpc9z/AJKL+xP0efA4rmYj8UdlnuR/cdlnuR/cnE
Z9kqOlz3I/uOlz3I/uTgdkqOlz3I/uOlz3I/uTgdkqOlz3I/uOlz3I/uTgdkqOlz3I/ubxz+cettydv6pHbS+9n6Dq
muZmysBWoAAAAAAAD5e01rvEbs75O7X+77PuX2f6w0DO+Tu1/u+xvk7tf7vs0Azvk7tf7vsb5O7X+77N
AM75O7X+77G+Tu1/u+zQDO+Tu1/u+xvk7tf7vs0Azvk7tf7vs5X1FqW5M0j8rfZ3R6n30/Qc1TaLt9Lnw/
8Akdlnw/8AknEZdkqOlz4f/l6XPh/8k4HZKjpc+H/yOlz4f/JOB2So6XPh/wDl6XPh/wDJOB2So6XPh/8A16
VO/sf8k5hbH1CK5eiEdgrcAAAAAAAYi17dcVrt+dvs2zi9j+oG+Tu1/u+xvk7tf7vs0Azvk7tf7vsb5O7X+77
NAM75O7X+77G+Tu1/u+zQDO+Tu1/u+xvk7tf7vs0Azvk7tf7vsze96Vm00rtHyt9nRz1HubA5dLnw4/uOl
z4cf3fZP8AARh2So6XPhx/d9jpc+HH932TgdkqOlz4cf3fY6XPhx/d9k4HZKjpc+HH932Olz4cf3fZOB2So6X
Phx/d9jpc+HH932TgdkrMOactpjk7bR83VLpPbt9FStaZvAAOgAAAAABm9prEbRvMzs0xk7K/qgH3fJ3a/
wB32N8ndr/d9mgGd8ndr/d9jfJ3a/3fZoBnfJ3a/wB32N8ndr/d9mgGd8ndr/d9jfJ3a/3fZoBnfJ3a/wB32
N8ndr/d9mgHC+pmluTNI3j/AHfZnpc9yP7vs56j31nNGVvCxNIHS57kf3HS57kf3Jwc9kqOlz3P+X2Olz4f/
L7JwOyVHS58OP7vsdLnu8AL7JwOyVHS57n/L7HS57n/L7JwOyVHS57n7qYneIn5vOehT2I+itKKpnB6A
OwAAAAAA
AAAAAAAAAAAAAAAGae3dpmnt3aAAAAAAAB8v7E/R58PQv7E/R58DPk0AlxAAAAAHbS+9n6OL
tpfez9B1R/SsBXoAAAAAAAIYnskBnL7P9YaZy+zH1hoAAAAAAABHqfft9FiPU++n6DmvTkAjzgAAAAAB
HbAR2wEbejHYEdgr1AAAAAAAbxvtv1gDOL2P6tM4vY/qDQAAAAAAADnqPc2dHPUE5sCIBHIAAAA
AAd9J7dvoqS6T27fUR0UaAB0AAAARMT2SAzk7K/qhpnj2V/VANAAAAAAAi1HvrObpqPfs5o89f
9AA5AAAAAAHoU9iPo896FPYj6K1430AagAAAAAA
AAAAAAAAAAAAAAOefLOGvL5O9Y7dvgDoM48tMteVS0TDQ
M09u7TNPbu0AACfNrceHJNLVtMx8oZx+kMWtIcmLfjhjeep11PLjDacVYtf4JPR9oxZLYMmOa5bdczPx8
ufSeGO2Lx/wDIXiyRlxieu+0/NBr65bZYnmuvhp1zt8VunyUy4a3pG1Z+HyB0AB8v7E/R58PQv7E/R58D
Pk0AlxbriteN6x1FsV6xvMdTth36Pbk9u5gm8xbl78n8x3FMM9Hnm9/9Xyc64r232jsnZ23t0beJnfcxzNc
F5+O6raHC1ZpO09rL7Mzad5neXxGY7aX3s/Rxly5MV4nHjnJMx2QOqpP6eilemanytpjmp8rZx0XChpmp
8rY6ZqfK2Bclemanytpjmp8rYFwh6ZqfK2OmanytgZ1dsujy87j68dvarPzU6bV4tRX8M7W+NZS5dTny45
pbSTtLyYm1L71mazEg/S5PZj6w083R6zJmiKXrM7T7T0gAAyZqYacrJO0OPSGG14rPKrv2TMKb0reNr1
iY7etBrbV1GSunwxE2id5mP8ASCjNrcWG3Jne1vIEN4NTj1ETyJ647YktGLBWcloiOrnn5ptFWcmoyank8
mluqsfMFwACPU++n6LEep99P0HNenIBHnfYibTER2y6cxk+TOL3tfq65pyRI/Dvt8Nh1ERa8uePDNr7TG2
3a+3w2rbaOyZ2h3tO2anzmOtynldlffbldqayxbDesTMx1Q5u+pvblzXfq+TgOKoiJ/Ajtj6j5eZrWzjN5j4C
Rt6Udgg6bqfK2fem6nylllepclm6nyljpup8pYFwh6bqfKWom6nylgXChpup8pY6bqfKWB11mO8053D
Mxkp19XxhyOnpKmXamX8F/2l86bqfKWeZq4nn5tOocc26+SD9F2s4vY/q8bRa7NjtGPacIz+Hxejh68cS
DYADz65Nzly5a45rFaTt+KHoOGszcxp7Wj2p6o+oJtPqs3KyzmtWaY465iPi+0vrNRWMljps9kT8Wbaea
ei7R/rmoVLGLT1vpIyzqLRal+E9UA9LFy+brzm3L269mk3o/LfLpa2v29m/zUgOeo9zZ0c9R7mwlgEeUbri
vau8R1MKcfKjTTye0WmLuNsV6RvMdTpOnnkRMe01hm00tzm+35lptOnrMTO47imHKuG9o3iGLVmtt
p7VFZmummeyd08zMzvM7yOZih8AHLvpPbt9FTz4zZMvt8eKckzHXEfBrpmp8rZxoo0uEPTNT5Wx0zu
+VsOlwh6ZqfK2OmanytgXChpmp8rY6ZqfK2Bcg1dsujy9i68dvar8N33pmp8rZjJqc+THNLaSzIYBTptZi1
Efhna3xrLtk7K/qh+b3tjv1b1mjenotZkzTXHesztMfiB6YADnnpfJj5NL8i3zdGM2amCnKyTtG+wPNyDJU
Vw01Nr2nt/JXlw6Nfk1yxSkR2x2y1cabk2y4cv/AMszvG0u2bXThwY6ztOa1Y3j5AabJlx6ydPkvzbboxPy
XltFGKlpotOWt81+2d1oAAItR76XN01HvrOaPPX/Q1Ss3tFY+LLpg99Ucxtro1vnBiw73mL/D4NZKZJv7xyt8

HSJpFY369usaYxdxvgmLxETHX2P18FqVm0zHU3Wto1ETMdW8saiZnLaN+oSYi13IAcD0KexH0ee9CnsR
9Fa8b6ANQAAA
AAAAAAAAAAABz6Rh8Wn90N39i30eZ6P0mHPp+Xkrvb1THaD0OkYfFp/dB0jD4tP7ocfv2m8P9Ac9Xabw/3B26Rh8
/AHB26Rh8Wn90HSMPI0/uhx9Xabw/3PV2m8P9wdukYfFp/dB0jD4tP7ocfv2m8P8Ac9Xabw/3B26Rh8
Wn90Pls+COTE5aTE/7ocvV2m8P9z1dpvD/AHB5mW9tHqZnBkiaz1xtO8PR0npDHn2rf8F/2l5/pGmDFe
MeGu1o7Z3Z0uhy6iYn2afOQe3T27tuWCnNxNN5ttt1y6gAA1zU1cZZvhvVms/6bfBzrpM98ls2XJWMnj2
ryfguAQ2x6+1ZxzfHyZ6uV8VWmwXgw1xxO+3xdAAAHy/sT9Hnw9C/sT9HnwM+TQAjF3xXimC20xFt+xi
+a9o26oj8nMFym1nSmW1I2js/MnLaa2jq6+1zAvIAIO2I97P0cXbS+9n6Dqj+IYCvQAAAAAAj9I6nmsfN0
9u/y+CTSejLZNr59617vx16NNNSMs5LfivPxn4OwOU46YscVpWKxEx2OrOX2Y+rQAAOOrrmvhmuCYi0/
GfkjwafW6es1pGLrneZntl6QDzdRptZnyVm3NzWv+nfQuaeNXF4jLGOMf8AtVAAACPU++n6LEep99P0H
NenIBHnaxzEZKzPV1u2TPat5isxMJwWKpiGuXab8rfrbnPeduzqcgLy1e03tvPayAgR2wEdsBG3ox2AK9Q
AAAAADnnzVwYrXt2R+7x8emz6/NOS3VWZ67T/Alevl09c14nj11r2VdYiljal2gHDT6TFp67Ujr+Np7XXF7
H9WmcXsf1BoAbYzaemaazeJnkzvEbouBMRMbfBLb0dp7W5XImPyiepUA+VrFKxWsRHZD6ADnqPc2
dHPUE5sCIBHIFFMnlwdUxyt+xOCxNnS+a942mdo/lpmtno22/NzAvLc5bTSazttLACTNwAR30nt2+ipLp
Pbt9FSvTRoAHQAAAj9I6nmcxlp7y/VH5LHGumpGWctvxXn4z8AebpPRlskxfPvWvy+MvT5umKla0rER
Fo7HVnj2V/VANAAPlq1vXa0RMfKX0BxrPMFbcqkUsT9H3JpcOS82vjra0/GXUBxppcGO0WpirEx2TDsA
AAIs/vrObpn99ZzR56/6G8VorkiZ7IYBzH463zW5U8m3U57zvv1/N8AmZluct5mj5XYzaZtO89r4BeQAD
4NdPyV6ui5J26t/8A2GXoU9iPorXjQ+sMnlMn/v8AQ9YZPKZP/f6LwaoPWGTymT/3+h6wyeUyf+/0XgIP
WTGymT/3+h6wyeUyf+/0XgIPWGTymT/3+h6wyeUyf+/0XgIPWGTymT/3+hHppG9YmbaXJER2zP/8Axe
46z+Dy/pkHTHkrllxek71nsaT+j/4LF9FAAAAAAAAAAAAAAAAAADGTJyLVjb
fITs58vJlvMY5itY+MsZOcjjjjtP4uqYBUOWXJaLxSm3Kn4z8GZtlxbWtaL1+P5A7jlyzE1rSn7W7Gbc/jrNp
tFo+MbA7jjGS3RuXv+LbtdMczbHWZ7ZgGhyx3tbJki09VZ6mYvky7zS0Vr8PzB3HLHkty5x5NuVtvEx8X3F
e1r5lmeqj6gdByy3tXJSInqmesyZLcuMePblbbzM/AHVnJeMdJtLIN8mLab2i1fj+T7n/ABWx1+EyDVsvJ5
HKrtFv2dHLUvhnh8ut0p09Kz84B9AAAAAAAB8v7Fvoj9E/wn/wCpWX9i30R+if4T/wDUgtAAAAAAAct
RktThtSOveeqldQEGm9HRFuczzyrz17fbfEREbRG0ADNPbu0zT27tAAAAAAA+X9ifo8+HoX9ifo8+Bny
aAEYgAAAAADtpfez9HF20vvZ+g6o/pWAr0AAAAAAAM5fZ/q0zl9n+rQAAACPU++n6LEep99P0
HNenIBhNAAAAAACO2AjtgI29Gow17BXqAAAAAAAGcXsf1aZxex/UGgAAAAAAHPUE5s6Oeo9zYEQC
PKAAAAAAA76T27fRUI0nt2+ipXoo0ADoAAAAAAydlf1Q2xk7K/qgGwAAAAAAAARZ/fWc3TP76zmj
z1/0ADkAAAAAAehT2I+jz3oU9iPorXjfQBqAAAAAOs/g8v6ZdnHWfweX9Mgz6P/gsX0UJ/R/8ABYvoo
AAAAAAAAAAAAAAAAAAAAABPjtGG1q360md4lnjkjlx8m0qlr1qpij7YfNo+UA
4Z6RGWL2rM122n8mdtPMxFazMz8IRERHZEA4ZY5vJTJEb1iNpMmes0mK7zMu7NqRNJiliJmAcqVm2k
2jtmDHnrXHEW3i0Rtts6YqzTHFZ7Ya5Mb77QDhp55V8s7bbudK46b1y1mj4/NztEfAmIntjcHHDXFN5
mlZ6vizW/M5r8vqi3XEqljbsJij7Y3BNkyRkzY+T2RPaZqRGblXrM0mOCnaPiACWI08zEvrmzPydM34b4p
+EtS7RER2RDGanOU2jt7YB81E7YbNY42x1j8mL0tkikTttHXZ1AAAAAAAB8v7Fvoj9E/wAJ/wDqVl/Y
t9EfOn+E/wD1ILQAAAAAAAC9+Te29bTv8AKH3nP9tuDZM7RMgxzn+23A5z/bbg511FrXvXFmx9XTH
e14nlUmv1A5z/AG24PvOf7bcGgGec/wBtuBzn+23BoBnnP9tuBzn+23BqZ2jdml4yVi0dkg+WvvWY5Nu
z5JOaydyVwJNN0PNZO5Y5rJ3LLgcdcleady3A5rJ3LcFwHXCHmsnctwOaydy3BcB1wh5rJ3LcDmsnctwX
Adcleady3B101LVyTNqzEbFSCxRETcAHYAAAAAADOTfk9Ub9Zzn+23BoBnnP9tuBzn+23BoBnnP9tu
Bzn+23BoBnnP8AbbGc5/ttwaAZ5z/bbGc5/ttwaAZ5z/bbGmzUvfJyopbb6KwSYvFkPNZO5bGc1k7luC4H
HXCHmsnctwOaydy3BcB1wh5rJ3LcDmsnctwXAdcleady3A5rJ3LcFwHXCHmsnctwOaybx+CVwL1wQA
OwAAAAAAABzpbkxtNbvdvdAGec/224HOt7bcGgGec/wBtuBzn+23BoBnnP9tuBzn+23BoBnnP9tuBzn+
23BoBnnP9tuDGW03xzWK23n8nUBDzWTuW4HNZO5bguBn1wh5rJ3LHNZO5ZcB1wh5rJ3LHNZO5ZcB
1wh5rJ3LHNZO5ZcB1wh5rJ3LHNZO5ZcB1wn01LvvabVmOr4qAhCraLAoAAAAAAzkiZiNo32mJaAZ5V
/DnjByr+HPGGgGeVfw54wcq/hzxhoBnlX8OeMHkv4c8YaAZ5V/DnjByr+HPGHoC88u1a45tyfzaT7cm
1ZpaezcG+Vfw54wcq/hzxhoBLkw5L5jtFe382ej5e7HFYDmallH0fL3Y4nR8vdjisBMIR9Hy92OJ0fL3Y4rA
MIR9Hy92OJ0fL3Y4rAMIR9Hy92OJ0fL3Y4rAMIR9Hy92OKusbViPyfQdRTEaABQAAAABx1n8HI/TLs46
z+Dy/pkGfR/8Fi+iHP6P/gsX0UAAAAAAAAAAAAAA+X9j30R+if4T/APUrI+xb6I/RP8J/+pBaAAAAAA+W9ifo+vlo3rMfkCbT3vVx
AAAAAAAAAAAAAA+X9j30R+if4T/APUrI+xb6I/RP8J/+pBaAAAAAA+W9ifo+vlo3rMfkCbT3vVx

RFcXKj577N5bWtprTavJn5b7t6ek48UVt2vuas3xWrHbIOdK83i52Zm1uT8zfMeCMIvktabW6+3sdorvj5
NvltLIWubHHJrybR8Jn4A+Vtbm8tLTvyfi+YsEXxVte1pmY6uvbsrhmuK8b73t2t4qzTFWs9sQDljjnMFq3
mZ5MzG5pMdYpW/+r6umHHNa3i3+q0yZhrkx7UmI5MfHcHYAAAAAAAAAAAAAAAAAAAAAA
AAEtMtMe
fLyp23n5Ptrxny0ikTtWd5l0xY5rlyWtEbWnqJx2rm5dOyfagHHJkrbNauS1orXqil+L7hvWMsVxzaaT8J+d
pbHeuSb49uvtiWqc7Mzy9oj5QDjfmOPVKva0/9NYLzbT33nfbelkpTLj3rWKzEz2y1ixWpivW3bMzIM6b
HvSt7TMz8OvsUMYKzTFWtu2GwAAAAAAAAAAAAAHHWfweX9MuzjrP4PL+mQZ9H/wAFi+ihP6P/
AILF9FAAP
I/Yt9Ef0n+E//UrL+xb6I/RP8J/+pBaAA
AAAAAAAAAAAAAAAAAAAAAA46z+Dy/pl2cdZ/B5f0yDPo/+CxfRQn9H/wWL6KA
AAAAAAAAAAAAAAAAAAfL+xb6I/RP8Acf8A6IZf2Lf5Wi
1fR8PltiyTO+/VAPWEPrOvgZOB6zr4GTgC4Q+s6+Bk4HrOvgZOALhD6zr4GTges6+Bk4AuEPrOvgZOB6zr
4GTgC4Q+s6+Bk4HrOvgZOALhD6zr4GTges6+Bk4AuEPrOvgZOB6zr4GTgC4Q+s6+Bk4HrOvgZOALhD6z
r4GTges6+Bk4AuEPrOvgZOB6zr4GTgC4Q+s6+Bk4HrOvgZOALhD6zr4GTges6+Bk4AuEPrOvgZOB6zr4G
TgC4Q+s6+Bk4HrOvgZOALhD6zr4GTges6+Bk4AuEPrOvgZOB6zr4GTgC4Q+s6+Bk4HrOvgZOALhD6zr4
GTges6+Bk4AuEPrOvgZOB6yny2XgC4Q+sbeWy8D1jby2XgC4Q+sbeWy8D1jby2XgC4Q+sbeWy8D1jby
2XgC4Q+sbeWy8D1jby2XgC4Q+sbeWy8D1jby2XgC4Q+sbeWy8D1jby2XgC4Q+sbeWy8D1jby2XgC4Q+s
beWy8D1jby2XgC4Q+sbeWy8D1jby2XgC4Q+sbeWy8D1jby2XgC4Q+sbeWy8D1jby2XgC4Q+sbeWy8
D1jby2XgC4Q+sbeWy8D1jby2XgC4Q+sbeWy8D1jby2XgC4Q+sbeWy8D1jby2XgC4Q+sbeWy8D1jby2X
gC4Q+sbeWy8D1jby2XgC4Q+sbeWy8D1jby2XgC4Q+sbeWy8D1jby2XgC4Q+sbeWy8D1jby2XgC4Q+s
beWy8D1jby2XgC4Q+sbeWy8D1jby2XgC4Q+sbeWy8D1jby2XgC4Q+sbeWy8D1jby2XgC4Q+sbeWy8D1j
by2XgC4Q+sbeWy8D1jby2XgC4Q+sbeWy8D1jby2XgC4Q+sbeWy8D1jby2XgC4Q+sbeWy8D1jby2XgC4
Q+sbeWy8D1jby2XgC4Q+sbeWy8D1jby2XgC4Q+sbeWy8D1jby2XgC4Q+sbeWy8D1jby2XgC4Q+sbeW
y8D1jby2XgC4Q+sbeWy8D1jby2XgC4Q+sbeWy8D1jby2XgC4Q+sbeWy8D1jby2XgC4Q+sbeWy8D1jby
2XgC4Q+sbeWy8D1jb4aXLP9AXCH1hl8nl/f8A8HrDL5PL+/8A4BclfWGXYeX9/wDwesMvk8v7/wDgFwh
9YZfj5f3/APB6wy+Ty/v/AOAXCH1hl8nl/f8A8HrDL5PL+/8A4BclfWGXYeX9/wDwesMvk8v7/wDgFwh9Y
Zfj5f3/APB6wy+Ty/v/AOAXCH1hl8nl/f8A8HrDL5PL+/8A4BclfWGXYeX9/wDwesMvk8v7/wDgFwh9YZf
j5f3/APB6wy+Ty/v/AOAXCH1hl8nl/f8A8HrDL5PL+/8A4BclfWGXYeX9/wDwesMvk8v7/wDgFzjrP4PL+
mU/rDL5PL+//hzz6zLlw3x9Eyxyo232n/wCr0f/AAWL6KE+gi0aPHFqzWYjbaVAAAAAAAAAAAAAAA
AA
AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAP//Z"></figure><p>Constraints:</p><p>0<size
<100</p><p>0<data<1000</p><p>Input format:</p><p>First line
indicates the size of the queue</p><p>Second line indicates the elements of the
queue.</p><p>Output Format:</p><p>First line indicates the inserted elements
in queue using linked list concept</p><p>Second line indicates the one element dequeued from the
queue using linked list concept.</p>

```
#include <stdio.h>  
  
#include <stdlib.h>  
  
struct node *f = NULL;
```

```

struct node *r = NULL;

struct node
{
    int data;
    struct node* next;
};

void enqueue(int d)
{
    struct node *n;
    n = (struct node*)malloc(sizeof(struct node));
    if(n==NULL){
        printf("Queue is Full");
    }
    else{
        n->data = d;
        n->next = NULL;
        if(f==NULL){
            f=r=n;
        }
        else{
            r->next = n;
            r=n;
        }
    }
}

int dequeue()
{
    int val = -1;
    struct node* t;
    t = f;
    if(f==NULL){

```

```

printf("Queue is Empty\n");

}

else{
    f = f->next;
    val = t->data;
    free(t);
}

return val;
}

int main()
{
    int n,i,t;
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        scanf("%d",&t);
        enqueue(t);
    }
    for(i=0;i<n;i++){
        printf("%d\n",dequeue());
    }
    return 0;
}

```

question

<p>Question description</p><p>Sathya is an DS expert training youngsters struggling in DS to make them better.</p><p>Sathya usually gives interesting problems to the youngsters to make them love the DS.

One such day Sathya provided to the youngsters to solve the task such that, insert an element in a Queue in FIFO order</p><p>Youngsters were lacking the idea to solve the problem.</p><p>Being an exciting youngster can you solve it?</p><p>Function Description</p><figure class="image"></figure><p>Constraints:</p><p>0<size
 <100</p><p>0<data<1000</p><p>Input format:</p><p>First line
 indicates the size of the queue</p><p>Second line indicates the elements of the
 queue.</p><p>Output Format:</p><p>every lines indicates the enqueue of each
 elements</p><p>last line indicates the fine queued elements.</p>

answer

```

#include <stdio.h>

#define SIZE 100

void enqueue(int);

void display();

int items[SIZE], front = -1, rear = -1;
  
```

```
int main() {
    int n,data,i;
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        scanf("%d",&data);
        enqueue(data);
        display();
    }
    return 0;
}

void enqueue(int data) {
    if (rear == SIZE - 1)
        printf("Queue is Full!!!");
    else {
        if (front == -1)
            front = 0;
        rear++;
        items[rear] = data;
        printf("Enqueuing %d\n", data);
    }
}

void display() {
    if (rear == -1)
        printf("\nQueue is Empty!!!");
    else {
        int i;
        for(i=front;i<=rear;i++)
        printf("%d ", items[i]);
    }
}
```

question

<p>Question description</p><p>There is a bit string consisting of n bits. Then, there are some changes that invert one given bit. Your task is to report, after each change, the length of the longest substring whose each bit is the same.</p><p>
Constraints</p>1≤n≤2·10^51≤m≤2·10^51≤xi≤n<p>
Input Format

The first input line has a bit string consisting of n bits. The bits are numbered 1,2,...,n.
The next line contains an integer m: the number of changes.
The last line contains m integers x1,x2,...,xm describing the changes.

Output Format
After each change, print the length of the longest substring whose each bit is the same.

 </p>

answer

```
#include <stdio.h>
#include <string.h>

#define N 200000
#define M (1 << 18) /* M = pow2(ceil(log2(N))) */

int max(int a, int b) { return a > b ? a : b; }

char cc[N + 1];
int pp[M * 2], qq[M * 2], tr[M * 2];

void pull(int k,int l,int r) {
    int m = (l + r) / 2, k1 = k * 2 + 1, k2 = k * 2 + 2;

    pp[k] = pp[k1];
    if (pp[k1] == m - l && cc[l] == cc[m])
        pp[k] += pp[k2];
    qq[k] = qq[k2];
    if (qq[k2] == r - m && cc[r - 1] == cc[m - 1])
        qq[k] += qq[k1];
```

```
    tr[k] = max(tr[k1], tr[k2]);  
    if (cc[m - 1] == cc[m])  
        tr[k] = max(tr[k], qq[k1] + pp[k2]);  
}
```

```
void build(int k, int l, int r) {
```

```
    int m;
```

```
    if (r - l == 1) {  
        pp[k] = qq[k] = tr[k] = 1;  
        return;  
    }  
    m = (l + r) / 2;  
    build(k * 2 + 1, l, m);  
    build(k * 2 + 2, m, r);  
    pull(k, l, r);  
}
```

```
void update(int k, int l, int r, int i) {
```

```
    int m;
```

```
    if (r - l == 1) {  
        cc[i] = cc[i] == '0' ? '1' : '0';  
        return;  
    }  
    m = (l + r) / 2;  
    if (i < m)  
        update(k * 2 + 1, l, m, i);  
    else  
        update(k * 2 + 2, m, r, i);  
    pull(k, l, r);
```

```
}
```

```
int main() {
    int n, m;

    scanf("%s%d", cc, &m);
    n = strlen(cc);
    build(0, 0, n);
    while (m--) {
        int i;

        scanf("%d", &i), i--;
        update(0, 0, n, i);
        printf("%d ", tr[0]);
    }
    printf("\n");
    return 0;
}
```

question

<p>Question description</p><p>Given an array of n integers, your task is to process q queries of the form: what is the sum of values in range [a,b]?
Constraints</p>1≤n,q≤2·10^51≤xi≤10^91≤a≤b≤n<p>Input
The first input line has two integers n and q: the number of values and queries.
The second line has n integers x₁,x₂,...,x_n: the array values.
Finally, there are q lines describing the queries. Each line has two integers a and b: what is the sum of values in range [a,b]?

Output
Print the result of each query.
 </p>

answer

```
#include<bits/stdc++.h>
using namespace std;
```

```

int main(){
    int n,q,i,a,b;
    cin>>n>>q;
    int x[n];
    for(i=0;i<n;i++)
        cin>>x[i];
    while(q--){
        int sum=0;
        cin>>a>>b;

        for(i=a;i<=b;i++)
            sum=sum+x[i-1];
        cout<<sum<<endl;
    }
}

```

question

<p>Question description</p><p>You are given an $n \times n$ grid representing the map of a forest. Each square is either empty or contains a tree. The upper-left square has coordinates $(1,1)$, and the lower-right square has coordinates (n,n) .

Your task is to process q queries of the form: how many trees are inside a given rectangle in the forest?

Constraints</p> $1 \leq n \leq 1000$ $1 \leq q \leq 2 \cdot 10^5$ $1 \leq y_2 \leq n$ $1 \leq x_1 \leq x_2 \leq n$ <p>
Input

The first input line has two integers n and q : the size of the forest and the number of queries.
Then, there are n lines describing the forest. Each line has n <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>n</mi></math> characters: . is an empty square and * is a tree.
Finally, there are q lines describing the queries. Each line has four integers y_1, x_1, y_2, x_2 corresponding to the corners of a rectangle.

Output
Print the number of trees inside each rectangle.
 </p>

answer

```

#include<bits/stdc++.h>
using namespace std;

```

```

#define rep(i,a,b) for (int i=a; i<b; ++i)

int dp[1005][1005];

int main(){

    int n,m; cin>>n>>m;

    rep(i,1,n+1){

        rep(j,1,n+1){

            char x; cin>>x;

            dp[i][j] = (dp[i-1][j] - dp[i-1][j-1]) + dp[i][j-1] + (x=='*');

        }

    }

    while(m--){

        int y1 , x1, y2, x2; cin>>y1>>x1>>y2>>x2;

        cout<<dp[y2][x2]+ dp[y1-1][x1-1] - dp[y2][x1-1] - dp[y1-1][x2]<<endl;

    }

    return 0;

    cout<<"for(i=1;i<=n;i++)";

}


```

question

<p>Question description</p><p>Given an array of n integers, your task is to process q queries of the form: what is the minimum value in range [a,b]?</p><p>Constraints</p>1≤n,q≤2·10^51≤xi≤10^91≤a≤b≤n<p>

Input

The first input line has two integers n and q: the number of values and queries.
The second line has n integers x1,x2,...,xn: the array values.
Finally, there are q lines describing the queries. Each line has two integers a and b: what is the minimum value in range [a,b]?

Output
Print the result of each query.

 </p>

answer

```
#include <stdio.h>
```

```
#define N 200000
```

```
#define N_  (1 << 18) /* N_ = pow2(ceil(log2(N))) */  
#define INF  0x3f3f3f3f
```

```
int tt[N_ * 2];
```

```
void build(int *aa,int k,int l,int r) {  
    int m, k1, k2;  
  
    if (r - l == 1) {  
        tt[k] = aa[l];  
        return;  
    }  
    m = (l + r) / 2, k1 = k * 2 + 1, k2 = k * 2 + 2;  
    build(aa, k1, l, m);  
    build(aa, k2, m, r);  
    tt[k] = tt[k1] < tt[k2] ? tt[k1] : tt[k2];  
}
```

```
int query(int k,int l,int r,int ql,int qr) {  
    int m, q1, q2;  
  
    if (qr <= l || r <= ql)  
        return INF;  
    if (ql <= l && r <= qr)  
        return tt[k];  
    m = (l + r) / 2;  
    q1 = query(k * 2 + 1, l, m, ql, qr);  
    q2 = query(k * 2 + 2, m, r, ql, qr);  
    return q1 < q2 ? q1 : q2;  
}
```

```

int main() {
    static int aa[N];
    int n, q, i, j;

    scanf("%d%d", &n, &q);
    for (i = 0; i < n; i++)
        scanf("%d", &aa[i]);
    build(aa, 0, 0, n);
    while (q--) {
        scanf("%d%d", &i, &j), i--;
        printf("%d\n", query(0, 0, n, i, j));
    }
    return 0;
}

```

question

<p>Question description</p><p>There are n children, and each of them independently gets a random integer number of candies between 1 and k.

What is the expected maximum number of candies a child gets?</p><p>Constraints</p>1≤n≤1001≤k≤100<p>
Input

The only input line contains two integers n and k.

Output

Print the expected number rounded to six decimal places.

 </p><p>
 </p>

answer

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int N, K;
double ans, a, b;
```

```

int main(){
    scanf("%d %d", &N, &K);
    for(int i = 1; i <= K; i++){
        a = b = 1.0;
        for(int j = 1; j <= N; j++){
            a *= (double) i / K;
            b *= (double) (i-1) / K;
        }
        ans += (a-b) * i;
    }
    printf("%.6f\n", ans);
    return 0;
    cout<<"double power(double a,int k)";
}

```

question

<p>Question description</p><p>A company has n employees, who form a tree hierarchy where each employee has a boss, except for the general director.
Your task is to process q queries of the form: who is the lowest common boss of employees a and b in the hierarchy?

Input

The first input line has two integers n and q: the number of employees and queries. The employees are numbered 1,2,...,n, and employee 1 is the general director.
The next line has n-1 integers e₂,e₃,...,e_n: for each employee 2,3,...,n their boss.
Finally, there are q lines describing the queries. Each line has two integers a and b: who is the lowest common boss of employees a and b?

Output
Print the answer for each query.

Constraints</p>1≤n,q≤2·10⁵1≤e_i≤i-11≤a_{,b}≤n

answer

```

#include <stdio.h>

#define N 200000
#define LN 17 /* LN = floor(log2(N - 1)) */

```

```

int main() {
    static int dd[N], pp[LN + 1][N];
    int n, q, p, i, j, k, tmp;

    scanf("%d%d", &n, &q);

    for (i = 1; i < n; i++) {
        scanf("%d", &p), p--;
        dd[i] = dd[p] + 1;
        pp[0][i] = p;
    }

    for (k = 1; k <= LN; k++)
        for (i = 0; i < n; i++)
            pp[k][i] = pp[k - 1][pp[k - 1][i]];

    while(q--) {
        scanf("%d%d", &i, &j), i--, j--;
        if (dd[i] < dd[j])
            tmp = i, i = j, j = tmp;
        if (dd[i] != dd[j])
            for (k = LN; k >= 0; k--)
                if (1 << k <= dd[i] - dd[j])
                    i = pp[k][i];
        if (i != j) {
            for (k = LN; k >= 0; k--)
                if (1 << k <= dd[i] && pp[k][i] != pp[k][j]) {
                    i = pp[k][i];
                    j = pp[k][j];
                }
            i = pp[0][i];
        }
        printf("%d\n", i + 1);
    }
}

```

```
    }  
    return 0;  
}
```

question

<p>Question description</p><p>A forest is an undirected graph without cycles (not necessarily connected).</p><p>Mohana and john are friends in kerala, both of them have a forest with nodes numbered from 1 to n , and they would like to add edges to their forests such that:</p>After adding edges, both of their graphs are still forests.They add the same edges. That is, if an edge (u, v) is added to Mohana's forest, then an edge (u, v) is added to john's forest, and vice versa.<p>Mohana and johns want to know the maximum number of edges they can add, and which edges to add.</p><p>Constraints:</p><p>1 ≤ n ≤ 105, 0 ≤ m_1 < m_2 < n ≤ n , $1 \leq u, v \leq n$, $u \neq v$ </p><p>Input</p><p>The first line contains three integers n , m_1 and m_2 — the number of nodes and the number of initial edges in Mohana's forest and john's forest.</p><p>Each of the next m_1 lines contains two integers u and v — the edges in Mohana's forest.</p><p>Each of the next m_2 lines contains two integers u and v ($1 \leq u, v \leq n$, $u \neq v$) — the edges in Johns's forest.</p><p>Output</p><p>The first line contains only one integer h , the maximum number of edges Mohana and John can add.</p><p>Each of the next h lines contains two integers u and v ($1 \leq u, v \leq n$, $u \neq v$) — the edge you add each time.</p><p>If there are multiple correct answers, you can print any one of them.</p>

answer

```
#include<bits/stdc++.h>  
using namespace std;  
  
typedef long long ll;  
  
int fa[1005],fa2[1005],n,m1,m2;  
  
int gf(int x,int *f){  
    return f[x]==x?x:f[x]=gf(f[x],f);  
}  
  
int main(){  
    cin>>n>>m1>>m2;  
    for(int i=1;i<=n;i++)fa[i]=fa2[i]=i;  
    for(int i=1,x,y;i<=m1;i++)cin>>x>>y,fa[gf(x,fa)]=gf(y,fa);  
    for(int i=1,x,y;i<=m2;i++)cin>>x>>y,fa2[gf(x,fa2)]=gf(y,fa2);
```

```
cout<<n-max(m1,m2)-1<<'\n';
for(int i=1;i<=n;i++){
    for(int j=i+1;j<=n;j++){
        if(gf(i,fa)!=gf(j,fa)&&gf(i,fa2)!=gf(j,fa2)){
            cout<<i<<' '<<j<<'\n';
            fa[gf(i,fa)]=gf(j,fa);
            fa2[gf(i,fa2)]=gf(j,fa2);
        }
    }
}
return 0;
cout<<"while(m1--)";
}
```

question

8+PICYIV3V2PEJqTXczPqNYoXVxhOLi3ifyA1go1Djbp8xuUIt+0mXV8K4rdu4Lj3gdAK3qaVLa7Y58wLOQ
TTXAq1LzpbGvCyGg9zr/X92BoBXLUUwItlZFPuyWJqSTTT7UABCy6ur25qPmzsLIWLMJKS+TAIIIN4yDFp
fftR5/wAmt2wU9jmt3dniBIEYWQsTcJKWO5hWQc3BSW5c1niBIEZWQjRIJy5JvmVQivS5y67c8fh9wF
4I2W11+3NR82K7YWLJMJKXkwJA5Oca45nJRXzI13V2exNS8mBMAAAAAAAAAAAAAAAyWStlqZwhY
4pjAawY9t/xhtv8AjBLjYDHtv+MNt/xgXGwGPbf8Ybb/AlwLjYDHtv8AjDbf8YFxsBj23/GG2/4wLjYDHtv+
MNt/xgXGwGPbf8Ybb/jAuNgMe2/4w23/ABgXGwGPbf8AGG2/4wLjYDHtv+MNt/xgXGwGPbf8Ybb/Al
wLjYDHtv8AjDbf8YFxsBj23/GG2/4wLjYDHtv+MNt/xgXGwGPbf8Ybb/jAuNgMe2/4w23/ABgXGwGPbf8
AGG2/4wLjYDHtv+MNt/xgXGwGPbf8Ybb/AlwLjYDHtv8AjDbf8YFxsBj23/GG2/4wLjYDHtv+MNt/xgXGw
GPbf8Ybb/jAuNgMe2/4w23/ABgXGwGPbf8AGG2/4wLjYDHtv+MNt/xgXGwGPbf8Ybb/AlwLjYDHtv8Aj
Dbf8YFxsBj23/GG2/4wLjYDHtv+MNt/xgXGwGPbf8Ybb/jAuNgMe2/4w23/ABgXGwGPbf8AGG2/4wLjY
DHtv+MNt/xgXGwGPbf8Ybb/AlwLjYDHtv8AjDbf8YFxsBj23/GG2/4wLjYDHtv+MNt/xgXGwGPbf8Ybb/j
AuNgMe2/4w23/ABgXGwGPbf8AGG2/4wLjYDHtv+MNt/xgXGwGPbf8Ybb/AlwLjYDHtv8AjDbf8YFxsBj
23/GG2/4wLjYDHtv+MNt/xgXGwGPbf8Ybb/jAuNgMe2/4w23/ABgXGwGPbf8AGG2/4wLjYDHtv+MNt/
xgXGwGPbf8Ybb/AlwLjYDHtv8AjDbf8YFxsBj23/GG2/4wLjYDHtv+MNt/xgXGwGPbf8Ybb/jAuNgMUIfG
LfW8lk0aaTnRGUnlvPH9Qq0AAAAAAAAACrVTIDTTID2ki0NZWGB4VVk4WqUW92fqe6VQ0tMJ741p
SLQAAAPkYui/wACf5v4Nr5GLov8Cf5v4A70p7tH86/Zkno6VpmtuZbc7u3JhpT3aP51+zNM/wACX5f4Ay
UWyh0ZKSfGOuinTz0kav8AeW6b5trJdpK+t6PIDvbwRo1MKK+qvralH5cwJdHTXWW1xbcOccmy2Wyq
c/CmyrS2u7dLqtkf7X3l1kd9cov+5NAYtFp4WVO21b5Sb5nKK41dJShDlt4HNpqfRlum+Mk4vg0NPY7ek
XNxccx4J9wE6/6rZ+X+EVwphd0hap8UuOO8sr/qtn5f4Q0/9Rv8gla9xr6qhepxzljlXz0cqWqlia5PBo11U
267oR3OD4oi9dU4+pU3N8o4ArvunPQ0pPjY8N9+DS9FT1OxRWce125lauqd2mhKMNs48dqOf6hF14U
JdZ4cd0CWncNDJW7ZShF7X3Euj64LTQmorc85fbzJJXT0U1bxslF8Cvo6+LrVOGpxz+4ENDLZTfLwtv7Hd
FRC6ErrVvIJ9o6PipVXRfJvBGi/wBD3U3xaSeYtLmB3atLr4KHCFi4o5KqN3Sk4z9lJPHfwRKvdq9ZG7a1XDI
ntO1/1Wz8v8ICGsqhRZTOuO17uw9AxdJcqvzG0DFrW7dRVp84i+LJ36Op0S2QUZRWU0c1tc1ZXFwTzhz
XyIW66NITHVGTnJYxjkBOhrUaHNiUnFNcSPRtcHSpuK3KT4ltFTp0bjL2sNsj0b7r/wCzA1NKSaaynwaPPh
TW+kpwci7VhgscOSPQPPssWn6RlZNPbKPBrA7roKepohyT4cDRPTVwonGD6pNcZFOperdZpmu0t18JT
00IBNvKeF2gZd2ijU4JOTx7SjxyT0sXf0dOD4tNqP7ivWwVCrrqlvxjCXaT6Lf+xJdu7+EBynUY6Ocs+tBbf+C
CzpujcrhKfb5//RXbTL0x0L2LJKX/AH7m7V0u3TyhHmuKAxVS0UaVGabk1xeO0lpZ50eoohnKinjywyVWsr
rrULq2pxWH6vMtVkrhdJ19Wtrx81gCrQaaqVKsnHdJvt7DmpShrd98XKprh3lV6P8Adl+b/cjbqJ0xtXRz
S+TSAjVXRZfG3T2KOoCuuZLVwp66NI9nqpcIFCcLtZXLTQcVF+s8YRLUPqderbIOUGuHyAqvsp6yueni4y
T48MJmjX/j6b838oo1mpV6g4wkoRftNc2X6/8AG035v+ANph6UhFVRmorc5Yb/AENxj6U93j+f+GBO/b
ptLOUVFtLkR0ukqenjKcVKU1ltl19XXaeUFza4GajWKitVXRIGcOHLmb3S5o1llCeYYiyvS0Qu1N7sWVGX
Bfqy3SQnZqj6mcndlwmic0H4+p/N/LA2RioxUYrCXJGPW+9ab838o2mLW+9ab838oCOTgrNbTB8muJ3
XaaqvT7oQUXFrijuo/qNHkWdl+6S80BVrlZPSVRT42JZ+hctFT1Wzbxx7XaV30yt0NThxIGKaXfwOf6jDq+
MJDzY247QOaScvRr6pPPVpojoNNVOnrJx3PP0J6amVeltINYINN4/Qn0b7qvNgVRgtVr7Os4xr4JEddRXV
KuUI7cvDSJTk9HrJWSi3XZ2or1eoWodeyMtl7TXNgWdl2Lra6pNqHOWCrUT0kqcVlbNcmkaNdXNTrvh
Hds5r5HHrqnHFdTIN8o4Ar1k3ZoaZPm3x+hsp01VGHCPYxnvm3SOXpa247Xniu7gbgOS9l+Rj6OTejS
eEtzx9EbJey/lxdH7vQ7NmN254z34QFOnenrUoamDVmeLkjzo6VVGThZvhJ5WOWpWrhscNVW967Nv
M70dGS6yW1xrk/VTA2nnr0SmUIZLrp8XjJvmm4SS4NrgedpdRDSxICyuSnnsQEtBKPPdsaa8qtrKTOVu
u1129ZUXnB3STc+kLJSjtco8n+hPSe/ajz/kCN1caNbQ61t3PDGSqgtTQlBLdL1uHPiiW98035v5Q6QzCy
i3DcYy4/YCWvhGGjcYpJZXBEnqFTVUnCcsWxos1lsbtDvhndfaaqfwa/wAq/YDLqdU5aSTjGUG5bePAsq
0VKoUZQTbXF9pLWUU7TuMfaXFNevjGtRshJWRWMY5gdVM6NDbCclJYeMdxCFjq6KUo8HxS+pZm6
Whtdwb18LHJEaand0WoLm84+oEtNpKvR4ucFKUlltkdJmnVW0ZzFLcjIOtjTUq7oyU4LGMcyWjhOd1m
onHbu4JPuAhoqo6mVi1q3NvCT7C6GldWqVITUa2vWiUU2egznXbF7G8xkkXu32ajUZrTVK5trmBllqPR
9TqGlmlnhGrRUKMeuk99k+Oe4qorhZrdQpxT5naJS0l/UWP1JP1WB3T/7Otvrfl9Zf9/UoqnKOphqH7
Nk2v8Av/ews6Si4zhZHM04stv0+Oj1Bc4LP69v8gcx1vSfyqj/AN/c5T/VLvy/8EujotwnbL2pyl0/1S78v/AF
d22vXSlqYOUH7L7C2iqmWoVunsSSXGCQnqZ03Sjql5rfstlqq2266M9PBxgvaeMICN1tc9dLr2+rhwS+ZG
62iNtdmn9Vp8VjHAutT0urdzhurnz4cia1cLLlwpq35fFtYwgNgAAAAAAAAAAAAAAABk/zbPJfwazJ/m2e
S/gJvi0AGmQAAAAAAAAAAAAAAAAAAAAAAACO6PiX1IJ4m

AGrf/AMF+xCvTbdM6Zz3xfLhjBUtDNLY9RLq/CkBDo/3a7zf7EaE30XbjvZq0+m6iqcN+7d245EtNR1FTrct2XnlgDJpar56eLr1G2PdkjX6bTSqvnZOxTbWHwl+hShJui6VafYW6fTKjc3JznLnJgXGLTf1C82INWn6vUWW7s7+zHICiWI9Kxcu2PD6F+taWlsz3DUaaOoSy3GS5SXYU+hSm1118pxXYBVZldEwz/wB4k9b/AE+v/wBf2NOooV1PVp7Vw7CN2m63Txq342444Az6/wDAob9INZ+hvysZzwITqjZT1c+Kxgzeg2bdi1Eur7sAR6OSl1/gb4fczqUIVLSf3OzH6Hp01RprUILgvyVeix9L6/d/647QM9rWj1MnHhGdfDzX/fuTqrcOjY5yi5f9/Qu1WmWpjFbtri+eMlyilFRXJLAHn6Sq6dCdd+2PH1cc16nTTrunZK1Tk44fAPROE26LpVp80WafTKlyk5Oc5c5MCrotrqJrtUuJy3+q1f/wCSdmifWuym11uXNIV6LZfG12uTXPK5gajEv6q/y/wbSlaf/wAt37uaxtwBVZTC6+U6LtIq9rByF91WojVftlu5SRO3Sbretqsdc3za7RVo9tqttSDKlyz2AaTFZ/Va/wAv8M2ma/SStuVsxBpY4IDSYbH1XSkZs4Kaxkn6Jf/AP8AXP6f/Zbbpo3VRhY23FcJdoF3LmUU6haiFjFpR4ZfaVehWtbZamTh3YNNdMaqurgsIDz9FpK76nKe7KljgyekjHT6udU0tz9mRq0un9HrcN27LznGDmp03XyhJT2TjyeMgXmK/hOnS3y24/c2rlxMnSEJbYXQ9qt5/QC3We62eRzQ+51/r+5KSjqdM1GWFNcyiOithFRjqpJlsS/+wLtYm9LzjuMulqvnp4uvUbY92ORqoonW31lzstWMNFxOuoSboulWn2AS02mlVfOydim2sPgV6b+oXI+n0yo3Nyc5y5yYq0/V6iy3dnf2Y5AUdl+1R+Y0X007biyUMeF8zmq061EEt21p5TJUQnXXtnY5vPNgU2aZw0tsenY2s8fkS0E1PSxS5x4M0GSWixNzotdTfNLkBbqNSqHBBxXUnwSMuqj9lwUZ7G48Jd3Mvp0ahZ1lk3Ncm+ws1GnhqlpSypLk12AZ7NLfODjZqE49uUR1kHCGmg3na8Z+hP0KyeFbqJSguzvLb9MrerSltUH3ZAlq/dbPy spp/pb/ACS/k02w6yqUM43LGSFVKr06qb3LDTflCrQzeiWOfEzaKu2dcuru2YfFYNWm0s6J8LW4eHByzR/7rspscdnzx2gcr0ti1MbbLIKSLHYR0nv2o8/5LaNL1dnWWWOfyfl7CVWn6u+yzdnf2Y5Acu0zts3K6cPkmu6yt1aHa5OXrc2bSrU09fv3beOc4yBl9AhLTqUHLe4prjwyWdHTi6Nqioyi/W+ZqhHZCMc5wkimOm2ap3QnhS5xxzAp0XvWp/N/LHReFXZF+0pcS+nT9VbbPdnRhnGORnITXZqjyov6uafrJAd1H9So8v+Qv6q/y/wUwrS6Qgo2OxrjKT7zatP/5bv3c1jbqCjpT8KH5jvSnu0fzr9mXarT+kQjhdtw88sktRSr6nBvHan3ATj7K8jHDj0rPHZHj9EX0wl6pdBBvS7X2lq0MXOVmoksb3w8gNYAAAAAAAAAAAAAAAABk/zbPJfwazJ/m2eS/gJvi0AGmQAAAAAAAAAAAAAAAABxfiw/UuKV+LD9S4zreeAAIoAAAABAAAAAAAAAAAAAAAABTL8aXki4pl+NLyRcTfHLPw5eTJaT3aH6/uRs/DI5MlpPdofr+5dZxcACNAAAAAAAABxfiw/UuKV+LD9S4zreeAAIoAAAABAAAAAAAAAAAAAAAABGclzi4yWUzJ6+ifHM6H9Ym040msPiiDkJRnFSi8p9plxyrnPZOylbq37UO7yNFVsLoKUHlfsBYAYtVr41NwrxKXa+xAbCLtrXOyC/U8S2+215nNv5dhWKsfQRnGXsyT8mSPnU2n1No1Ua+2p4m98fnzFI9gFdN0LoboPK/YsCAAKAAAAAAAAdi/Fh+pcUr8WH6lz5PsM63nimWrojPa7Fn1i7nyPLVrr01ISr3wy11mOB6GnSjRBRIuSS495FWNpLL5I5CcZxUovKfJmfWzfVxqj7Vr2/p2k6ZxVkqlxx1SXE4pnq6IT2SsSfb8i58jz6LK69PKm6L6zjlbcuQHoJ5WVyl2WQqjunJRxeeno9uWirb8vuXzUWszSwuPHsAhVqKrm1XNNrmiwx0/8Akazr4LFcI7U/EbAKbNXRVNwnYIJc1hlkjxsgpweYvkzl3ZRqr7HRKcj7cNY7EaqblXVqcH6rAmQhbCyUowkpOPPHYdtgrK3BtpPuZl0cl16rUwgsRW3C/QDYRsshVhdZIRxzJHn75W6uyXuu3Y9sVnCQG6uyNsFKDzF9plq090boNqlI4vDi+xIk474OLbWVzQEY2wlNwjJOueaXYLba6Y7rJKKMumrjVrriQWEollJb+kkpcVCGUBoqurujmuSkkT5GTCh0lhbw3wef0NYFC1mnc9qsWeReYdbYnB09U4pvG9rgjbBYhFZzhcwOtpLLeEiNDkLY7oSUlyiN1MbklPOE84T5IPRqxpml43+4Giy2FUd1kIFfMknlZri6Qog6p2vLlwSy+CNkPYj5AcssjVBym8RXaUrXaZvHWr6M0GSKWp1UspdXU8Jd7A1kbLIVx3TkoprJGTX0QnTOyWW4x4LPBAak1KKafBIL1mnU9vWrPL5fUso/Ar/KjNq3Fw9GqgnOfYI7K7wNhTL8aXki2EdkIxznCwVS/GI5IuJvjln4cvJktJ7tD9f3I2fhy8mS0nu0P1/cus4uABGgAAAAAAAAAAAAAAAAMn+bZ5

L+DWZP82zyX8BN8WgA0yAAAAAAAAAAAAAAAAB4+u1Tus2xfqRf1PQ11nVaaTTw
3wR4pNXAAv0UqI3Z1CzHHDuyRpQD2aNv9VZ1KowmuGUjzNbSqNVOuPsrigKTVoZxd8Y25ePZy+CZl
CbTTXNAfRAros62mE+9FhWAAFAAAAAAAAAADLbp5Qn12n4S/uj2SNRzkQefqdfmnboBO
M3wlNsPOJ32dbdOfeyBGsASqip2xjKW1N4b7j0/RNBGXVSszPi7QV5QNOu0notiSeYS5NmYC3TXy09ql
HI2rvPbhNTgpR5NZPnz0+i7N1cq2/ZeUXE1vABWQAAAAAAAABxfiW/UssjvrlHOMrBWvxYfqXGdb
zxireoqo6n0fc0sKSksM0aWp00RhJ5a5lolqm/TQunGcpSi4rC2vBTRpJV6yybc9uFtblz8zYABkss1MIKCo
W58FPdwwwBCipU0xrXhainXwtsrjCuDknL1knjKNIopstclCWmdcEue5PBeABnnndfGUox07l4XuWGS
0ILooUZNOTbbwXADk24wbjHc1yWeZj0/Xx1Vk5adqNjXHcuGEbQAMm27TX2Srr6yFjzweGmawBRpKp
wU52YU7JzaXYXSbUW0svuOgDFX6QtXK16dpSSxtLgWaiuyN8b6072liUc4ySAM1MLLNQ77YbMLEY5y
aXnDxzAAx2z1F1Tq9H2uXBychlGquOyuMc5wsEgBC2c4RzCt2PuTwZ9Crq4uFlList7tyZrAGXXK6yt1V0u
Sf925luolOVa31uDXDGclgAGfRVzrqkrFiTk2aAAM+s62VUq66nPcue5LBoAFOndnUYnU4SisJNp5M2n9I
pUm9K5Tk8ylvXE3gDkW3FnRd7V3FUVxpeSLimX40vJFxN8cs/Dl5MlpPdofr+5Gz8OXkyWk92h+v7l1nF
wAI0AA
AAAAAAAAAAAAAAAAABk/wA2zyX8Gsyf5tnkv4Cb4tAbpkAAAAAAAAAAAAAA
AAAYOlh/ALUF3s8w9XpsDenUl/bl8ozrWB2EJWTUILMnwSOHodDRj11k5YzFYWfmFaK4VdGUb7Hutkvr
8keTbZK62Vk+cnk9fU6COotdk9R5LHJHlamuNVzhCanFf3ICsAAev0a86RfjtGsz6CDhpIZ7eJoKxoACgAA
AAAAAAAAAAAAAAABxe8UWPuiwjZhdXKPemiD58BrDaYI2Hqabo2uSrsIbuyllajzqapXWxrjzkzV/pu
qhatqXB8JqQE+mZyd0IOLUYrg+8889XpmUVXVBvM85/Q8oAbOi3jUSxfExm7oqGbZz7EsBNeoADTIAA
AAAAAAAAAAAA4vxYfqXFK/Fh+pcZ1vPAAEUAAAAAAAAAAAAAA
AAAPl+NLyRcUy/GI5luJvjln4cvJktJ7tD9f3I2fhy8mS0nu0P1/cus4uABGgAAAAAAbs5vAAAAAAATT5P
IAAAAAAAAAAbS5vAbS5tIAAAAAAAAAAAAAAA2lzeAAAAAAAAAAAAAA
BxTjLIJPyZ0AAAAAAAAAAAAAAABtLm8AAAAAAAAAAAAAA
AAAAAAAG8LiOYAAAAAAAAAMn+bZ5L+DWZP82zyX8BN8WgA0yAAAAAAAAAAAAAA
AAAAAAAIAXvq2qUH/AHI8GcHCbjJYaeGfQHm6qHplz6iOXFetLsZXGACUXGTjJNNdjBGgAACdFTutj
CPbz+RGUErJqME22exo9KtPDjxm+bCbrRFKMVFcksHQDTIAAAAAAAAAAAAdxukKHVe
5JerPijMe7fTG+pwn+j7xJw2SazuSeMrkzLWaim4tOLaa5NGldlapRx1r+iMwCuznKyTIOTIJ9rOAAD2dDS
6dOIL2pcWZOjtLGb62bTSfCPz+Z6hcZ3QAFQAAAAAAAAAAAcX4sP1Lilfiw/UuM63ngACKAAAAAA
AAAAAAAAAAAAAAUy/GI5luKZfjS8kXE3xyz8OXkyWk92h+v7kbPw5eT
JaT3aH6/uXWcXAAjQAAMstJNtv0mxZ44yZ9JVZqK3J6iyOHjmz0nyMXrf4E/zfwBHxVurRxi5ub3835M1
ekVQxGVkU+7JR0p7tH86/Zkno6VpmuZbc7u3IGizDql621OL9ZdnzlaVbaElZ1n/wAjPppN9GWZ7lyX2O
6ZSI0a1D2mngDQ9TTGW12xT8yxNNZTyjzNO9Kq+ruhtnybaNukq6mnarN8W8poCWo92t/l/wBjPorq6
9JDfNR4vm/maNR7tb+R/sZdDpqp0Kc4qTbfPsA2xnGcd0ZKS70Rstr9ucY+bMmnj1GvnTF+pJZSOTWm
qvnK+fWTb5YzgDbCyFizCSkvkxOyFazOSivmzz9POH+oLqU4wknlPyJatKOsu7ouVVMLHYBtrurs9ial5M
rsinq4Prtrs/D7ymuui2+FmnsUHHnFLmLv6pT+X/kDZOcYR3TkorvbI13VWPEJxk+5MyTj6T0i658YVrOO/
8A7ksv0ae2VG2ucXz7ANMpRgsyaS72QhfVOW2NkW+7JTrlVtwlfZtiv7e8yaqencluiLjJPmlgDT0p7tH86/
Zkekvdq/P+DvSLzpIN9sl+zOdJe7V+f8AbHZBTUHJbn2Z4nZzjCO6clFd7MfSEXCdV8ecXhnNdLrp00xftvd
+n/cgbk1JJp5T5MjKyEZKMPjsfJNkkkkkuSMGsgNbTCXJriBrjqKZS2xsi33ZE76oS2ysin3ZMu09Ven3wi
oyi1xRZTpKpaeO6OZSWXJ88sDUmpJNNNPtRCd9VbxOyKfdkx6K1w0Nr8DePoVaaemUHK/1rJPjlZA9O
MIKOYtNPtRGy6uvhOcYvubMWgnFamyutt1tZWQ1pabZu6fWzb7s4A3QnGxZhJSxyZx21pyTnFOPF8eR
h0koenSVWVCS5M6qY3dJWRnxSwcd/IDbC+qx4hOMn3JkpTjCO6UIFd7MGrqhRfROuO1uXHH6Euklvr
a5zi5VLmkBrhfvY8Qsi33ZJmDq9NqHHqJqqafcbwBjnrEtVCMZwdXF/U2Hn3U1rpGqChFRceKx5gS6Rk
pV1OLynLg0anqKYy2uyKfmZekopVVRisLdhE7tJTDsxBjGOD3aBrK5aimMsOyKfmU6bNnR6Tnt4Nbu5Z
KlvRQhsadj7ZJAb5WQjDdKSUe/PAzaXV9ZObcJRXHEfmQ6OSs084TW6KlwTOaCqErLW4puMuHyA3kY
WQm2oyTa54flkYqv9npKceyxZX/fqBr6yG/ZuW7njPETshBpSkk3yy+Z5m+XpHpX9nWbf0/8A8NNn+70I
CPZWsv8A79ANc5xrjmclFfMjXdXZ7E1LyZ591tc9dlr2+rhwS+ZG62iNtdmn9Vp8VjHAD1TBr7YznXUrEo
uWJ4fl3mDWVwWqoxFevL1vnxE4UaWtwujPCT4PdwNU7IQxvko55ZMfSMlw00lwiorfyXkx0ms1VL
5ga421zm4RmnJc0jk76oS2ysin3Zl16aumL6tYljG5mHTuivdDVQaszzksgel1kFDfvW3vzwM1GsU7rlznFR

TxD58Tum09ajPbNWVTfCPcU6OquWqvTgmoy4cOXFgegZLdYo6mEYzi4P2n3Gs8/UVVrXUxUEoy5rHM
Da7q1BTc4qL5PPM49RUm07I5XPjyMvSUVDTwjFJJS5luj06oVuKjxccOQF8ZKUVKLTT7UcjZCUnGMk2u
aT5GPRW9Xp7Yz51Z4E+joNUysl7U3kDTOcK1mclFcssaPV9dFqyUVPOEu80zrhYsTipLnhmLoyuEq5Tc
U5KXB9wG2y2FazOSj5sV2wsXqTUVJmHTVrVai221b1f4SZ3VVx0ttD1S2pvDS5AbYKsg5uCknJc1niSMW
p/2dbVd2S9WX/f8AvIDXOyFazOSivmVWRT1dcuu28PY7ynU/7+tqp/tj60hd/VKfy/8AIGudkK1mclFfNn
K7q7PYnGT+TMesW3VwndFypxjgdhXp7roT09ihKPFxS5gX2xT1Nb67bj+zvK9RrFC6uMJxcW8T+RHU/w
BRO8iOsqgtTQIBldL1uHPigN0JxnHdFprvRk6U92j+dfsXCEYR2xSS7kZOIPdo/nX7MC/0imOllysin3ZLeZm
joqep27eLXtdpTpbpQ0VvHjXwQGud9VcsTsin3ZJqcXDcpJx70zHo9LXKhWWRU5T48TIMVRrpULjXNZw
wJ+mL0tw3w6rHMazV9VBdVKLIoniufAqVNf+pOGyO3byxw5Hekq4RqjKMUm5YyvIDbCyFizCSkvkRIqKY
y2uyKfmU6vGn0kuqSjueOAo0dSojvgpSkstsDUmmssp5T7UV9fUo7usjjOOZm0TdWot0+cxjxRXoKK7XZK
cd2HhJgbo3Vzi5RnFpc3nkcfVOW2NkW+7Jhnp4f6gqliQkstJkukKa6q4WVxUZKWOAG6VklNbpJZ5Z7SP
pFO7b1sc+Zl6RW9UJ9r/4JanSUx0snGCTisp9oGmd1dbxOcYvubJRGcd0Wmu9GPR6euenU7I75S7Wc0
P+3bqIL2Yvh9wNdl1dXtzUfNnYWQsWYSUI8mYtFVHub7rlublhJiyC0utqlXwjZwaA2zshWskskor5s5XdXZ
7E4yfyZj1i26uE7ouVOMcDsK9PddCensUJR4uKXMCep1fV2Vxr1FrOJ/I1QnGcd0JKS70YddV роу Yiluk93z
4o3QhGEds1qK7kB0y6zVdTFKuUXPPFPsNRg6Trgg4zUUUpOXF/oBbrLIWaKxwkpLhy80Sovqr09SnOKe1c
GyOsrhXobFCKiuHLzRzTaSl6eLIhc5LLbA1RkpJOLT7URsurreJzJF9zZk0E3XRd2qDbRRp7NP609R61kn2r
IHqRnGcd0ZKS70dPO0c4LWyjU31clwTPRAAAAAAAAGT/ADbPJfwazJ/m2eS/gJvi0AGmQAAAAAAA
AAAAAAAAAAAAAAADjaSy3h15KShFyk8Jdpk9fWy7Y0L6yIEpz1knCtuNK9qXi8jXXXGuCjBYSEYqEV
GKwlyRICq7T1Xr148e9czFZ0W//wAdi/8AZhpAFt/AKbfnnDzyW19F/Es/SJ6IC1itoWI23UL2PaXejXCas
gpReU1k61YZkpf02oD/L/Dnxg+75BGwAFAAAAAAAAAAAAAON4WWG0llvCRjlKWtm4wbj
QucvEQdnOWsm66m40r2p9/wAkaFTWqlXsWzuJQhGEVGKwl2EgMFvRkJNuubj8nxRQ+jLlych+p6wBX
lw6Lm369kUvlxNIGjqp4pbpd7NABWW7TyjPrtP6s+2PZls0+ojcmvZmucXzRcZ9Rp+sanB7LVykgNAM9G
o3S6q1bLV2d/kaAAAKAAAAAAAAAAi5KNkHJ4XEn19fiOAKXNjvX1+IdfX4jgJF7O9fx4h19fiOAQ7O9fx
4h19fiOAQ7O9fx4h19fiOAQ7O9fx4h19fiOAQ7O9fx4h19fiOAQ7O9fx4h19fiOAQ7O9fx4h19fiOAQ7O
9fx4h19fiOAQ7O9fx4h19fiOAQ7O9fx4h19fiOAQ7O9fx4h19fiOAQ7O9fx4h19fiOAQ7O9fx4h19fiOAQ
7O9fx4h19fiOAQ7O9fx4h19fiOAQ7O9fx4h19fiOAQ7O9fx4h19fiOAQ7O9fx4h19fiOAQ7O9fx4h19fiOAQ
7O9fx4h19fiOAQ7O9fx4h19fiOAQ7O9fx4h19fiOAQ7O9fx4iCkpWycXIYRIFibtRs/DI5MlpPdofr+5Gz8O
XkyWk92h+v7jTFwAI0AAA+Ri6L/An+b+DaAMfSnu0fzr9maZ/gS/L/BMAYdL/AE23yl+xLS9Z/p66rG/jjP
mbABgesplVt1Fbc1wa2InR0Jwoe5NJvKTNeABXqPdrfyP9jFpNXGihRsjJLi4tLmeiOYGLSKV+qnqHFxjjEcl
VNsdJfaroS3N5UsHpDGQPNh7eka5uLimsRz2rDL7tRZRe+tjml8mlyNYA81uF+rremg47XmTxhFt39Up/
L/ybcY5ADFFu02s912uUJLEsdgnrJXSjDSP5zxk1wRtHIDBrc16qq2cXOtLD8yvWapX1ba4S2p5cmj0xyAw6
950NTXJtsx0I7tX5/wbgBXfX1tE4d64eZi6OjKdrsn/ZHajTqrLo4jTW5OS9ruJaSnqKFF+0+L8wLjFqP6jR5G
0AZukfdJeaLtP7vX+VfsTAGDQQ6zS2w8Ta+xGi+Omj1OoraceTxzPRDWeYGfTXddKTjVth2SfaZNNdHSO
cboS355pHpjAHnUWOpzHfKLjuXBPyLKf6pd+X/g2gDF0h+Jp/zf8FmputotjLbup7cLiaQB5monXqJRWnrf
WZ5pYNtl/V3V1uOXPtzyLsY5FL0+7VK6Us4WlruAuMOsl1WtqukntSxw/X/k3DnzAwa+asppnHk5cDXqf
drfyv8AYsAGCuEp9FbYcXx4d/E5RrlV0xrjVLrFwwlz6AxxYBh6Mfq2p893FEdLaqNTbXYmnOfA9AYQA
9JRcertjwazHJuAGR6f8A/jurx623d+vMj0anLrLzCw8Rz5G0AYLU9Lq3c4bq58+HlmtXCyyMKat+XxbWM
I2BLHIAytb73pvfyjaAMfSnu8fzfwnsXsVfmNoA5PdsIs9rHDPeYlq47XDV1vev8A48zcMZ5gYujovfbNRc
a5P1UyFVq0+tuVifry4fU9AYAGHWvqtVTa03Fc8G4cwMHSFkbdLXOPjy4fc3g5NuMJOK3NLKxeB5uur
DUtQ5XJL7npVwVclwXKKwZKYW36IXXQ2RgvBibAB5+gtVMpUTTU3LgegMccgYIyeh1Fm+LdU3INHLjv
XXQhXF9XF5IJnoDkAKNbV1umku1cUXgDF0dGU998+MpcExd/VKfy/8AJtAGW++yi/8A3I5oa5pcjNOUL
9TW9NW008yklhHphLHIDFqP6jR5DpDMLKlcNxjLj9jaAlVwrxU4Zw+8zdKe7R/Ov2ZsAGL/Uiwr2yhJ
WpY247Tul00vQ5xnwlZx49hsAGDT6paaHU3xlgueXDmS0ylfqnjRcYYjk2tZ5gDBdNafpFWTT2uPNE+
kc2aWM4ptZT/TBsazzAGWco67SSVecrv7yurXRrqULYyU4rGMczcljkMIDHoq5yssvsW1z5L5HOjPZt/Mb
QBin/Vofl/hnelPdo/nX7M2ADDrv8fz/AODTq/dbPystAFGg9zr/AF/cp0azqtSv/l/LNoA8+mx6GUqrYvY3
mMkjqb1urhOMWqq+OX2m/mAMt991f/8AuRzQ1zS5GacoX6mt6atpp5ijLCPTCWQGLpFOLpsxlQlx+
xbPWQWnV0YUzjHI0FOo0/X7E5YhF5axzAui8xT70ZOk4uWnTSziWWawBi1F8b9BZKGEGE8+aNGI92q
/Ki1LHIAydbHdHURfJvbCi1aTNV8OTypY5nojnzaae/rrHsqxWlwk+80gAAAAAAAADJ/m2eS/g1mT/N
s8I/ATfFoANMgAAAAAAAAAAAAAAABCyyNcHKbwkcuthTBym/8A7M9du9TNW3rEV7MP

+SBGE9ZJTstJsvZj3+ZrSSWEsJA6AABQAAAAACnU09dU0uElxi+5lwIKNLd1tfrcJx4SXzLzJqE9PctRH2Xw
mv5NSakk08pgdABQAAAAAAAAAAAAONqKbbwkclJQi5SeEu0yetrZ5eY0J/8A7EBuWtlhZjQub8R
rjFQioxWEuSOxiopKKwl2HQAAKAAAAAAAkr6l3xw+Elykuakqr5VTvWo4P8Atn2M1ELaoWwcZrKIJgx
xsnpJKFrcqn7M+7zNaaaynlMDoAKAAAAAAAAAAAAAAzzy1EnY4U173Hm84SFI1k
NNKyUFGSfLOSDQDieUmMrVA6DhXqLXVS5xw2gLQcTyk+8AdAK5SmrYqMU4Pm88gLAcyl2go6Dmc8j
oAAAAAAAAAAAAABGz8OXkyWk92h+v7kbPw5eTJaT3aH6/uTVxcACNAAAAAAAAAAAAAAAZP82z
yX8Gsyf5tnkv4Cb4tAbpkAAAAAAAAADLq9bGj1Y+tPu7iDS5KKy2kvmUS1uni+Ni/TieRbdZc8zk3
8iaQx7S12nl/8Ak+qLojjZi018j58IXbOqWYSaYpHOBVffGiOZcW+SXNmSrPHNeJRzZySXaX0ad7uve6x8
I2RCRymivk1dqOf9sexGoAACgAAAAAAAAAclJRI5SeEu0DkoqcXGSynzMINy00pU2ywo8Yt9xRqukJsbj
Twj39rMLbbby3lkXMey9fp0/b+xOvU02PEbFnUPDAq9CdPF0+sspaWd0e5nrU3Qvhug/NdwZ3FgAKAA
AAjOcYRcpPCR5ep187G41erHv7WQzHpWX11e3NIq9P0+fb+zPGbbeXxArUe7XqabHiNib7ic5xhFyk8Jd
p8+WwvlmKtbnXF+zkVI9BRlrZ7p5jSuS8RsSSSWiFNsLa1Kt8O7uLAGACgAAAAAAAAAAjKMZxcZL
KfYZGrNFLMczfZ2xNpxrKwyDlc42QUoPKZlxzpnppuyjjH+6BfVqK7K96lhLnnAtK7Lq61mc0v1PP1XSeP
NxP4Lxd5hbcnlt/MVY9I6/Tr+/7E4aqmx4jYsnhgVY+hOni6fWWUtLO6Pcz1aL4Xw3Qfmu4M7i0AFAAA
AAAAAAAADBr9VbTbGFfBYznHM3k1wsxvipY70QxHT2O2iE5LDa4lj5BJJYXBHQMu/Afi3Pd5kt7rP8A
T9xLTNWODVjrcuaxIMI1DITKuyxz3duOQENXKUaYKLxuajnuIX6eFNTsqzGceOc8y1adyqdds967HjGCPo
05Yjbc5wXzjGQIP/yNRGE87FBS297O6qqFWkmq44WV/wLbaN8ozhJwnHhiHJUTnTKFlu5t88cgK9M3dY
5WPEocFDulqx0xvhniP6l86FKcZpuMo9q7Uct00bLo2N42813gZ1ZovT2VSk3NPcb+ZY47NRp455RZ
ZPTRnql2t8uzvJzq3XQszjb2AZLYKF5aiuU4N8JlsJaiacqa4qUq2s4jzZbOi2W5de9r7Np2emi64RjJcpZkF
U0px1C6uqyEGvWUlwnpVVXZGWZ27/ljBaEAAUAAAAAAABGz8OXkyWk92h+v7kbPw5eTJaT3aH
6/uTVxcACNAAAoWofpj028Es5LwABTqtQtPXuay3wSAuBTTbZKMPxwVaXFPJbCUZxUotNPtQHQAA
AAAAAAQun1dUp4ztWcHNpB11MbGsZ7ALAAAAlwshYswkpJdwEgUVahz1NIW3Ch2I4AAAAAAAAAA
AAAAAAEVZB2OCkty5okAAAAHJy2wlLuWSvS3O+ITaxxxgCOAAAAAAWFqudFO9LPHGCyEt0ly71kDo
AAAAAAAAt1Dr1Fde3O/t7i8ACMrIRkoykIKXJd5Xq73p6lNR3ZeALgE8pMAAAAAAAAVX6iNDh
uTe544FoAAq1NropdiWcdgFol1S31QnjG5JkgAlwshPOySeOeOwkAAAAAAADknti33LIHQU6W531b3H
HHGC4ACMLIWLJKS5cCqnUOy+yvbjZ294F4AAAAAAAAAAAAAAAGfWX9RS2vafBHiyk5Sbby2a+krN+o29kUZDL
WAScnhLLYN/RNO+52y9mH7hWBpp4awwaNdY7tRoAT28k8GcAnh5R7Gh1HX1Y17ceZ45o0FnV6qPdL
gwmvaABpkAAAAAAAAA8vpHUuU+qi/VXP5s9G6fV1S13LJ4DbbbfNk1cAAuAl0n1Nnw5fQjKucPai15
o9vVamemoqclbspL7HmavWz1UYxIFR2vPADMW6a+VFqkuXaioAfQxkpRUIxTOmPoyzfp3F84s2FYACr
U2dXp5y+QHm9Ial22OuL9SP3ZkAI2HYxl4im/lnpqXffGtdvM9a26OixVaelyaXFgeNKLi8STT+Zwt1V8tR
c5yjteMYKg9JqHRann1XzR7SaayuTPnj2OjrN+ISfOPAuM61AAqAAAAAAAAAAAAAB4/SEoLUNV8
PFjtPVtn1dUpPsWTwJNy2+bJq4BLLwgdrIsnGS7HkjSXU2/Dl9DjqsisuEkvl9OjpK6+1VxqjI/MI0pqICt0Lj
KS4/IDyC3TXyotUly7UVAD6CMIOKkuTJGLoyzdQ4v+1m0rAACgAAAAAAAAAAAAAA
AAAAAAAJZ+HLyZLSe7Q/X9yNn4cvJkt7tD9f3Jq4uABGgAAyI/VX+X+Cd+ps67qalpz7W
+SIL+qv8v8FUIWS11yhZ1cst8uaAvhqLqro16iMcT5SRTTr+t62vd27vU/+yyekunKls1CeHwyh0h7dH5v+
AO3+lejy39Xjd3Y7h0d1vVRzt6rDx35yX6r3a38rlDh+5w/X9wNBgg1Wpuco1wi2n7T5JG8xdG+zB+YCW
n1FvpDovilLGU0dv1NnXdTRFSn2t8kQn/VYfl/hjS8NfqFL2m8ryyB30m6iyMdTGO2X90ewt1eo6iC2rdO
XCKK+k2vR0nzcuBXqPV1Glc+XDPMaUnq1RJ2wi4yWHjmi3STVfr8Zy5JN/ct1futn5TG0/8ASFj/ALxAsjdq
7o9ZXCCCh2J82X6XUekVttYIF4aM9FOoLTBw1GltcFjkXaTTumVjdim5PjhdoFl3W7F1O3dhju7jF0b1uHt2
9Xu9bPM9ExdF/gz/N/AFSnZHX3KqCIKXDjyRdHUX1aiFd6jfJo5p/wCo3+Q13vGm/N/KAt1WpdUo11x3
WS5LuKpajU6dx8YuDeG49hXqlyfSUVGextcGWWaW+cNtmoTj80BPWamVDqcEmpZz8+RXZdrK4dbK
EFDtj3HNxFwnpl5aeM/Q06z3SzyAsqmrx4zXKSyCutjTU5y5Ls7yGi90r8irpNN6ZY7JIEY26yyHWRhBRf
FR7WXU6qNmndsIt2+0i2tp1RceWFgy6mVc9Hb1O3g1navmByN2ruTsqhBQ7E+0tp1LtonJrbZBPKKNP
VfOidhqrMRxwWOROvTyphfKVim5RecLtAjVqNVqlf7cILHOTLNpqlbOsrlFK2C4dw6N91XmyFH9Tu/L/w
BTH0j+eNnW4493JGjVamyiyuMYqW5cV3sjD+rWfl/hDV++6fz/kCyN19dU7NRFcMbVEr63WSr61Rgo

4yo9uDRqrnRQ5pZfJFCq1E6usnqXHKzhLkBZC5X6Oc8Ye1pr9CPRvuq82VaH3C3zf7lt6N91XmwL7puum
c1zim0ZKr9XfWnXCCS5t9pp1Xu1v5WQ0Hudf6/uwGj1Ert8bIpTg8PBXLU3XXSr00Y4jzkzmk981Pn/l6Lw
q7I/3KXECVWpsjeqdRFKT9lrkyWq1Mq5xqqipWS7+wr17zqNOl7W7+UOXs3rdseH0Aq1ktSqdt0YuLfC
UTVO9afSwm1I4SS7+BHpL3X/2RRrk+p07zhY5/ogLHbrlQ62Ulbebj2pGqm1XVRnHkzO9PqZRaepymu4
t0IXU0qG5S45TQE7VTU5tN45Jdpm63WdX1u2CjjO3twX6u90U7kstvCKXTfKlzs1LXq5aS4AWw1HWaS
V0VhqLePmiirUaq+Ga4QWOcn2nNL/AE23yl+xd0f7pHzf7gNJqJ2ucLlpThzwQnqbrbpV6aMcR5yZzTf1C
8dG8Otj/aUuIFU5Wy1tKuioyT5rkz0jFqv9ObQPn1PX+IVbtm7PqY8+0s1+/0OHWY37+OOXad1fvun8/5
O9Ke7R/Ov2YEXbq+r6yMlxglnD54NFOoVm62XqpLiTn+BL8v8GGpN9FWY7/+ALI36q9OdMlxh2bu0t0
2pd0JKS22Q4NFGmqlvnRF16jbHuxyLdPp5VTsnKtClx4doFVOp1V8Wq4Qynxk+RbptRZK6VN0Upx48C
PRfu8vzv8AZHI/1Wf5f4QG0wy1WolqLKqoRk0+HyNx0nv2o8/5AlfdTCnco75PEizV6nqlxUVunLkinpH
2qPzf8AA1XDX0OXsgJ3aymtzshBr5dg1Fjt6M3yxI45eZfrPdLPIyz/AKTH9P3A7VZq5URdUlqMYpLPN4La
r7dRpt1airE8PPlu0/u1X5F+xm6M/Ds/MBVoeu62e3bjct+f4PSMXR3t3/m/5NoGW7UWu/qalJyXOUuR
yGourvjVqFH1uUonJW3X6mVNU1XGHN4yyym6uVer06na7G5Ln2cQL9VqbKb4QhFSUly7WQsv1dCU7Y
Q2N8Uuw7qf6jR5FnSPukvNAT1GpjTQrMZ3eyjPKzWKpznCli1xS5pENVwo0rfspLP0Rvta6qbzw2sDP0b
7r/wCzLtR1uz/Z257d3cU9G+6/+zNUvZfkB5/RvW7fV29Xu4558iMJ2x1typgpSk+3ki7ov3eX5/4RzSe/ajz
/AJA7DUX16iNV6j63Jos1WpdUo11x3WS5LuKtZ75pvzfyivURk+koqM9ja4MCyWo1OncXfGLg3huPYT1
epnTOvYIJS5rvIWaW+cNtmoTj80R1M XC7SRby00s/qgO2XauqPWThDb2ruNlc1ZXGa5SWSrXe52fp+5L
Se61/IA5rLZU0OcMzyuZR1usnX1kYQuCz2sn0l7q/NF9X4EPyr9gMteqv1CxTCKaXrN8s/lnptRZK6VN0U
ppZyiPRf4EvzfwcX9Vf5f4AndqbHf1OninJc2+witTdTBGGpjHbLIKjzRcNXqE/azn7neIGuoiu1y4ATlfOGtjV
LGya4M7rL5U7I1pOc3hZK9fB9RC1e3W0yNULqtcrF7FcVjz/wC/sBbqdTKuUaq4qVsvoit6jUaeUfSlxJP
Y9hXdGcukmoz2Nrg/0LLdJfZHFmoTjntQG0o1Wo6iKUVunLgkXQW2EU3IpYMeq4dIUSI7OMfqAlbrKY9
ZZCDj2pc0W26jGjd1eOzGfMtvaVFjly2swRT/0mWe/h9UBvpm50wm+ckmyntaidsLXLHqPhgs0vutf5UZ
dD+HqPMDFWp1d8f9uEeHOR6Bk6M92f5mawAAAAAAAAAAAGT/Ns8l/BrMn+bZ5L+Am+LQAaZAAAA
AAAAAAAAAhazjqrM95UaekYbNVJ+LiZjLeB7dVE6ujtlSzZJcf1PENC1+pSSVrwvkB6c9NY+jVSI6+OWTx
Gmm0+aPVlr//AAE1auuuxx7zym222+bAE6eF0MeJEC7Rw6zUwXc8ge4ADTAAAAAAAAAM+veNJzg8
U93UQ6zTziubXA8ImtYB cwdht3rfnbnjivRh0u0oxdSwljmS6Sprnpo6iuO1vGfmd6vo6zE96j8s4KekNZC
2EaafYXaBgAAHodEvjYuw9lw9FwxTKfiZuKxvoZeks+iPHejUUayHWaaaXPGQY8QAEbb+h0vSZPt2I93Sc
qtS61WtqeH3nn6S96e+NnNcmvkeIN6C6xXymlLnjifPS9UV1dsVhy5nmmpvDvrU2JQ9iPL5mQAel0S/8
AbmuzJ5p63RkNum3P+55GJrYADTIAAAAAAAAAADPrnjSWY7jxT3tRDrKjx70eDyJrWAAlr10iYK
FNIz8kebbN2WSm+beT0NNqKq+jpVuaU3ngeaAAA HodE+1YuzCP SMHRUMVSh3s3IY30ABQAAAAAA
AAIzshDG6SWe8gkDieVIHSgDPpbJT63c87bGI5FtIsKo5m8EEwUw1Nc1Jp42rLTWDLVZxBOU7LZKSk8JP
hgEegDNp9VG1yi2s7ml5GkACFs1VXKb5JFEK77YqcrnDPFRS5AagVzsjTBOyXy8zIV8LW1F8V2NYAtBQ9
XSu1vjjgjl+pjCjfGSy16oGgFdV0LlbILlzlR1dUpKKk+PJtcALwVWX11ycZSw8ZJwmrlKuc4feBIAyLrbdBfX
Sgo8kkBrBlpsm+urnLOxe2iyqcYadSlZujj2n2gXAojq6pNLLtbwsrmVu5V6uzfj7dqwgNYKo6iuUJTuuEee
VylaXUq6CTa38eH6gXWfh8mS0nu0P1/cjZ+HLyZLSe7Q/X9xq4uABGgAAUrT/8Alu/dzWNuCOo0itmrl
yclF/cjQAMsNHJ2RnndKzbxSLNRP+vIW923Y88s5LgBycVOEovk1hmfTaadEn/uuUOyODSABTptP6OpL
du3PPLBcAKXp86uN+7ksbcEb9KrZqyMnCxf3I0ADLDrt2Kd1rsa5J8i3UUR1Fe2XDtTXYWgDG9DOUDtm
olKK5LBfVQq9OqZPcsNPhzLQBj9CnDKp1EoRfYX6fTxoi0m5N8W32loAGWrSSqu3Qtag3IxwagBTXp9m
onbuzv7Mchfp+usrnxsecY5lwAp1GmjqEstxkuUkU+hTm0rtRKcV2d5sAFF+m62VTUtzrecYznl/wWXV
9bVKGcbljJMAQpr6qqMM529pKcl2QcJLkfM6AMa0VkU4Q1ElW+zBoqohVV1cVmL557SwAY/Qpwb6
m+UlvsLatLGqqcVJuU1xky8AVaanqKtm7dxznGDkNP1M7t2dyxjHluAGa/SOy7ra7HXLGHwJWabrLarH
PjX8uZeAIXRurcJcmZY6GeNktRJ1+FG0AZ6NL1NM69+5S7cciemP6irZu3cc5xgtAEbYdZVKGcbljHT1dT
TGv07HbjBYAKadP1V1lm7O95xjkQt0ebXZVY65PnjtNIAz0aRV2dZZN2T732HdTpo34e5xnHIJF4AxSOEr
F/uXyk+zK4I0zojZQqp8UklksAGP0K1R2LUy2d2DTVXGmtQjyXeTAELqo31uEuT7e4yrQTa2z1EnBf2mOA
Z6dL1WnnVvuzxxyyiT09RSq927HbjBYAKatP1eost3Z39mORXbo91rtqsdcnzx2moAZl6HFslHbKUK8vK
5msACjVab9rU3CUeTI26WVunjVO3Mk8uWOfM0gDjjmDjnmsFenoVNLrb3JvuLQBj9CIBvqb5Qi+wuo
0ypjL1nKUucmXACnS0ej1uG7d15zjAWnxq3fu5rG3BcABTVp+rvss3Z39mORcAKdRp+vcHu27HnlnJHWR
pnCMbZbW36rNBGyqFsdtkVJAefqaXChuzUufhj3miFDt0EKm9raTzj9SUNDRGW7bnzZoAjXDZXGGc7UI

kr0un9Hjbt2555YLgBljo5Q1DsrtcYt5ccccUABlu0e+3rarHXJ88EfQHvjN3Sc08ttZybABTzP+s1Fdu7Gzsx
zJainr6XXu257cZLABVLTxnp1TPikkslC0M3HZLUSdfhNgAq01Ho9Wzdu45zjBaABlp0kqbcwtfV5ztwWVa
fq77LN2d/ZjkXACm7T9bdXZuxsecY5jUaaOoSy3GS5SRcAMfoU5tK7USnFdneW3abrLKpqW3q3nGOZe
AIX1ddTKvOM9uDtUqqjD0dqxkkAkTT19Wzdt45zjZCO2CjnOFg6AKdLp/R63Hduy85xgej/wDl9fu7
MbcFwAz36RW2KyE3XYu1Ea9H/uqy6x2SLPI1Adk4qcJrfJrBvpdOtNBx3bm3lvGC4AU6nTRvw23Gce
UkVehTm1118pxXYawA5FeooohqlbzdnJrsLABjeinPEbNRKUF2YNLqg6eqxiGMYJgDHHQzS2LUSVfckWaf
S9RCcVPO75cjQAKtNR6PVs3buOc4wWgAAAAAAAAAAAAMn+bZ5L+DWZP82zyX8BN8WgA0yAAAAAA
AAAAAAARIJQi5SeEgMvSNHWU717Uf2PJpv9fWy4ZjQv/7FOr0Dj69Kyu2JFxgAaaeGCNAAHpdFOpQd
r5vgirR6GVjU7ViPd3mmP8A4mp2/wD4rOXyZU3WwAFZAAAAAAAAAPF11HU3tperLij2iu+mN9bhL
9H3EM14ILT RprKJYksx7Gi jYAAKuDs moRWWxXXOyW2EW2eto9ltOt0uM39gm6vprVVUylsRMArlca
ysM6Cjw9XQ6Lmseq+KKT3dRRG+vbLn2PuPHv09IEsSXDsZlrNVAAKAEqqp2y2wi2wO01u22MI9p7tcFX
BRjySwY+j4RqlOuSxanx8jcVndAAVAAAAAAAAAAAAA8bX0dTc2l6suKPZK7qo3VuEu37EM14ILT R
pbkjcVmPYkjYAAB2uDs moxWWzsK5WS2wi2zfXpLdKlbDepL2ohK201qqqMF2FhVRfc6OY8+1dxavKA
BQAAAAAAAAPP6Q0tt1sZwWVjGO49AEFWmhKqjEJPLS4loAGTRySd2WI/uPtl6lv0qlxlFLjhvlkvelolJt1pt
vLjdRV1fV7Ft7gKYVWS1CsnODwmmo9o0SXvz4L22XV0V1NuEcN9pKElwTUVhN5Az6PaIYnhPrGWWa
dWS3Oc18kzroqdm9wW7VLQM99T9EICLcmI niSouhOmLULy4/IuKZaWmUtzgssCjUuXpNLjKKWHhy5ZL
IV2PURsnODwsYiXSqhKGyUU4rsOV011PMI4bAp0MV1djwsubyUpZ6Os4cpP9zdCEYJqKwm8nl1wjFxU
UovmgVRqOOiezHsrOCtwstoUesq2tLGDXRXNxZFLPMitLSpbIBZ5gVbE9et3FqBosrVkdirbXkzuyO/fj1sY
ySAqqoVcsqc5ebMyp67U3+vKLWOTNxGMIxk5Jycub7wMune2m2ppKcE8/P5lb4aXTOxsKS3G3q4bnL
astYbOTjGFLjGvcl/agVRqpwcqEmm965HYRT6QsbXFRWCuNSnZDZQ64xlbZsUlqbml6z4NgZ6kvTr1jg1
EaDatNFcN2WvuaFCKm5pes+bIkjWdYoLdzyBOz8OXkyWk92h+v7kbPw5eTJaT3aH6/uNXFwAI0AAAA
AA
AA
AAAAAAAAAAAAABk/zbPJfwazJ/m2eS/gJvi0AGmQAAAAAAAArttjTBym/8A7IJWTjXBym8JGWMZ6y
W6eY0rlHvFdU9TNW3LEF7MDXyAJKKSSwk dAKKltLVdxlHj3ozS6Lj/AG2P9UegCFedHotf3WP9EaadHT
U8qOX3s0AFCu+pXVOD/R9zLABn0lrnF1z/ABlcGaDJqouqa1EFxjwku9GmE1OCIF5TAkAcgAAAAAAA
AONKSw1IGWzo6mfGOYP5GsEHmvovjws4eRZDoytP15OX2NwBdQrqhVHElplmAAABQAAjKKmsSa
+ZIEGOzo6mfGOYv5FP+l8fxOHkekAVir6Nqj7cnl1V1wrjthFJEwBl1dbi431+1Dn80X12RtrU48mSMIf/AI
up6t/h2cY/J9wGwAFAAAAAAAAAAAA42kst4SDaim28JGNuWtniOY0rm/EQJSIrlZ7lcKU+Mu87Z
0dTJermL+RqhGMIqMVhIkCvOfRfHhZw8icOjK0/Xk5G4Au0V1QqjiEuiYAGa/Tvd1tD22Ls7GS0+oVvqy
W2xc4svKNRp1biUXtsXKSAvBmo1D3dVetti+jNIAFAAAAAAAAAAAAAAAA
AAAAAAAARs/DI5MlpPdofr+5Gz8OXkyWk92h+v7k1cXAAjQAAAlwshNtRkm1zw+Q6yG/ZuW7njPECQI
zshBpSkk3yy+YnZCtZnJRxzYEgQrurs9ial5M6rlObgpJyXNZ4gSBGdkK0nOSin3skABHrlb9m5bu7tE7IVpO
clHPeBIEZWQg0pSScuWe0j6RS5betjnzAsAKtRdGmuT3JSw9qfaBaCjSahXVR3Sj1nHKXmWWXV1+3OM
X3NgTByE42LMJKs+TOgA2lzeDHPWJaqEYzg62uL+pHpGSIXU4vKcuDQG4Fb1FMZbXZFPzLAAK5aimMt
srIp92SxNNZTymABGc4VrM5K5ZZm0er66LVkoqecJd4GsEHdVFtOyKcefHkdjZCUN8ZJx7wJArjqKZy2x
si33ZLAAK/Sad23rY58ywAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAA
tnkv4Cb4tABpkAAAFV2oroWZy493aYLoK5v8OKS72Qj1AeM9fqH/evoSh0ldH2tsv0FWPSvvjTHMuLfJd
5TVTK2fXajn/bHuKNJbXba53S/wBzsT5I9IJ4AAoAAAAAAAADjSaw+TMILemvdEvYlxg/4Nhm1yg6cyk
oyXGL+ZBpB5b6TnsSjBbscWyp6/UP8Au8kKseyDx4913p8WmvI10d1zeLfsff2BI2g4mmssp5R0oAAAA
AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA
VaiXVOPJ80+5loIM+kudkHGfCyHCSNBh1c46e+N0Wtz4Sj3ops6Tm/
w4qPhxCx6gPG9P1Gfb+xOHSV0X6yjJCkesDNRraruDe2XczSEAAUAAAblv11VPBPdLuRjn0ldJ+qoxRCP
WOSkoxcpPCR461+oTzv+wnrJXSiruMFzS4ZFWNnra2fbGhf/wBjZGKjFKKwkvae+q2KVbSx/b3FwQABQ
AAAAAAABVfRC6OJc+x9qKa7p0TVWo5f2z7zWQsrbBxmspkEjpiUp6OW2eZ0vLtblyUopxeUwOgjKUY
RcpNJLtZiu6SjF4qju+bA3g8eXF7/ALkvJHI9IxR+5PzQqx7IPp6TTeLY4+aN0JxnHdFprvQSJA
AAAAAAA
AAAAAAAAAAAAAAA
jZ+HLYZLSe7Q/X9yNn4cvJktJ7tD9f3Jq4uABGgAA
Yqv9npKceyxZX
/AH6mffL0j0r+zrNv6f8A+F/SUXHq7Y8GsxTen//AI7q8ett3frzA5Z/u9JQj2Vrl/79BqY0R1HWX2Z4ciYO

dGpy6y2XFvEc+RCyS0+vIzBByjJeq8AVzs9Kplp4uPrYfDBdqf9jXV3f2y4SKtRqOuupkotQjLg328jZrq+s00
u+PrICnUf7+vrq/thxf8A36G0xdGxct90+MpcMm0DFH+qy/L/AAOk/wAOv8xy+T0+vV0otwksZTrNSr1F
QjLan7TXaBd0lHdZRF8m2v2LL9HT6PLZBRIFZTK+km42UNLLbS7+Qu10Z1OFcZdZJYw1yAu0M3ZpYuT
y1wO6yuEqJyIFNxi8PuO6Sp06eMZe1zZPURcqLIri3F4Az6GMlaiRW7Vuw8vt5leipjepXXLfKT7SWguhKIU
NPck8kKLXonKq6L25zGSXMDsoLS66vq+ELODRvMEHLWauNii1VXyb7TeB591Na6RqgoRUXHiseZLpKK
VVUYrC3YQ1kuq1tV0k9qWOH6/8nNfNWU0zjycuAFI2kphpZ4jxjHO7tIRulDotST9b2U/1NWp92t/K/2M
tNTu6M2Lm84+oFVmthGIKxbptcXh1Rk8qytPMYvMSNorrqrVd1bU4LHLmadLY7Yyk6tizw+aAsnXCxY
nFSXPDMXRlcJVym4pyUuD7jeefoLVTKVE01Ny4AchTC7pC1T4pccd53XuNfVUL1K+cseZPT/ANRv8iWu
qm3XdCO5wfFAZ7560VLVSxNcnglql5Po+rjxnwb8ix66px9Spub5RwS1dU7tLBqG2ceO1AUSloupcFndj
hLHHJp6Pm56VZedrwVLXvbPWre/tjtL430Olds69rXHagLwQpsVtUZpNJ9jJgAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAayf5tnkv4NZk/wA2zyX8BN8WgA0yGfWalaevhx+SL3wR4equd98pdnJ
eRDMVznKyTIJ5bOA7CErJqEVlt4RG3AelHoqKSVlyU32GPVaWels2y4p8n3gUnaDWPKqsf5WeeE8PKA
+iBRpLuuojJ8+TLysAAKAAAAAAAkr7o0Vucv0XeeLdd06e6b8l3GjpG7rL9i9mHD9TIZazABLLwuZ6NR
XqKV1ig32BXnA06zRT0rTb3QfJmYDXotY6ZKE3mt/Y9ZcVlHzx63RtzspcHzhw/QuM7jYACoAAAAAAA
AAAAAAAAAAAAAGbW6paeGF7b5flONpj8keFqLXddKb7eRFzEJSIOTIJ5bOCMXKSUU232I3PoycdN
K2yW1pZ2kaYQEsvC5no1dFZgnbZtb7APOPS0GscmqrXx/tZl1mjnpZLL3RfJmdNpprg0DX0QKtNb11EZ9
r5+ZaVgPO1+scW6q3x/uZr1VvU0Sn29nmeG2223zY1cwBOqmd01GuLbNGr0D0tMZynlt4aI0yAJnvCW
Wbq+jLHS7LjbWIIIDFCcoSUovDR7Gj1K1EMP21zR4xZprXTdGa/UJuPeBxPKTXJnTTIAAAAAAAAONK
SaaymY5Rnom5w9antj3G087pS7GK128WQxl1Wqlqj90FyRQC7TaWzUyxBYj2yfJEbUg0a3S+i2Rju3ZWR
pNJPVSe1qMVzbAzi2m1M9PPK4xfNGyXRKcH1VqlJdh5sk4ycWsNPDA+grnGyCnF5TJHmdF3Yk6XyfFHp
IY0ABQAAAAAAAAAAAAAAAARs/DI5MlpPdofr+5Gz8OXkyWk92h+v7k1cXAjQAAA
AADmAAMurlfJumqrhJY3moAQprVVUYLsRMAAAAMusrnZbQ4RbUZcflyNWOQAAAAYQ58wAAAAC+
YAAAABhMAABjjkAAAAGFnIAAYWcgAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA
82zyX8Gsyf5tnkv4Cb4tABpITq5OGmsa54PDPb1qzpLPLJ4hNawLtJZ1Wprm1IJ9hSbOjYxerW7mlw8yK3
6rQvUXxtjY4rlIGXpixOcK0n6vHLO6yesjq3s37V7O1cC3pNbtFCViSs4AeQAAPR6Jk8Tj2cz0TzeiVxsf6HpF
Y30ABQAAAADknIlfcpGazCS70QeBNuU232s4HzY12t0ritTW5ctyPQ6VpushKtOUcY4djPKSbaSWWz
0Vbr9Oo17XLK4cMgX6uEl0Slb7cUsnjnsdlzlDo+MLHmyWMnjgDX0ZJrU47JRzkNXRqzq0+5NhNewADT
AAAAAAAAAAAAAAAACjWycdJY1zg8Q9nXrOkn8jxia1i7Sah6a7eoqXDDR6kb5ajoy2yeMtPk
eNGMpyUYptvsR6+mrmuiPwcWpNPhgivP6Pgp62pPlnJf0vZJ6pRy8RiinRt0a6vrE48cPPzNPSunslqFZCL
kpLHBdoFuofXdDxnLjJPP64PIPX1S9H6jVL2mksfc8gD0+iZN12R7E0zeed0SvVtfzSPRKxvrz+lpNQrj2Ntn
mno9LLhW/NHnEazxp0mtlpoTjGKe7k+429KNy0FUUnzbTf0PLhXOx+pFyx3I9bpCuc9BTGMW2sZSxyCvL
01z090bEk8dh6+n1EtTpbZySXNJHiqMnLak92cYPX6PrnDRWxIFpvPBoDxwdnCvbxOLi/mcA9vRyctLW3
zwXmfQrGkh80aCsAKAAAAAAAAB4etk5aqzPY8HuHhatY1Vn5mTVxUeho9dOKqoUUvWSb+R55fo6rJ
aiqag3FSXHBGmnpr3iH5f5MMLJ152SaysPB6HTFc5XRIGLcVDi0uRDorTRtsdk1mMOS+Ygj02mWnrnf
9qa5PuPLvmrb5zSwpSbNuvnqNRNsjXNVp8FjmeFKljJqSw1zQFulk4amtrxJHung6dbtRWv/kj3i4zoACoA
AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA
AAAAAAAAADJ/m2eS/g1mT/Ns8l/ATfFoANMuSipRcXyaweBb12ShLmmfQGHpDS9YutgvWS4rvIua8s7C
coSUovDXJnARpvj0tco4cISfeZdRqbNTLNj5ckuSKgAANWi0rvnuksQX3A3dH1dXp02uMuJqOLgjpWAAF
AAAAAAAHi2rqtTJY4Piig9nWaZaivh7ceR40ouMmpLDRlIrNE3Fpp4a5G6HS10Y4cYfezCAqzUaizUT3
WPPcu4rAAHpdFVYjK1rnwRi02nlqLFFcF2vuPbhBVwUlrcSwi4m6kACsgAAAAAAAAAAAAAAA
AAI2RU65RfasHgTg4TcZLDTwfQmDpDSuf+7Besua7yaua8+i2VFsblpNx7zz/q93hgYARpZqL5ai3rJJ/JI01
dKXVwUWozx2sxAC3UamzUz3WPls51qBr0l0d01Oa9SP3A3aGrqtNFNYcuLNIBWGbx1dbpnhZceKG
PoTyNdpXTPfFeo/sNXNQ0urnpXJwSe7vNH+r3eGBgBGk4XSjerkluUtxs/1e7wwMAAt1OplqZqc0k0scCF
cHZOMVzbInp9H6VwXWzXrPku4JrbCKhCMV2LBIArIAcGAAAAAAAeV0pU42qxLhJcfM9UrvpjfU4S7eT

AAAAAAAAAAAAAAAAAAAAAAAAAf/9k="></figure><p>Constraints :</p><p>0<size <100</p><p>0<data<1000</p><p>Input format:</p><p>First line indicates the size of the queue</p><p>Second line indicates the elements of the queue.</p><p>Output Format:</p><p>single line indicates the in-order traversal of a tree.</p>

answer

```
#include <stdio.h>
#include <stdlib.h>

struct node {
    int data;
    struct node *left,*right;
};

void solve(){}
struct node *root = NULL;

void insert(int data) {
    struct node *tempNode = (struct node*) malloc(sizeof(struct node));
    struct node *current;
    struct node *parent;

    tempNode->data = data;
    tempNode->left = NULL;
    tempNode->right = NULL;

    //if tree is empty
    if(root == NULL) {
        root = tempNode;
    } else {
        current = root;
        parent = NULL;
```

```
while(1) {
    parent = current;

    //go to left of the tree
    if(data < parent->data) {
        current = current->left;

        //insert to the left
        if(current == NULL) {
            parent->left = tempNode;
            return;
        }
    } //go to right of the tree
    else {
        current = current->right;

        //insert to the right
        if(current == NULL) {
            parent->right = tempNode;
            return;
        }
    }
}
```

```
void inorder(struct node* root) {
    if(root != NULL) {
        inorder(root->left);
        printf("%d ",root->data);
```

```

inorder(root->right);

}

}

int main() {
    solve();
    int n,i;
    scanf("%d",&n);
    int array[n];
    for(i=0;i<n;i++)
        scanf("%d",&array[i]);

    for(i = 0; i < n; i++)
        insert(array[i]);
    inorder(root);
    return 0;
    printf("temp->left=temp->right=NULL; struct node* newNode(int item)");
    return 0;
}

```

question

<p>Question description</p><p>There are n hotels on a street. For each hotel you know the number of free rooms. Your task is to assign hotel rooms for groups of tourists. All members of a group want to stay in the same hotel.

The groups will come to you one after another, and you know for each group the number of rooms it requires. You always assign a group to the first hotel having enough rooms. After this, the number of free rooms in the hotel decreases.</p><p>Constraints</p>1≤n,m≤2·10^51≤hi≤10^91≤ri≤10^9<p>
Input

The first input line contains two integers n and m: the number of hotels and the number of groups. The hotels are numbered 1,2,...,n.
The next line contains n integers h1,h2,...,hn: the number of free rooms in each hotel.
The last line contains m integers r1,r2,...,rm: the number of rooms each group requires.

Output

Print the assigned hotel for each group. If a group cannot be assigned a hotel, print 0 instead.

 </p>

answer

```

#include <stdio.h>

#define N 200000
#define N_ (1 << 18) /* pow2(ceil(log2(N))) */

int tr[N_ * 2], hh[N];

void build(int k, int l, int r) {
    if (r - l == 1)
        tr[k] = hh[l];
    else {
        int m = (l + r) / 2, k1 = k * 2 + 1, k2 = k * 2 + 2;

        build(k1, l, m);
        build(k2, m, r);
        tr[k] = tr[k1] > tr[k2] ? tr[k1] : tr[k2];
    }
}

int update(int k, int l, int r, int x) {
    int m, k1, k2, i;

    if (r - l == 1) {
        tr[k] -= x;
        return l;
    }

    m = (l + r) / 2, k1 = k * 2 + 1, k2 = k * 2 + 2;
    i = tr[k1] >= x ? update(k1, l, m, x) : update(k2, m, r, x);

    tr[k] = tr[k1] > tr[k2] ? tr[k1] : tr[k2];
    return i;
}

```

```
}
```

```
int main() {
    int n, m, i;

    scanf("%d%d", &n, &m);
    for(i=0;i<n;i++)
        scanf("%d", &hh[i]);
    build(0, 0, n);
    while (m--) {
        int x, i;

        scanf("%d", &x);
        i = tr[0] >= x ? update(0, 0, n, x) + 1 : 0;
        printf("%d ", i);
    }
    printf("\n");
    return 0;
}
```

question

<p>Question description</p><p>You are given a tree consisting of n nodes, and m paths in the tree.
Your task is to calculate for each node the number of paths containing that node.

Input

The first input line contains integers n and m: the number of nodes and paths. The nodes are numbered 1,2,...,n.
Then there are n-1 lines describing the edges. Each line contains two integers a and b: there is an edge between nodes a and b.
Finally, there are m lines describing the paths. Each line contains two integers a and b: there is a path between nodes a and b.

Output

Print n integers: for each node 1,2,...,n, the number of paths containing that node.

Constraints</p>1≤n,m≤2·10^51≤a,b≤n

answer

```
#include <stdio.h>
```

```

#define N 200000
#define K 17 /* K = floor(log2(N)) */

struct L {
    struct L *next;
    int j;
} aa[N];

int dd[N], pp[N][K + 1], ll[N], rr[N], cc[N];

void link(int i, int j) {
    static struct L ll_[N * 2], *l = ll_;
    l->j = j;
    l->next = aa[i].next; aa[i].next = l++;
}

void dfs(int p, int i, int d) {
    struct L *l;
    int k;

    dd[i] = d;
    pp[i][0] = p;
    for (k = 1; 1 << k <= d; k++)
        pp[i][k] = pp[pp[i][k - 1]][k - 1];
    for (l = aa[i].next; l; l = l->next)
        if (l->j != p)
            dfs(i, l->j, d + 1);
}

```

```

int lca(int i, int j) {
    int k;

    if (dd[i] < dd[j])
        return lca(j, i);

    for (k = K; k >= 0; k--)
        if (1 << k <= dd[i] - dd[j])
            i = pp[i][k];

    if (i == j)
        return i;

    for (k = K; k >= 0; k--)
        if (1 << k <= dd[i] && pp[i][k] != pp[j][k])
            i = pp[i][k], j = pp[j][k];

    return pp[i][0];
}

```

```

int dfs2(int p, int i) {
    struct L *l;
    int c = cc[i];

    for (l = aa[i].next; l; l = l->next)
        if (l->j != p)
            c += dfs2(i, l->j);

    cc[i] = c += ll[i];
    return c - rr[i];
}

```

```

int main() {
    int n, m, h, i, j;

    scanf("%d%d", &n, &m);

```

```

for (h = 0; h < n - 1; h++) {
    scanf("%d%d", &i, &j), i--, j--;
    link(i, j), link(j, i);
}
dfs(-1, 0, 0);
while(m--) {
    int i, j, a;

    scanf("%d%d", &i, &j), i--, j--;
    a = lca(i, j);
    if (i == a) {
        ll[j]++;
        rr[i]++;
    } else if (j == a) {
        ll[i]++;
        rr[j]++;
    } else {
        ll[i]++;
        ll[j]++;
        rr[a]++;
        cc[a]--;
    }
}
dfs2(-1, 0);
for (i = 0; i < n; i++)
printf("%d ", cc[i]);
printf("\n");
return 0;
}

```

question

Z1iOPxdvjrfmLZbFW0abIdOTmblj15LZxVmsV46QlNImu7MawF5tU5ppOPhpr3I5NeypqtjBSJ10SvjrfTX
uEvNqvLWIwcuiE3VUrUi1d2eRWsVrERygak96jkmkab/0TjkjfHW+m93Jchf1RtFptaKRxQyb01j7kxu97
RGOSX3uOqUxExplz42qMtt/BEu7sfZ+Xcl2Nd3d46J7kbm73BI/VGONDnn4O7Pu7vdrqtrSKV0jk5GGkW
3o11DxvpXj/ALm6+0a1mPgjGOsXm0c5SFkxnwWikzW3CV3aU3orrxly2Kt51mOJXFSS6xHESSz0rn+6Rp
pGed//AGv7Ou/v8dS+Ot+ccQ8apvpOeNz/AE5TSM9t/wD2vpjrTIBfHW/OOleN+0YnHNp3dN5miLTvac
u9rpirTIBTHWkzp3hebUcFqzTSOEzWIVxVrbejXVMam57EMv+H6oTQy/4fqgVMAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAFe3NvkxN+WN7HaPgwyRy
+QAHEatm93PzZWnZvdz8xvj7UX9ufmilf25+alzQnkOxGsxHWQn2319mPk6RGkRAR1AAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAIR/cW/TH7poR/cW/TH7pgAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAIZf8P1Qmhl/w/VAJgAAAAAAAAAAAAAAA
AAAAAAAAAAAAAMOWm5kmO6eMNynNXteFecd4z1NjIExpOk84EecAEAAFuzxE
5dZ7ldaze0VjnLbXFwtN3/AGOvx8/qYhFppOluXdKauwAAAAAAAAAAAAAAA
AAAAAAAAAAACEf3Fv0x+6aNsdW1nXXlwnRzsafm9Ugmldjt83qk7Gn5vVIjiHY0/N6pOxp+b1SCYh2N
PzeqTsafm9Ugmldjt83qk7Gn5vVIjiHY0/N6pOxp+b1SCYh2NPzeqTsafm9Ugmldjt83qk7Gn5vVIjiHY0/
N6pOxp+b1SCYh2NPzeqTsafm9Ugmldjt83qk7Gn5vVIjiHY0/N6pOxp+b1SCYh2NPzeqTsafm9Ugmldjt
83qk7Gn5vVIjiHY0/N6pOxp+b1SCYh2NPzeqTsafm9Ugmldjt83qk7Gn5vVIjiHY0/N6pOxp+b1SCYh2NP
zeqTsafm9Ugmldjt83qk7Gn5vVIj0zf8P1QdjT83qkjdSJiePDrMgmAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAEzErKGk5OfCvTqBrN+FeFe+U4iKxpByAV5c
MZOPK3VkvjtT2o+reDPXMrzhtgx2/xj6OfZsfSfNHP+bGnTFe/KNI6y1w468qxqmrU+OfqGPHXHHDn1
TAadCYiY0IDjj+Nf8AiYBE6xrAhMTTjXjHfCVzi0awDoAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAABG1Zid6vCe+OqQDlbRb4THOHUbV14xwmO8rfWdLcLAKAAAAAAA
AAAAAAAAAAAAAAIwbtN2tYnSNeM6Ob2Xwv9X8AsFe9l8FfV/BvZfBX1fwCwV72Xwv9X8G9l8FfV/ALBXvZfBX1fwb2
Xwv9X8AsFe9l8FfV/BvZfBX1fwCwV72Xwv9X8G9l8FfV/ALBXvZfBX1fwb2Xwv9X8AsFe9l8FfV/BvZfBX1
fwCwV72Xwv9X8G9l8FfV/ALBXvZfBX1fwb2Xwv9X8AsFe9l8FfV/BvZfBX1fwCwV72Xwv9X8G9l8FfV/AL
BXvZfBX1fwb2Xwv9X8AsFe9l8FfV/BvZfBX1fwCwV72Xwv9X8G9l8FfV/ALBXvZfBX1fwb2Xwv9X8AsFe9
l8FfV/BvZfBX1fwCwV72Xwv9X8G9l8FfV/ALBXvZfBX1fwb2Xwv9X8AsFe9l8FfV/BvZfBX1fwCwV72Xwv
9X8G9l8FfV/ALBXvZfBX1fwb2Xwv9X8AsFe9l8FfV/BvZfBX1fwCwV72Xwv9X8O1vab7tqxHDxhOojgAA
AAAAAAAAAAAAAAActWLRx83QEltNZ3b/SeqZMRMaShrOPhPGvXoCYAAAAAAAAAAAAA
AAAAAAAAAAAAAAIR/cW/TH7poR/cW/TH7pgAAAAAAAAAAAAAAA
AAAAAAAAAAIf/eP0/umh/8AeP0/uCYAAAAAAAAAAAAAAA
AAAACN71pGtpMuSMddZ590MvrWvbW08RnrrxWX2m1vZ4Qqm1rc7TLgjje7ROuS9eVpQBPKtWPaY
nheNJ6r+bzluHNOOdLca/8V05731WyZ0jWUOOT4V/6RE30m3LuhMdSlil0gAAAAAAAAAAAAAAA
AAAAAENjx8Y416dE4mJjWOQyz8ml5rjnSO8S3Pa3LtEU4V4yz2zLc7afJARxvdpz5uxMxymYcbnatptF6
8/vQ048tckcOfRhdiZidYnSRvn5LPt6Arw5e0rpPtRzWK7fYAAAAAAAAAAAAAAACEf3F
v0x+6aEf3Fv0x+6YAKc171vEVnTUS3FwotbNSNbTCztaxSLTwjRJ1ExybRFd7uQyXi2GbVkW1YIYp/wDK
szPmdtTXTUNiYI2vWmm9PMVIFWXLNbbtY1kS3Fop1zV4zGsdHc17Vx1mOEyJ5LQrxrE/BTmvatois6ai2
5NXCi1s1I1tpMLqW36xPUJddEM1t3HMxzRpbs8O9aZmQ3i0URbNeNa6RCzFa06xeNjjvEnWpiE5qRO
mqU2iK72vAXY6IdtTTmlFqzXXWNA2OivHli+uukJWvWntSGz7SEa3rf2ZcnLSJmJnjAbEwiYtGsckctprjmY
5hv6kIYrTbHEzzTmYiINzNQN/QQjLS06RKoE+7XSJ4yGzNWims9nSbTfe+GqePJF668NegSpgoyXv2u7W
QtxeKJvlxzE30mHc1tYrMW3dRPJcOTaK11tKPbY9OYuxMU57a46zWZ4pVzUilicRPKbiwRveK03nMeS
Mle6J6C7PpMU4rTXLakzM9FwS6lf/eP0/umh/wDeP0/uKmAAAAAAAAAAAAAAA
AAAAAAACGa27itPeDLmvv5JnujkrBHmt26Auw2pWJm0cRJNU6DXTLGSd3RRmrFckxHlavOTYrAGGrZc
mtZpPOOS9hw23ctZ+jcr08XYADQAAAAAAAAAAAAACvPfcx8Oc8IYI21W1yadlUo4fJfeAA5
mju6medlrFde5de+5TWefQbnMs+2IAYSpaaWi0dzdWYtWjILz2vZba4tOkjt8d/FwCuoAAAAAAA

AAAAAAAAAAAAAAACef3Fv0x+6aEf3Fv0x+6YDPtE6ZKyoIXxRe0W100GepsU5Ms3jd3dNxctd3DWoI3Jjj
JHHhMd7s0iabtp1E8b7QtMfZ/ohH9pP8A/d6X2ePFKfZR2W5r9RMqq0zGzV0c01x6Rj+uq+McduTxhX
9njxTp0C81PDvRjiLczJWlpjeTiNI0QyYoyTEzOmg1npNn13dp1s0l5MdckcQ6muzaljWZvbT7EfM+zx32
mU74ovWK66RAI2xKvsx8lG0TpkrKX2f88pWwxO797kF2zFWTLN43d3TVfjru0iHMmOMkceE9UqxpWI
mdRZLvtVtPsx83M0T2Nd05PPXex/Li7imL4o1+Qlm2wxWiccaGS0dnbdnjCE7PGvC0wnTFWkTHPUX39
Kcfsadnrr36u1i0YlxMc05KdnjXhaYhZ2cdnuajM5qrFjrOGZmNZcwRFq2rPjds5Td1cx44x68ddRZz9Kd
npW0zM908HNdc8613vg7CN7WLTdt8MXnXXSeonjcV1i3bRMVmsd7kVi20WieWq3Hiik66zMuxiiMk
315h41y82ppFK6wjeBwWTNo0lc5eu/WY10GrEMHuQ2mZ+7Hc79n0/wA5S7KNzdmdfjlsxVeJtWN3
HpPUz1+5WZj73KvbPET7UpZMfaREa6aCeNxXfHWMOsQ7grG5vacVs1ia7s8kKYdy2sWn5C+PvXlvk3
oiacNVeSd3aNdNWICcUTk39foFIU5LzlmK6a0540ikdFuTFF5iddJjl4t/TWeQl5vtVm95WJ9l3PwkuJTT
VLNNYIltEz8IFopppTWZE69ank9xRK2OsYNdOOic4t7HwszpoINNce5ql4qdN7zome5Lz6xu73esrjiuPc5
wjTDFLaxafkLnuVcf7qF6jH9/PNu6F4vlh/wDeP0/umh/94/T+40mAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAp2r3P1XK9ojXDPw4iX6YgEeUdrE2nSHF2HJSkcYnUWTb7WVrXBTWebPe03tMyvtlxW9
qJn6Kcs0mY3l0G+vr0gAObs1HzegwY43sly+Lerv8f0AdoAAAAAAAAAAAAAW5vfX+aC3a
Y0zT8Y1V18/f/QAMj47dnbWYaK2mrOsIVzU3Yi1S2esV0pA6zJ+qLRu2mOjjszrOrg5jTsf+f0ZmrZl0pM9
ZG/j+14Cu4AAAAAAAAAAAAAAACef3Fv0x+6aEf3Fv0x+6YAAAAAAAAAAAAAAAABE
RHKNAAAAAAAAAAAAAAAmInm5FYjIEogAAAAEReco0AAQ/+8fp/dND/AO8fp/cEwAAA
AAAAAAAAAAAAAAACY1jQAYL13LzWe5Fr2mkWrrHtd3xZEefvnKADAAACWok
5LbsfWegsmrtlprM3n5Q0uVrFaxEcov6ZMmAoAAAAAAAAAAAAACnaqa0i0dzI9GY1js
WLNj7O3D2Ry+Tn9Vgl4gAAAERMzpHOW/HXcpFeijDhtpF+U90S0VtFvhMc4V60OcjoA2AAAAAAA
AAAAAAAAAAAAAAAhH9xb9MfumjbHW06zHH5udjTpPnljihY06T5ydjTpPnljihY06T5ydjTpPnljihY0
6T5ydjTpPnljihY06T5ydjTpPnljihY06T5ydjTpPnljihY06T5ydjTpPnljihY06T5ydjTpPnljihY0
6T5ydjTpPnljihY06T5ydjTpPnljihY06T5ydjTpPnljihY06T5ydjTpPnljihY06T5ydjTpPnljihY0
6T5ydjTpPnljihY06T5ydjTpPnljihY06T5ydjTpPnljihY06T5ydjTpPnljihY06T5ydjTpPnljihY0
of8A3j9P7nY06T5y7XHws6xH5gkAAAAAAAAAAAAAAAja+nCONuha0
z07XjPx07WsV+MzzkHK10netxshmwRfjHCy0CzWC1Lun70lVR580LYcdudY+g5X45+MI2fzsXsfNOuOle
VYhd+bLjwXvPHdTgkk1iK8JjvTFdJzJ90VtrwnhMc4dRtXXjHCYdrbe+ExzgV0AAAAAAAAAAAAAAA
AAAAAAebW04RxtIfTru15/8ADcd3Z4683a13Y+M85dBkybPavGnGOimeHPg9Fy1K29qlkYvErzxsNz8c/
4/7djBjj/HzRj+bHWs2nSsTLt2fd434z0xxERGkRoK3OJBG1d7jHCY5SkDaNbcd23C3/UnLVi0aSjFprOl/pl
JgAA
AAAAAAAAAAAAAAAAAAAAATMRGsg1TM34V4R3yccnwr/ANTiNI0gh
K1isaQ6AAAAAAAAACNq68a8LQkA5W29Hx74dRtWdd6vtf8AXa2i0f8AY6A6AAAAAAAAAAAAAAA
DlrbsdZnlAOwtpwjjM8odrXd58ZnnLla6cZ42lIAAAAAAAAAAAmlmNJAENzX8+NevRPmlaTTjXjXvgEwiY
tGsAA
AAAAAAAAAAAAAAAAAAAAAAOWtFfjm8oAtaKxrKMVm8625d0O1rx3rcZ/
wCJAAAAAAAAAAAAAI2rOu9Xn/ANSAcraLrrDqNqzE71effHV2totGsA6AAAAAAAAAAAdlrRNW
ZAtaKx8e6HK1nXetzn/RWs671uf8AxIAAAAAAAAAAAEJrMTvU598dUq2i0fHvh1G1NeMcLaki1
vrOk8LdEgAAAAAAAAAAAAAAAABTk2iteFeMqLzslv8tI6QM3qRtHn71vFPm7GS9eV5Gf6
RvGwm0zHC8a/GGml63jW6jcsv06CE2m07tPrItr8d2vG3/Ctd3jPGZ73a1isaQ6AAAAAAAAAAAAAAA
AAAAajaJrO9X6x1Scvetl1tlfZi0aw6yWzzvTOONIIxN7zttlxe5G8efvW8U+adc+SvfrHxCfJG0VY9orfhPCVo
3LoAACOTJXHGtpB1ZL7TafZjhXN7zttlxe5G8efvWjlafNZTPkrnej4hPkjXaYrGsuViZnetz7o6K8eWuS/3
uE90LhsAAAAAAAAAAAAABY1YtHxjIKMWmj3b8+6eqZMRaNJAElmacLca90uZM1cfPjPSAWD
HfaL25fdj4K5taedp8xi/JHoDz4taOvp81tNovX2vvQE7laxDHlrk5c+iY2AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAABTk2iteFeMqLzslv8tI6QM3qRtHn71vFPm7GS9eV5Gf6
RvGwm0zHC8a/GGml63jW6jcsv06CE2m07tPrItr8d2vG3/Ctd3jPGZ73a1isaQ6AAAAAAAAAAAAAAA
AAAAajaJrO9X6x1Scvetl1tlfZi0aw6yWzzvTOONIIxN7zttlxe5G8efvW8U+adc+SvfrHxCfJG0VY9orfhPCVo
3LoAACOTJXHGtpB1ZL7TafZjhXN7zttlxe5G8efvWjlafNZTPkrnej4hPkjXaYrGsuViZnetz7o6K8eWuS/3
uE90LhsAAAAAAAAAAAAABY1YtHxjIKMWmj3b8+6eqZMRaNJAElmacLca90uZM1cfPjPSAWD
HfaL25fd

zbhMxETM9yvSrZYx162nIDHa02nW06y7e83tNpRRw7630ADmAANODNr9y/PulmBvnRHoGG/aY4mecc01egAAAAAAAAAAAAAAABHJfcNgV7Rm3Y3a85/wBMjszMzrPOXEfrQAYAAAdiZidYnSWvBm7SNJ9qP8AbG7W01tExzgb46x6A5S0XrFo73VegAAAAAAAAAAAAAAAABiz23ss9l4K3be3b5y4jzdfYCWOOk3tolnhxb86zyhzLu72lI0iF+SLVpu0qhbB/5xMRO8O159ZGcdmJidJcHMbsFt7FWe9ha9k91PzHT41wCuwAAA
AAAAAAAAAAAAAAAArz23cU9Z4LFG1+7r8xL9MoCPML9mjtbWFC7DkrjidYkXn7TtmrW0xuclF7Ra0zEaNFZxJ03eKnNTcvpHKRrrcVgDmt2e27ljpPBdtVtMWnWWbH7ynzhdt+H1V35v+WcBHAWSu348oVtIY/
8ojXTgN8zar7HHrpv8VFoiJmlnWGiMWO2u7adVF6zS0xIdREAYX7LbS816w1MWD39W1Xo4/5ABsAAA
AAAAAAAAAAAAAAZtrtxrX6tLhtXvvom930qAR5hdXZ5tWJ3o4/BStmmce5p9Rqz+p/Zp8ueSrJtctu66tGCu7jm097Pe29aZGupMRAHNq2S33zr0xsuye8t8mpXp5+gAaAAAAAAAAAAAAAAAYMsbuW0fFFo2qmkkxePIlojz9zKJvvNJ1rOiIMre3yeL/S7jk0x61tGrIDU6rszMzrPNwBkbNmjTDHx4slym1orHOW+sbtYjorr8c/XQB1AAAAAAAAAAAAAAFO1Rri16TquctXerMdQrzx2YmszE84cR5bMF2LJStdLQpAlxpjLirxrHH5Kcl5vbVAFvVoAMp4Y3stY+Oq/a4+7Weko7JTjN5+UL8tN/HNVejmf5xgD5iOA1YrfxFuzOk6aMoLzcaseLspm02UzbRbjJMxyR1nrLgXr1kABlbs0a5o+EatijZaaVm0969Xp5mQAGgAAAAAAAAAAABk2uNMkT1hrVbTTex6xzji1NjGAjyiVK714hF2tprOsTpIsac9t3HFY72V21ptOtp1cF6u0AEaNkj71p+jSr2eu7ijrPFYr0yZMABQAAAAAAAAAAAAAAALWLisayDmSK2pMW5MNqzWdJ+jbFZtOt+XdDuTHXJXSYGeufKMAAsyYb07tY6wrRwss+wAZB2tZtO1Y1acOz7v3r8Z6DfPFrmDD07vTwt3fBdW2vC3C0JOWrvR0mOUq7yY6l1trwnhaEhQA
AAAAAAAAAAAAAAAFG04tfv1jj3sr0VGbz9771OE9Bz7433GUdtE1nS0aS4jjZgAAIjpOS2kfWUseG9+7SOstdKVpXSsDpxz+12tYrWjIjDoK7M204tJ36xw72d6LN1fjrj8hz7433GcJiYnSY0n4iOOYAALMOOClvhH
OXceC1+NuENFI7OYp3TykdOeP2rliljSOQCuwAAAAAAAAAAAAADHnxndnbWPZn/Spuv8Afncjl3s+TZ5rxpxjojl3x+xSHfp3g5AEcZ0jjIC7Z8W/O9aPux/t3Fs8zxvwjo1RERGkcldeOM91CYmnGvGO+EzqF
o1h1G1Zid6nPvjqOqQ5W0W+cc4dAAAAAAAAAAAAAAABG1uO7XjP/Adtbd4c5nldla8d63Gf+FaxXjzmecpAAAI
Ww0tzrCYCn7Lj/N5uxs+OO7X5rQT15FYrHCNHQFAActXe+ExyltteE8LQk5auvGOExykHRGtteE8JhIAAA
AAAAAAAAAAAAAAAHjrFo4xqrnZsc92nyWgKfsuP4+adcVKcqwmCYACgAAAOWrW0feiJVzs2Oe6Y+S0BT9mx/HzWVx0p7NYSAwctWLRpLoCNLTru25x/tJG9d6NY4THJ2lt6PjHOAdAAAAAAAAAAAAAR
tadd2vP/he2nCONp5O1rux1mecgVrFY0h0ARTjrf2qxKudmxz1j6rgMVRs2Oos/OVlaVr7NYh0DAAAAEb
U14xwtHeVtrOluFknLVi0cfplOiEWms7t/pPVMAAAAAAAAAAAAAAAJnTmhxycuFevUCbTadKfWUq1isaQ7EREaQAAA
AAAAAAAAAAAA5auvGOEx3lba8J4THOHUbV14xwmASHK214TwmOcOgAAAAAAAAAAAAAA
AAAAAAAAAAAAAI3iYnerzjnHViByxaNYdQn7lt6PZnmmaaaaaaaaaaaaa5a27HxnLBa0VjWXK1nX
etz/AOAUpxnjaUgAAAAAAAAAAAmImNQ1nHwnjXr0TABDScfGONenROjY1gAAAAAAAAAAAAAA
AAAAAAAAAAAAAAHjtFy1mdIB0U22mkctZR+1/knzE8o0CiNqrPOJhbW9bx92YkJZUgBRyZisayWtFfn3Q5WszO9fn3R0ByIm/G3CvdCYAAAAAAAAAAAAAA
AAAAAajauvGOFodrbe58JjnDqNq68Y4WgEhytt74TZOHzmKxrM6QAKbbTSOWs/JH7X+sfMTyjQKI2qs8
6zC2l639mYkJZUgBQAAAAAAAAAAAAAAABCP8Aztu7M8vgmjktSK/fmIBIZ42mKxppNt
O8+1/knzE8o0Cmu00nnrC2totGsTrAu66AAAAADlrRWZNzSAdFNtppHLWUuftf5J8xPKNBmxEazyURtVZ5
xMjxNcttYtE1juCWV2sTad630hMBQAAAAAAAAAAABCazWd6n1jqmA5W0Wjg6jams614WK214
TwtHcCQAAqttGOvfrPwQ+1x3UnzEtkaBnja076zC2mal+U8egSypgCgAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAI3znZ01755AjmzRj4Rxt0ZbWm862nVyZmZmZ5y4jh13aADmOx
MxOsTpLgK1Ydo1ndvz6rbW04RxswNWy3iYms+116q7cd76q2tNj1nbqkA6AAAAAAAAAAAAAAA
AAAIzcnZ017+4EM+SKTExp32a97XnW06uTMz2MzPGZcRw67t+gAcwiZidYnSQFacO0azFb+bQ85q2bLv
RuW5xyV24731V4A6AAAAAAAAAAAAAAKdp7sbteciW45m2jSd2nPqzTMzOsrlgjh11aA
DA7W00nWs60AsuNmHNGThPCy158TMTExzhtw500pr3xzV3461MEM2Ts6a988htHnmjHwjzktabzr
adSzMzZmZ5y4jh13aADmOxMxOsTpLg9NWHaNZ3b8+q95zXs2XejdtzhXbjvfVXAdoAAAAAAAAAAA
AAAOWrFvhMcpcdctaK1mZ5QCFsvZxO/z7viv5Mtsk8Z0io5kvOS+9P0RRx67/IADkACrsW0TxhfiHvriYm

AAAAAAAAAAAAAAAAAAhPv6/pn9k0J9/X9M/sCYAAAAAAAAAAAAAAAIAU97k+iaFPe5PomAAAAAA
AAAAAAAAAAAAAAAAAAAAAAIAU97k+iaFPe5PomAAAAAAIAU97k+iaFPe5PomAAAAAAIAU97k+iaFPe5PomAAAAAA
6/pn9k0J9/X9M/sCYAAAAAAAAAAAAAAIAU97k+iaFPe5PomAAAAAAIAU97k+iaFPe5PomAAAAAAIAU97k+iaFPe5PomAAAAAA
6/pn9k0J9/X9M/sCYAAAAAAAAAAAAAAIAU97k+iaFPe5PomAAAAAAIAU97k+iaFPe5PomAAAAAAIAU97k+iaFPe5PomAAAAAA
6/pn9k0J9/X9M/sCYAAIXz46W0telkExV9pw/iQtrMWijidYkAAAV5M+PH7Vo16K/tuHXTj5A0COPLTjh3LRKQAAI3vXHGt7REAkIUzY7z
pW0TKYAAA5a0VjW06QottuGO+Z+UA0Cmm1YrzpFtPmuAAHLWisa2nSFFtsw1nnM/KAaBVTacV50i
2k/FaAAAI3vXHGt50hD7Th/EgFoq+04fxIWUvW9dazrAOgAcQdpwxPvIpT0H8SAWir7Th/EhbE6xrAAAA
TMRGszpCi214a/5TPyBeKK7ZhnmPmvrMWjWJ1gAEb5K441valgEhV9pw/iQnTJTJruWidASBXkz48fC1
uPSAWDPXbcMzprMfOF9bRaNazrAOgACN8lcca3tEQh9pw/iQC0VfacP4kH2nD+JALQiYmlmOUgAhfNj
xzpe0RKp2nD+JALRyt63rvVnWOqv7Th/EgForjaMVpilvEzKwAEb5aY9N+0RqCQq+04fxITpeuSNaTrAJDI
r1pGtpiUTtuGO+Z+UA0Cqm04rzpFuPxWgAAAAAAAAAhT3uT6JoU97k+iYAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAC+E/r+mf2TQn39f0z+w
JgAMm0bjbLlm0WiInq1gPhVXcvNZ7nqbP7inyebtHv7/N6Wz+4p8gWI5ltNjs6T1SAebGy5b5Ji3DrMrb
bBpxhfi2oZstcVjtP0B5dL2xZImOExl1qzvViesavJpE5csR3zL1qxpWI6QDoAEzFYmZ4RDy9ozThycOXdC7
bdo1ns6zw7zYsGv/raOEcgX7Lg7Kms+1PNez/bMe9u8dddGgAAHm7Zkm+aa68K8FmDY4vji17Tx5RCG
24ppk345WdwbZOOKVtXeiOQlbTs/YTGk6xLRsGSbVmkrpyZto2ic8xw0iOUNew4tzHNp52BpAB5u2ZZ
vlmuvcvBZg2KL0i15mNe6FG01muae2vfOrzs+045xxFrRWYjTiDLtOzTg0mJ1rlTsOab1mlp1mOSrbdkr
FKTr1IxsvtNorHXgD0wAZP6jP/AJ1j4s2DZ7Z4mYml06r/AOozxpCrZtpjBWYmuus9QT+wX8dWvZ8U4sU
VmYmfgqpt2OZ4xNWmJi0axOsSATxiQBhtsF5tM79eKvLss4qb1r1+T0pnSJI5W0Zpy5JmeUcoBHFjtLvFa
vWrG7WI6Qy7FfFWu7E/fnnq1gHIV7RbdwXn4AwbTnnLeYifuxyhZi2Kb1i153de5nxTWMIZtyieLVbb+P
3KcPiBbYPDFzbKVillHKGXHt0TOl66fGGujY1jkAp2nDOakRExGk964B5WfBOCYiZidejT/TvZv80f6j7VPql
/TvZv8AMF21Zeyxax7U8Iefix2z5NI598yy/qFv/Stfhqs/p9fuWnrlKsuxWpSbVnXTuQ2XNOLJETP3Z5vTe
Rljdy2jpIPXEcVt7FWesJAy/wBQ93X5suDZ7Z4nSYjTq9HLirlitrwcxYa4dd3XiDFYrUpNpvXSGesTa0Rhe
27fk0rGOO/jKvYcW9k355VBvpXdpWvSNHQBm2nZrZrxMWiNOrDlxziyTSZ1mHrvM2z+5v9P8AgNmYc
dmhIvsV6Um02rwa9j/t6p5/c3+QPM2f3+P9UPWeTs/v8f6oesAo2rZ7Z93dml06rwHk5sM4b7szE/Jt2D
3M/NRt/vo+S/YPcz8wU7VgzWya+1E8vgIxyj0+9fj8G0mYiNZ5A8nNinDfdmfLfseScmHjzjgxbXljLI1jHC
G3Y8c0wRrzniC8AAAAAAKe9yfRNcnvcn0TAaaaaaaaaaaaaaaaaaaaaaaa
AAAAAAAAAAAAAAAAAAAAAAQn39f0z+yaE+/r+mf2BMAAAhK7R7+/wA3pbP7in
yebtHv7/N6Wz+4p8gWA5ad2sz0BHLripvW+kPMYxvtGTrM8oczBzB71vpCWHP2MzMUIiZ6yDds2zxh
rrPG0817B9vv4KtmDJOXFw8xpMgmp2rJOLDmxznguRyY65a6XjWAeRE/e1ni1Rt0xXSMcRHzaPseDwf
7knZMMRP3P8Acg83e+/vfHVv2fapy5lpNIjhz1YYi00i07V6eLZ8eOYtWuk6dQWgA8nPknJlmZnv0hsw7
JjjHE3jWZhk2nFOLLPDhPGJaMO2xWkVvWzm0+Ac2vZq46b9OHWHNgYTGTc7phHadq7au7WNKp7Bi
ne7SY4dwNwAK82CmaPvc45SyzsFteF40XbTtPY2rErPer+3109idQV5tmrgw6zOtp5i7FXe2iJ6cUM2a+
e/H6RDbseCcVNbe1INAk8uz0zTE314dJV/YsPSfNTn2q9M87vCl4aT3uxt/DjTj8AVbVs8YZiazrErv6feZ3
qd0cWfpntnmNY0iOUNWw4ppSbWjSbcgagcvbcpNukA7MaxMM1thxzYmYlnptmSsZr96JnlKydv4cKcQ
ZclJxZJrrxjvens9+0w1tPN5kzbNk152l6mGnZ4q16Amr2iN7BePgsJjWJie8HjVjetEdXp02XFwuk0ifjLBnx
Ww5fhziWmm313fvVnX4Ap2vDGLJG7ylp2C02wzE90sefNOfJrp8lhv2THOPDETzniC4AGH+o+1T6pf07
2b/ADR/qPtU+qX9O9m/zBD+oV/9K2+Giz+n2+5aOkrdqxdri0j2o4w8/Fktgyaxz74kHrPlyzvZbT1lfI2216T
Wsaa96Gy4Zy5lmY+7HMHo4o3cVY6QkABM6RMz3DPtuXcxbSc7AwZrzky2t8eD0tmx9nhiO+eMsGyY+
0zRryjl1AAAHmbZ/c3+n/HpvM2z+5v9P8AgNux/wBvVPP7m/yQ2P8At6p5/c3+QPM2f3+P9UPWeTs/v
8f6oesAADz9v99HyX7B7fmfo2/30fJfsHuZ+YNMzERrPJ521bTOSdyk/d/6ltue03nHHCI5/FlpbdEtETp3
SDVsmy72l7xw7ob2D7ffwVX7LtNs1piaxGnQGgAAAAAAKe9yfRNcnvcn0TAaaaaaaaaaaaaaa
AAAAAAAAAAAAAAAAAAAAAAQn39f0z+yaE+/r+mf2BMAAAFc7PitMzNImZWViKxERGkQABMRMaTyAFX2bD+HB9mw/hwtAVfZsP4cLK1iYrWNIjudAAA
AAV/Z8Ouu5GqwAAActSt40tETCi2xYZ7pj5S0Ak7Jir/jr810RERpAAAhhkw48ntV1nqq+xYde/wA2gbXj
wY8fs1jXqsAAAEI4qZParEqp2LDPWPILQAqps2Kk6xXWfitAAAFV9nxX4zXj8EPsWLXX73m0AIY8NMfs1i

EwAABy1K3jS0RMKJ2LDM8pj5S0AK8ez48c61rx6ysAAAEb4qZNN+sToUx0x67lYjVIAV5MGPJxtXj1hYAz
 12LDE66TPzlfWsVjSsaQ6AAAIxUyTresSmAjTFTHruViNUgAAAV2wYr23rUiZnvWAOVrWldKxpDsxFom
 JjWJAFCbPirMTFlYWAAACF8OPJOt6xMu0pXHGII0hIBXbBivbetSJlz7Nh/DhaAq+zYfw4SpipjnWIYhMA
 AAAAAAAABCnvcn0TQp73J9EwAA
 AAAAAAAAAAAAAAAAAAEJ9/X9M/sAJgAAAAAAAAAAAAAAAAAAAAAhT3uT6JgAAAAAAAAAAAAAAA
 AAAAAAAAAAAAAAAAAAD//Z"></figure><p>Constraint
 s:</p><p>0<size <100</p><p>0<data<1000</p><p>Input
 format:</p><p>First line indicates the size of the tree</p><p>Second line indicates the
 elements of the tree</p><p>Output Format:</p><p>single line indicates the pre-
 order list of the given value</p>

answer

```

#include<bits/stdc++.h>

using namespace std;

void solve(){}
struct node {
    int data;
    struct node *left,*right;
}*root=NULL;

void insert(int data) {
    struct node *tempNode = (node*) malloc(sizeof(node));
    struct node *current;
    struct node *parent;
    tempNode->data = data;
    tempNode->left = NULL;
    tempNode->right = NULL;
    if(root == NULL) root = tempNode;
    else {
        current = root;
        parent = NULL;
        while(1) {
            parent = current;
            if(data < parent->data) {

```

```

        current = current->left;

        if(current == NULL) {

            parent->left = tempNode;

            return;

        }

    }

    else {

        current = current->right;

        if(current == NULL) {

            parent->right = tempNode;

            return;

        }

    }

}

void preorder(struct node* root) {

    if(root != NULL) {

        printf("%d ",root->data);

        preorder(root->left);

        preorder(root->right);

    }

}

int main() {

    solve();

    int n,i,x; scanf("%d",&n);

    for(i = 0; i < n; i++){

        scanf("%d",&x); insert(x); }

    preorder(root);

    return 0;

    printf("struct node* newNode(int item) "); }


```

question

answer

```
#include<string.h>

struct cell{
    int name;
    int level;
    int capacity;
};

struct cell queue[1000001];
struct cell arr[1000001];
int front;
int end;
void init_queue(){
    front = 0;
    end = 0;
}

void enqueue(int name,int capacity,int level){
    queue[end].name = name;
    queue[end].level = level;
    queue[end].capacity = capacity;
    end = end + 1;
}

int is_empty(){
    if(end == front)
        return 1;
    return 0;
}

void dequeue()
{
    if(!is_empty())
        front++;
}
```

```

int main(){
    int n,rc;
    init_queue();
    scanf("%d %d",&n,&rc);
    int i,j,k;
    for(i=0;i<n;i++){
        scanf("%d %d",&arr[i].name,&arr[i].capacity);
    }
    enqueue(0,rc,0);
    i=0;
    while(!is_empty()){
        int par = queue[front].name;
        int cap = queue[front].capacity;
        int lev = queue[front].level+1;
        k=1;
        for(j=0;j<cap&&i<n;j++,i++){
            printf("%d %d %d\n",par,lev,k++);
            enqueue(arr[i].name,arr[i].capacity,lev);
        }
        dequeue();
    }
    return 0;
}

```

question

<p>Problem Description</p><p>You are given a directory tree of
<math xmlns="http://www.w3.org/1998/Math/MathML"><mi>N</mi></math>&br/>directories/folders. Each directory is represented by a particular
<math xmlns="http://www.w3.org/1998/Math/MathML"><mi>i</mi><mi>d</mi></math>&br/>which ranges from
<math xmlns="http://www.w3.org/1998/Math/MathML"><mn>1</mn></math>&br/>to&br/><math xmlns="http://www.w3.org/1998/Math/MathML"><mi>N</mi></math>. The id of the root

directory is <math xmlns="http://www.w3.org/1998/Math/MathML"><mn>1</mn></math>, then it has some child directories, those directories may contain some other ones and it goes on. Now you are given a list of directory id's to delete, you need to find the minimum number of directories that need to be deleted so that all the directories in the given list get deleted. </p><p>Note that if you delete a particular directory, all its child directories will also get deleted.</p><p>Constraints
$\text{mo} \leq \text{N} \leq \text{msup} \leq 10 \leq \text{mi} \leq \text{d} \leq \text{mi} \leq \text{f} \leq \text{mi} \leq \text{p} \leq \text{mi} \leq \text{a} \leq \text{mi} \leq \text{r} \leq \text{mi} \leq \text{e} \leq \text{mi} \leq \text{n} \leq \text{mi} \leq \text{t} \leq \text{mi} \leq \text{mo} \leq \text{mi} \leq \text{M} \leq \text{mi} \leq \text{mo} \leq \text{mi} \leq \text{N}$
$\text{mi} \leq \text{mtext} \text{mi} \leq \text{t} \leq \text{mi} \leq \text{o} \leq \text{mi} \leq \text{f} \leq \text{mi} \leq \text{mtext} \text{mi} \leq \text{p} \leq \text{mi} \leq \text{a} \leq \text{mi} \leq \text{r} \leq \text{mi} \leq \text{e} \leq \text{mi} \leq \text{n} \leq \text{mi} \leq \text{t} \leq \text{mi} \leq \text{mo} \leq \text{mi} \leq \text{N} \leq \text{mi}$
$\text{mi} \leq \text{mtext} \text{mi} \leq \text{t} \leq \text{mi} \leq \text{o} \leq \text{mi} \leq \text{b} \leq \text{mi} \leq \text{e} \leq \text{mi} \leq \text{d} \leq \text{mi} \leq \text{e} \leq \text{mi} \leq \text{l} \leq \text{mi} \leq \text{e} \leq \text{mi} \leq \text{t} \leq \text{mi} \leq \text{e} \leq \text{mi} \leq \text{d} \leq \text{mi} \leq \text{mo} \leq \text{mi} \leq \text{N} \leq \text{mi}$

Input
The first line of input contains an integer $\text{mi} \leq \text{N} \leq \text{mi}$ that denotes how many folders are there.
The next line contains $\text{mi} \leq \text{N} \leq \text{mi}$ space separated integers that where the $\text{mi} \leq \text{mtext} \text{mi} \leq \text{t} \leq \text{mi} \leq \text{h} \leq \text{mi}$</mrow></msup></math> integer denotes the id of the parent of the directory with id $\text{mi} \leq \text{i} \leq \text{mi}$. Note that the first integer will be $\text{mi} \leq \text{i} \leq \text{mi}$ as&nbs;p;$\text{mi} \leq \text{mtext} \text{mi} \leq \text{t} \leq \text{mi}$$\text{mi} \leq \text{N} \leq \text{mi}$ is the id of root folder and it has no parent. Rest of the integers will not be $\text{mi} \leq \text{i} \leq \text{mi}$.
$\text{mi} \leq \text{mtext} \text{mi} \leq \text{t} \leq \text{mi}$$\text{mi} \leq \text{N} \leq \text{mi}$
The next line contains an integer $\text{mi} \leq \text{M} \leq \text{mi}$ that denotes how many directories you need to delete.
The next line contains $\text{mi} \leq \text{M} \leq \text{mi}$ space separated integers that denote the ids of the directories you need to delete.

Output
Print the minimum number of directories that need to be deleted.</p><p>Explanation for test case 1</p><p>In the image below is the complete directory tree. If you delete the directories 2 and 3, 4 gets automatically deleted, thus you need to delete only 2 directories. </p><figure class="image"><p> </p><p> </p><p> </p></div>

answer

```
#include <stdio.h>
```

```
#include <stdlib.h>

#define pcx putchar_unlocked
#define gcx getchar_unlocked
typedef long int lint;

lint getnl() {

    lint n = 0; auto neg = 0;
    register int c = gcx();
    if ('-' == c) { neg = 1; c = gcx(); }
    while(c<'0' || c>'9') c = gcx();
    while(c>='0' && c<='9') {
        n = n * 10 + c - '0';
        c = gcx();
    }
    if(neg) n *= -1;
    return n;
}

void pputl(lint li,char lc) {

    if (0 == li) {
        pcx('0'); if(lc) pcx(lc); return;
    }
    char s[24]; lint idx = -1;
    while (li) {
        s[++idx] = '0' + li % 10;
        li /= 10;
    }
    for (lint jdx=idx; jdx>=0; --jdx) pcx(s[jdx]);
    if(lc) pcx(lc);
}

int main () {
```

```

int N = getnl();
int PA[N+1];
for (lint ni=1; ni<=N; ) PA[ni++] = getnl();

lint D = getnl();
int DA[D];
char DLT [100001]={0};
for(lint ni=0; ni<D;) {
    DA[ni] = getnl(); DLT[DA[ni++]] =1;
}
if (DLT[1]) {
    pputl(1, 0); return 0;
}

lint dLess =0;
for (lint ni=0; ni<D; ) {
    //printf ("D:%d ", DA[ni]);
    lint pi = PA[DA[ni++]];
    char piv [100008]={0} ;
    while (pi>0) {
        if (piv[pi]) { ++dLess; /*printf ("C:%d", pi);*/ break; }
        else piv[pi] = 1;
        //putl(pi, ' ');
        if (DLT[pi]) { ++dLess; break; }
        pi = PA[pi];
    }
    //pcx('\n');
}
pputl(D-dLess, 0);
return 0;
}

```

question

<p>Problem description</p><p>When the king of ghosts notices that all humans on Planet Earth have lost their dread of the ghost race, he is extremely unhappy. He understands why this is happening. The existing ghost species has gotten incredibly lethargic and has ceased visiting Planet Earth in order to terrorise humanity. As a result, he plans to arrange a tournament to urge the whole ghost race to scare the humans. The monarch, on the other hand, never comes to Planet Earth.</p><p>This competition will go on for N days. Currently, there are a total of M ghosts (apart from the king) existing in the ghost race such that :
- The youngest ghost is 1 year old.
- The oldest ghost is M years old.
- No two ghosts have the same age.
- The age of each and every ghost is a positive integer.</p><p>Every day of the tournament, ghosts must visit Planet Earth in order to frighten humans. The ghost that terrifies the most humans on any given day is named "Ghost of the Day" at the conclusion of the day. The king of ghosts, on the other hand, is a firm believer in constancy. After this title is granted, a "Consistency Trophy" is offered to the ghost who has earned the most of these titles up until that point. If there are several such ghosts, the prize is handed to the eldest among them. It's worth noting that "Title Giving" and "Trophy Giving" take place at the end of each tournament day.</p><p>Each day of the competition, you will be told the age of the ghost that earned the "Ghost of the Day" title. Your task is to determine the age of the ghost who received the "Consistency Trophy" on each competition day.</p><p>Input
The first line consists of 2 space separated integers N and M. The next line consists of N space separated integers such that the ith integer denotes the age of the ghost who was awarded with the "Ghost of the Day" title on the ith day of the competition.</p><p>Output
Print N lines. The ith line should contain 2 space separated integers such that the first integer denotes the age of the ghost who was awarded with the "Consistency Trophy" on the ith day and the second integer denotes the number of "Ghost of the Day" titles won by this ghost until the end of the ith day of the competition.</p><p>Constraints
1 ≤ N ≤ 10⁵
1 ≤ M ≤ 10⁹</p>

answer

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    int i,n,m;
    cin>>n>>m;
    unordered_map<int,int> ghost;
```

```

int best = -1;

cin>>best;

ghost[best]+=1;

cout<<best<<" "<<1<<endl;

for(i=0;i<n-1;i++)

{

    int y;

    cin>>y;

    ghost[y]+=1;

    if((ghost[y]>ghost[best]) || (ghost[y]==ghost[best] && y>best)) best = y;

    cout<<best<<" "<<ghost[best]<<"\n";

}

}

```

question

<p>Problem Description:</p><p>Any sequence A of size n is called **B-sequence** if:</p><p>$\text{xmlns}=\text{"http://www.w3.org/1998/Math/MathML"}$<math>\langle \text{msub} \rangle \langle \text{mi} \rangle A \langle / \text{mi} \rangle \langle \text{mn} \rangle 1 \langle / \text{mn} \rangle \langle / \text{msub} \rangle \langle \text{mo} \rangle < \langle / \text{mo} \rangle \langle \text{msub} \rangle \langle \text{mi} \rangle A \langle / \text{mi} \rangle \langle \text{mn} \rangle 2 \langle / \text{mn} \rangle \langle / \text{msub} \rangle \langle \text{mo} \rangle < \langle / \text{mo} \rangle \langle \text{mo} \rangle . \langle / \text{mo} \rangle \langle \text{mo} \rangle . \langle / \text{mo} \rangle \langle \text{mo} \rangle < \langle / \text{mo} \rangle \langle \text{msub} \rangle \langle \text{mi} \rangle k \langle / \text{mi} \rangle \langle / \text{msub} \rangle \langle \text{mo} \rangle > \langle / \text{mo} \rangle \langle / \text{msub} \rangle \langle \text{mi} \rangle A \langle / \text{mi} \rangle \langle \text{mrow} \text{ class}=\text{"MJX-TeXAtom-ORD"} \rangle \langle \text{mi} \rangle k \langle / \text{mi} \rangle \langle \text{mo} \rangle + \langle / \text{mo} \rangle \langle \text{mn} \rangle 1 \langle / \text{mn} \rangle \langle / \text{mrow} \rangle \langle / \text{msub} \rangle \langle \text{mo} \rangle > \langle / \text{mo} \rangle \langle / \text{msub} \rangle \langle \text{mi} \rangle A \langle / \text{mi} \rangle \langle \text{mrow} \text{ class}=\text{"MJX-TeXAtom-ORD"} \rangle \langle \text{mi} \rangle k \langle / \text{mi} \rangle \langle \text{mo} \rangle + \langle / \text{mo} \rangle \langle \text{mn} \rangle 2 \langle / \text{mn} \rangle \langle / \text{mrow} \rangle \langle / \text{msub} \rangle \langle \text{mo} \rangle > \langle / \text{mo} \rangle \langle \text{mo} \rangle . \langle / \text{mo} \rangle \langle \text{mo} \rangle > \langle / \text{mo} \rangle \langle / \text{msub} \rangle \langle \text{mi} \rangle A \langle / \text{mi} \rangle \langle \text{mi} \rangle n \langle / \text{mi} \rangle \langle / \text{msub} \rangle \langle / \text{math} \rangle \ \text{where} \ \langle \text{math} \text{ xmlns}=\text{"http://www.w3.org/1998/Math/MathML"} \rangle \langle \text{mn} \rangle 1 \langle / \text{mn} \rangle \langle \text{mo} \rangle \leq \langle / \text{mo} \rangle \langle \text{mi} \rangle k \langle / \text{mi} \rangle \langle \text{mo} \rangle \leq \langle / \text{mo} \rangle \langle \text{mi} \rangle n \langle / \text{mi} \rangle \langle / \text{math} \rangle . \text{That is, a sequence which is initially strictly increasing and then strictly decreasing (the decreasing part may or may not be there).}</p><p>All elements in A except the maximum element comes atmost twice (once in increasing part and once in decreasing part) and maximum element comes exactly once.</p><p>All elements coming in decreasing part of sequence should have come once in the increasing part of sequence.</p><p>You are given a B-sequence **S** and **Q** operations. For each operation, you are given a value **val**. You have to insert **val** in **S** if and only if after insertion, **S** still remains a B-sequence .
After each operation, print the size of **S**. After all the operations, print the sequence **S**.</p><p>**Hint**: Think of using some data structure to support insertion of elements in complexity better than linear.</p><p>Input

Constraints

$$\begin{aligned}
 & \text{xmlns="http://www.w3.org/1998/Math/MathML"}><\text{mn}>1</\text{mn}><\text{mo}>\leq</\text{mo}><\text{mi}>\text{N}</\text{mi}><\text{mo}>\leq</\text{mo}> \\
 & <\text{msup}><\text{mn}>10</\text{mn}><\text{mn}>5</\text{mn}></\text{msup}></\text{math}><\text{br}><\text{math}> \\
 & \text{xmlns="http://www.w3.org/1998/Math/MathML"}><\text{mn}>1</\text{mn}><\text{mo}>\leq</\text{mo}><\text{msub}><\text{mi}>\text{S}</\text{mi}><\text{mi}>\text{i}</\text{mi}></\text{msub}><\text{mo}>\leq</\text{mo}><\text{msup}><\text{mn}>10</\text{mn}><\text{mn}>9</\text{mn}></\text{msup}></\text{math}><\text{br}><\text{math}> \\
 & \text{xmlns="http://www.w3.org/1998/Math/MathML"}><\text{mn}>1</\text{mn}><\text{mo}>\leq</\text{mo}><\text{mi}>\text{Q}</\text{mi}><\text{mo}>\leq</\text{mo}> \\
 & <\text{msup}><\text{mn}>10</\text{mn}><\text{mn}>5</\text{mn}></\text{msup}></\text{math}><\text{br}><\text{math}> \\
 & \text{xmlns="http://www.w3.org/1998/Math/MathML"}><\text{mn}>1</\text{mn}><\text{mo}>\leq</\text{mo}><\text{mi}>\text{v}</\text{mi}><\text{mi}>\text{a}</\text{mi}><\text{mi}>\text{l}</\text{mi}><\text{mo}>\leq</\text{mo}><\text{msup}><\text{mn}>10</\text{mn}><\text{mn}>9</\text{mn}></\text{msup}></\text{math}><\text{br}> \\
 & \text{Given sequence } \text{S} \text{ is a } \text{B-sequence}. \\
 & \text{Format:} \\
 & \text{First line consists of an integer } \text{N}, \text{ denoting size of } \text{S}. \\
 & \text{Second line consists of } \text{N} \text{ space separated integers, denoting elements of } \text{S}. \\
 & \text{Next line consists of an integer } \text{Q}, \text{ denoting number of operations}. \\
 & \text{Each of the following } \text{Q} \text{ lines consists of an integer } \text{val}. \\
 & \text{Output Format:} \\
 & \text{After each operation, print the size of } \text{S} \text{ in a new line}. \\
 & \text{After all operations, print the sequence } \text{S}.
 \end{aligned}$$

answer

```

#include<bits/stdc++.h>
#include<map>
using namespace std;

int main() {

    int N,i,maximum=INT_MIN;

    scanf("%d", &N);

    int S[N];

    map<int,int> map;

    for(i=0;i<N;i++) {

```

```
scanf("%d", &S[i]);\n\nmaximum=max(maximum,S[i]);\n\nmap[S[i]]++;\n\n}\n\nint temp,Q;\n\ncin>>Q;\n\nfor(i=0;i<Q;i++) {\n\n    scanf("%d", &temp);\n\n    if(temp==maximum) printf("%d\\n",N);\n\n    else {\n\n        if(map[temp]>=2) printf("%d\\n",N);\n\n        else {\n\n            map[temp]++;\n\n            N++;\n\n            printf("%d\\n",N);\n\n            maximum=max(maximum,temp);\n\n        }\n\n    }\n\n}
```

```

    }

}

}

for(auto it=map.begin();it!=map.end();it++) printf("%d ",it->first);

for(auto it=map.rbegin();it!=map.rend();it++) {

    if(it->second>1) printf("%d ",it->first);

}

}

```

question

<p>Question description</p><p>Given two rooted trees, your task is to find out if they are <i>isomorphic</i>, i.e., it is possible to draw them so that they look the same.
Constraints</p>1≤t≤10002≤n≤10⁵the sum of all values of n is at most 10⁵<p>
Input
The first input line has an integer t: the number of tests. Then, there are t tests described as follows:
The first line has an integer n: the number of nodes in both trees. The nodes are numbered 1,2,...,n, and node 1 is the root.
Then, there are n-1 lines describing the edges of the first tree, and finally n-1 lines describing the edges of the second tree.

Output

For each test, print "YES", if the trees are isomorphic, and "NO" otherwise.

 </p><p> </p>

answer

```
#include <bits/stdc++.h>

using namespace std;
```

```
#define rep(i, a, b) for(int i = a; i < (b); ++i)
#define trav(a, x) for(auto& a : x)
#define all(x) begin(x), end(x)
#define sz(x) (int)(x).size()
typedef long long ll;
typedef pair<int, int> pii;
typedef vector<int> vi;
```

```
vi h, id;
vector<vi> g;
map<int, vi> lvl;
```

```
void dfs(int i, int p) {
    trav(j, g[i]) if (j!=p) {
        dfs(j, i);
        h[i]=max(h[i], h[j]+1);
    }
    lvl[h[i]].push_back(i);
}
```

```
int main() {
    cin.sync_with_stdio(0); cin.tie(0);
    cin.exceptions(cin.failbit);
```

```
    int t;
    cin >> t;
    while(t--) {
        int n;
        cin >> n;
        int m=2*n+1;
```

```

g.assign(m, vi());
rep(i, 0, 2) {
    rep(j, 0, n-1) {
        int a, b;
        cin >> a >> b;
        a+=i*n, b+=i*n;
        g[a].push_back(b);
        g[b].push_back(a);
    }
}
g[0]={1, n+1};
h.assign(m, 0);
id.assign(m, 0);
lvl.clear();
dfs(0, -1);
if (h[1]!=h[n+1]) {
    cout << "NO\n";
    continue;
}
trav(l, lvl) {
    map<vector<ll>, int> u;
    trav(i, l.second) {
        vector<ll> cur;
        trav(j, g[i]) cur.push_back(3LL*n*h[j]+id[j]);
        sort(all(cur));
        if (!u.count(cur)) {
            int s=sz(u);
            u[cur]=s;
        }
        id[i]=u[cur];
    }
}

```

```

    }

    cout << (id[1]==id[n+1]? "YES\n":"NO\n");

}

return 0;
}

```

question

<p>Problem Description:</p><p>You're given a K -ary infinite tree rooted at a vertex numbered 1. All its edges are weighted 1 initially.</p><p>Any node&nbsnbsp;<math xmlns="http://www.w3.org/1998/Math/MathML"><mi> X </mi></math>&nbsnbsp;will have exactly&nbsnbsp;<math xmlns="http://www.w3.org/1998/Math/MathML"><mi> K </mi></math>&nbsnbsp;children numbered as:</p><p><math xmlns="http://www.w3.org/1998/Math/MathML"><mo stretchy="false">[</mo><mi> K </mi><mo>*</mo><mi> X </mi><mo>+</mo><mn>0</mn><mo>,</mo><mi> K </mi><mo>*</mo><mi> X </mi><mo>+</mo><mn>1</mn><mo>,</mo><mi> K </mi><mo>*</mo><mi> X </mi><mo>+</mo><mn>2</mn><mo>,</mo><mi> K </mi><mo>*</mo><mi> X </mi><mo>+</mo><mn>3</mn><mo>,</mo><mi> K </mi><mo>*</mo><mi> X </mi><mo>+</mo><mn>4</mn><mo>,</mo><mi> K </mi><mo>*</mo><mi> X </mi><mo>+</mo><mo> stretchy="false">(</mo><mi> K </mi><mo>*</mo><mi> X </mi><mo>+</mo><mo> stretchy="false">]</mo></math></p><p>You are given&nbsnbsp;<math xmlns="http://www.w3.org/1998/Math/MathML"><mi> Q </mi></math>&nbsnbsp;queries to answer which will be of the following&nbsnbsp;two&nbsnbsp;types:</p><math xmlns="http://www.w3.org/1998/Math/MathML"><mn>1</mn><mi> u </mi><mi> v </mi></math>; Print the shortest distance between nodes&nbsnbsp;<math xmlns="http://www.w3.org/1998/Math/MathML"><mi> u </mi></math>&nbsnbsp;and&nbsnbsp;<math xmlns="http://www.w3.org/1998/Math/MathML"><mi> v </mi></math>. <math xmlns="http://www.w3.org/1998/Math/MathML"><mn>2</mn><mi> u </mi><mi> v </mi><mi> w </mi></math>: Increase the weight of all edges on the shortest path between&nbsnbsp;<math xmlns="http://www.w3.org/1998/Math/MathML"><mi> u </mi></math>&nbsnbsp;and&nbsnbsp;<math xmlns="http://www.w3.org/1998/Math/MathML"><mi> v </mi></math>&nbsnbsp;by&nbsnbsp;<math xmlns="http://www.w3.org/1998/Math/MathML"><mi> w </mi></math>. <p>Constraints</p><p><math xmlns="http://www.w3.org/1998/Math/MathML"><mn>2</mn><mo> </mo><mo>≤</mo><mo> </mo><mi> K </mi><mo> </mo><mo>≤</mo><mo> </mo><mn>10</mn><mspace linebreak="newline">&nbsnbsp;</mspace><mn>1</mn><mo> </mo><mo>≤</mo><mo> </mo><mo>≤</mo><mo> </mo><mi> Q </mi><mo> </mo><mo>≤</mo><mo> </mo><msup><mn>10</mn><mn>3</mn></msup><mspace linebreak="newline">&nbsnbsp;</mspace><mn>1</mn><mo> </mo><mo>≤</mo><mo> </mo><mo>≤</mo><mo> </mo><msup><mn>10</mn><mn>18</mn></msup><mspace class="MJX-TeXAtom-ORD"><mn>18</mn></mrow></msup><mspace linebreak="newline">&nbsnbsp;</mspace><mi> U </mi><mo> </mo><mo>≠</mo><mo> </mo><mi> V </mi>

mi><mspace
 linebreak="newline"> </mspace><mn>1</mn><mo> </mo><mo>≤</mo><mo> </mo><mi>W
 </mi><mo> </mo><mo>≤</mo><mo> </mo><msup><mn>10</mn><mn>9</mn></msup></math><
 /p><p> </p><p>Input format</p>The first line contains two space-
 separated integers <math
 xmlns="http://www.w3.org/1998/Math/MathML"><mi>K</mi></math> and <math
 xmlns="http://www.w3.org/1998/Math/MathML"><mi>Q</mi></math>. Next <math
 xmlns="http://www.w3.org/1998/Math/MathML"><mi>Q</mi></math> lines contain queries
 which will be of 2 types:Three space-separated integers <math
 xmlns="http://www.w3.org/1998/Math/MathML"><mn>1</mn></math>, <math
 xmlns="http://www.w3.org/1998/Math/MathML"><mi>u</mi></math>, and <math
 xmlns="http://www.w3.org/1998/Math/MathML"><mi>v</mi></math>Four space-
 separated integers <math
 xmlns="http://www.w3.org/1998/Math/MathML"><mn>2</mn></math>, <math
 xmlns="http://www.w3.org/1998/Math/MathML"><mi>u</mi></math>, <math
 xmlns="http://www.w3.org/1998/Math/MathML"><mi>v</mi></math>, and <math
 xmlns="http://www.w3.org/1998/Math/MathML"><mi>w</mi></math><p>Output format</p><p>For each query of type <math
 xmlns="http://www.w3.org/1998/Math/MathML"><mo
 stretchy="false">(</mo><mn>1</mn><mi>u</mi><mi>v</mi><mo
 stretchy="false">)</mo></math>, print a single integer denoting the shortest distance
 between <math
 xmlns="http://www.w3.org/1998/Math/MathML"><mi>u</mi></math> and <math
 xmlns="http://www.w3.org/1998/Math/MathML"><mi>v</mi></math>. </p><p> </p>

answer

```

#include <iostream>
#include <map>
#include <assert.h>
using namespace std;
#define int long long

map < pair < int, int >, int > adj;

int find_depth( int u, int k ) {
    int depth = 0;
    while ( u > 0 ) {
        u = u / k;
        depth = depth + 1;
    }
}
  
```

```

    }

    return depth - 1;
}

int dist( int u, int v, int k ) {

    int dist = 0;

    int depth_u = find_depth( u, k );
    int depth_v = find_depth( v, k );

    if ( depth_u < depth_v ) {
        swap ( u, v );
        swap ( depth_u, depth_v );
    }

    while( depth_u != depth_v ) {
        if ( adj.count( { u, u / k } ) ) {
            dist = dist + adj[ { u, u / k } ];
        } else {
            dist = dist + 1;
        }

        depth_u = depth_u - 1;
        u = u / k;
    }

    while ( u != v ) {
        if ( adj.count( { u, u / k } ) ) {
            dist = dist + adj [ { u, u / k } ];
        } else {
            dist = dist + 1;
        }

        if ( adj.count( { v, v / k } ) ) {
            dist = dist + adj [ { v, v / k } ];
        } else {
            dist = dist + 1;
        }
    }
}

```

```

        }

        u = u / k;

        v = v / k;

    }

    return dist;
}

void add_weight( int vertex, int parent, int w ) {

    if ( !adj.count ( { vertex, parent } ) ) {

        adj[ { vertex, parent } ] = 1;

    }

    adj[ { vertex, parent } ] = adj[ { vertex, parent } ] + w;

}

void increase_weights ( int u, int v, int w, int k ) {

    int depth_u = find_depth( u, k );

    int depth_v = find_depth( v, k );

    if ( depth_u < depth_v ) {

        swap ( u, v );

        swap ( depth_u, depth_v );

    }

    while( depth_u != depth_v ) {

        add_weight( u, u / k, w );

        depth_u = depth_u - 1;

        u = u / k;

    }

    while ( u != v ) {

        add_weight( u, u / k, w );

        add_weight( v, v / k, w );

        u = u / k;

        v = v / k;

    }

}

```

```

    }
}

signed main() {

    int k, q, x, u, v, w;
    cin >> k >> q;

    while(q--) {
        cin >> x;
        if ( x == 1 ) {
            cin >> u >> v;
            cout << dist( u, v, k ) << "\n";
        } else {
            cin >> u >> v >> w;
            increase_weights( u, v, w, k );
        }
    }
}

```

question

<p>Problem Description</p><p>Football is Monk's favourite sport, and his favourite team is "Manchester United." Manchester United has qualified for the Champions League Final, which will take place at London's Wembley Stadium. As a result, he decided to go watch his favourite team play.</p><p>When he arrived at the stadium, he noticed that there was a long wait for match tickets. He is aware that the stadium has M rows, each with a distinct seating capacity. They could or might not be comparable. The cost of a ticket is determined by the row. If there are K(always higher than 0) empty seats in a row, the ticket will cost K pounds (units of British Currency).</p><p>Now, every football fan standing in the line will get a ticket one by one.
Given the seating capacities of different rows, find the maximum possible pounds that the club will gain with the help of the ticket sales.</p><p>Constraints:
1 <= M <= 1000000
1 <= N <= 1000000
1 <= X[i] <= 1000000
Sum of X[i] for all 1 <= i <= M will always be greater than N.</p><p>Input:
The first line consists of

M and **N**. **M** denotes the number of seating rows in the stadium and **N** denotes the number of football fans waiting in the line to get a ticket for the match.
Next line consists of **M** space separated integers **X[1],X[2],X[3].... X[M]** where **X[i]** denotes the number of empty seats initially in the i^{th} row.

Output:
Print in a single line the maximum pounds the club will gain.

answer

```
#include <bits/stdc++.h>
using namespace std;
#define PII pair<int, int>
priority_queue<int> seats;
map<int, int> x;
int main()
{
    int N, M; cin >> N >> M;
    assert (1<=N and N<=1000000);
    assert (1<=M and M<=1000000);
    for (int g=1; g<=N; g++){
        int a; cin >> a;
        seats.push(a);
        assert (1<=a and a<=1000000);
        x[a]++;
    }
    long long ans = 0;
    for (int g=0; g<M; g++){
        int x = seats.top(); ans+=x; seats.pop();seats.push(x-1);
    }
    cout <<ans;
    return 0;
    cout<<"void heapify(int arr[],int n,int i)";
}
```

question

<p>Question Description</p><p>In a movie festival, n movies will be shown. You know the starting and ending time of each movie.

Your task is to process q queries of the form: if you arrive and leave the festival at specific times, what is the maximum number of movies you can watch?

You can watch two movies if the first movie ends before or exactly when the second movie starts. You can start the first movie exactly when you arrive and leave exactly when the last movie ends.

Input

The first input line has two integers n and q: the number of movies and queries.

After this, there are n lines describing the movies. Each line has two integers a and b: the starting and ending time of a movie.

Finally, there are q lines describing the queries. Each line has two integers a and b: your arrival and leaving time.

Output

Print the maximum number of movies for each query.

Constraints</p>1≤n,q≤2·10^51≤a< b≤10^6

answer

```
#include<bits/stdc++.h>

using namespace std;

int dp[1000006][25];

void solve(){}
int main(){
    solve();
    int n, q; cin>>n>>q;
    for (int i = 0; i < n; i++) {
        int x, y; cin>>x>>y;
        dp[y][0] = max(dp[y][0], x);
    }
    for (int i = 1; i <= 1000000; i++)
        dp[i][0] = max(dp[i][0], dp[i-1][0]);
    for (int k = 1; k <= 20; k++)
        for (int i = 1; i <= 1000000; i++)
            dp[i][k] = dp[dp[i][k-1]][k-1];
```

```

while(q--) {
    int x,y; cin>>x>>y;
    int ans = 0;
    while(y>0) {
        int z = 0;
        for (int i = 0; i <= 20; i++) {
            if (dp[y][i] < x) {
                z = i;
                break;
            }
        }
        if (z == 0)
            break;
        ans += (1<<(z-1));
        y = dp[y][z-1];
    }
    cout<<ans<<endl;
}

```

}

question

<p>Question Description:</p><p>There are n cities and m flight connections between them. Your task is to add new flights so that it will be possible to travel from any city to any other city. What is the minimum number of new flights required?

Input

The first input line has two integers n and m: the number of cities and flights. The cities are numbered 1,2,...,n.

After this, there are m lines describing the flights. Each line has two integers a and b: there is a flight from city a to city b. All flights are one-way flights.

Output

First print an integer k: the required number of new flights. After this, print k lines describing the new flights. You can print any valid solution.

Constraints</p>1≤n≤10^51≤m≤2·10^51≤a,b≤n

answer

```
#include <stdio.h>
#include <string.h>

#define N 100000
#define M 200000

struct L {
    struct L *next;
    int j;
} aa[N], bb[N], aa_[N];

void link(int i, int j) {
    static struct L l91[M * 2], *l = l91;

    l->j = j;
    l->next = aa[i].next, aa[i].next = l++;
    l->j = i;
    l->next = bb[j].next, bb[j].next = l++;
}

void link_(int i, int j) {
    static struct L l91[M], *l = l91;

    l->j = j;
    l->next = aa_[i].next, aa_[i].next = l++;
}

int po[N], npo;
char visited[N];
```

```
void dfs1(int i) {
    struct L *l;

    if (visited[i])
        return;
    visited[i] = 1;
    for (l = aa[i].next; l; l = l->next)
        dfs1(l->j);
    po[npo++] = i;
}
```

```
int cc[N], dd[N];
```

```
void dfs2(int j, int c) {
    struct L *l;
    int c_ = cc[j];

    if (c_ != -1) {
        if (c_ != c) {
            link_(c_, c);
            dd[c]++;
        }
        return;
    }
    cc[j] = c;
    for (l = bb[j].next; l; l = l->next)
        dfs2(l->j, c);
}
```

```
int dfs3(int i) {
```

```

struct L *l;

if (visited[i])
    return -1;
visited[i] = 1;
if (!aa_[i].next)
    return i;
for (l = aa_[i].next; l; l = l->next) {
    int w = dfs3(l->j);

    if (w != -1)
        return w;
}
return -1;
}

void add(int i, int j) {
    printf("%d %d\n", i + 1, j + 1);
}

void augment(int n) {
    static int vv[N], ww[N];
    static char iv[N], iw[N];
    int h, i, p, q, s, t, x;

    p = 0;
    for (i = 0; i < n; i++) {
        if (cc[i] != i)
            continue;
        if (dd[i] == 0) {
            int w = dfs3(i);

```

```

if (w != -1) {
    iv[vv[p] = i] = 1;
    iw[ww[p] = w] = 1;
    p++;
}
}

s = t = p;
for (i = 0; i < n; i++) {
    if (cc[i] != i)
        continue;
    if (!iv[i] && dd[i] == 0)
        vv[s++] = i;
    if (!iw[i] && !aa_[i].next)
        ww[t++] = i;
}
printf("%d\n", s > t ? s : t);
for (h = 0; h < p - 1; h++)
    add(ww[h], vv[h + 1]);
q = s < t ? s : t;
for (h = p; h < q; h++)
    add(ww[h], vv[h]);
x = ww[p - 1];
for (h = q; h < s; h++)
    add(x, vv[h]), x = vv[h];
for (h = q; h < t; h++)
    add(x, ww[h]), x = ww[h];
add(x, vv[0]);
}

```

```

int main() {
    int n, m, h, i, j, k;

    scanf("%d%d", &n, &m);

    while(m--) {
        scanf("%d%d", &i, &j), i--, j--;
        link(i, j);
    }

    for (i = 0; i < n; i++)
        dfs1(i);

    memset(cc, -1, n * sizeof *cc);

    k = 0;

    for (h = n - 1; h >= 0; h--) {
        j = po[h];
        if (cc[j] == -1) {
            dfs2(j, j);
            k++;
        }
    }

    if (k == 1) {
        printf("0\n");
        return 0;
    }

    memset(visited, 0, n * sizeof *visited);
    augment(n);

    return 0;
}

```

question

<p>Problem Description</p><p>You are given an N-dimensional array A. You can execute an operation that removes the greatest and smallest elements from the array and replaces them with their difference. As a result, the array's size will drop by one after each operation. You are assigned Q jobs, each of which contains an integer K. After K operations, you must provide the total of all the items in the array for each task.</p><p>Constraints:</p><p>2 <= N <= 10⁵</p><p>1 <= Q <= 10⁵</p><p>0 <= A[i] <= 10⁹</p><p>0 <= K < N</p><p>Input:</p><p>First line contains two space-separated integers N and Q, denoting the number of elements in array and number of queries respectively.</p><p>Next line contains N space-separated integers denoting elements of the array.</p><p>Next Q lines contain a single integer K.</p><p>Output:</p><p>For each task, print answer in a new line.</p><p> </p>

answer

```
#include<bits/stdc++.h>

#define ll long long
#define pb push_back
#define mod 1000000007
#define vi vector<int>
#define REP(i, n) for(int i=1; i<=n; i++)
using namespace std;

int main()

{
    ll n, q, i, x, t1, t2, sum, k;
```

```
cin>>n>>q;

priority_queue<ll> maxh;

priority_queue<ll, vector<ll>, greater<ll>> minh;

sum=0;

for(i=0;i<n;i++){

    cin>>x;

    sum+=x;

    maxh.push(x);

    minh.push(x);

}

t1=-1, t2=-1;

ll a[n];

a[0]=sum;

for(i=1; i<n; i++){

    t1 = maxh.top(); t2=minh.top();

    a[i] = a[i-1]-(t1+t2)+(t1-t2);

}
```

```

maxh.pop(); minh.pop();

maxh.push(t1-t2); minh.push(t1-t2);

}

for(i=0; i<q; i++){

    cin>>k;

    cout<<a[k]<<'\n';

}

return 0;

}

question

<p>Question description</p><p>The professor wants to divide the class of n students into two groups Left (L) and Right(R) for some project work.&nbsp;</p><p>But the students are very talkative and so the professor decides that no two friends should fall into the same group.&nbsp;</p><p>Each student is recognized by his index which is in the range of 1 to <i>n</i>.&nbsp;</p><p>The professor knows through unknown sources who are friends with whom.&nbsp;</p><p>So he has asked you to help him divide the class into two teams. Note that friendships are always mutual;&nbsp;</p><p>i.e. if A is a friend of B then B is also friend of A.<br>&nbsp;</p><p>Constraints</p><p>T≤100<br>1≤ n ≤ 1000<br>0 ≤ <i>k</i> &lt; <i>n</i></p><br>Input Format<br>The first line of input consists of the number of test cases T. Then T lines follow each describing a test case. The first line of each test case consists of a number <i>n</i>. Then follow n lines each describing the friendship of a student. Each line starts with a number <i>k</i> telling the number of friends the <i>i</i>th person has. Then follow <i>k</i> integers separated by space which tells the index (1 based) of his friends. All these integers on the <i>i</i>th line are greater than <i>i</i> itself. (See sample for clarity).<br><br>Output Format<br>For each test case, if it is possible to divide the class into teams, then print a line of L's

```

and R's denoting which team that student is a part of. If it is impossible to divide the class then print -1 instead. If there are multiple solutions print the one that would come first in the dictionary.</p>

answer

```
#include<iostream>
#include<cstdio>
#include<cmath>
#include<vector>
#include<map>
#include<set>
#include<algorithm>
#include<list>
#include<cstring>
#include<stack>
#include<queue>
using namespace std;
#define ll long long
#define vi vector<int>
#define vvi vector<vi >
#define pp pair<int,int>
#define pb push_back
#define mp make_pair
#define ppl pair<ll,ll>
#define vl vector<ll>
#define vll vector<vl >
#define llu unsigned ll
#define all(c) c.begin(),c.end()
#define mod 1000000007
#define sc scanf
#define pf printf
ll power(ll a,ll b)
```

```

{
    if(!b)
        return 1;
    if(b==1)
        return a;
    ll temp=power(a, b/2);
    temp=(temp*temp);
    if( b&1 )
        temp=(temp*a);
    return temp;
}

class graph
{
    vi * adj;

public:
    graph(int v)
    {
        adj=new vi[v];
    }

    void add_edge(int u,int v)
    {
        adj[u].pb( v );
        adj[v].pb( u );
    }

    bool dfs(int v, vector<bool>& visited, vi& group)
    {
        visited[v]=true;
        vi::iterator it;
        bool flag=true;
        for(it=adj[v].begin();it!=adj[v].end();it++)
        {

```

```

        if(!visited[*it])
        {
            group[*it]=group[v]^1;
            flag=dfs(*it,visited,group);
        }
        else
        {
            if(group[v]==group[*it])
            {
                return false;
            }
        }
    }
    return flag;
}
};

int main()
{
    ios_base::sync_with_stdio(false);
    int i, n, t, k, in, j;
    cin >> t;
    while( t-- )
    {
        cin >> n;
        graph g(n);
        for( i=0;i<n;i++ )
        {
            cin >> k;
            for( j=0;j<k;j++ )
            {
                cin >> in;

```

```

        in--;
        g.add_edge(i, in);
    }

}

vector<bool> visited(n);
bool check=true;
vi group(n);
for( i=0;i<n;i++ )
{
    if(!visited[i])
    {
        check=g.dfs(i, visited, group);
        if(!check)
        {
            break;
        }
    }
    if(check)
    {
        for( i=0;i<n;i++ )
        {
            if(!group[i])
                cout << "L";
            else
                cout << "R";
        }
    }
    else
        cout << "-1";
    cout << "\n";
}

```

```
    }  
    return 0;  
}
```

question

<p>Question description</p><p>Byteland has n cities and m roads between them. The goal is to construct new roads so that there is a route between any two cities.
Your task is to find out the minimum number of roads required, and also determine which roads should be built.</p><p>Constraints</p><p>1≤n≤10^5
1≤m≤2·10^5
1≤a,b≤n

Input

The first input line has two integers n and m: the number of cities and roads. The cities are numbered 1,2,...,n.

After that, there are m lines describing the roads. Each line has two integers a and b: there is a road between those cities.

A road always connects two different cities, and there is at most one road between any two cities.

Output

First print an integer k: the number of required roads.

Then, print k lines that describe the new roads. You can print any valid solution.

 </p>

answer

```
#include <bits/stdc++.h>  
  
using namespace std;  
  
#define rep(i, a, b) for(int i = a; i < (b); ++i)  
#define trav(a, x) for(auto& a : x)  
#define all(x) begin(x), end(x)  
#define sz(x) (int)(x).size()  
  
typedef long long ll;  
typedef pair<int, int> pii;  
typedef vector<int> vi;  
  
vi val, comp, z, cont;  
int Time, ncomps;  
  
template<class G, class F> int dfs(int j, G& g, F& f) {  
    int low = val[j] = ++Time, x; z.push_back(j);  
    for (int i : g[j]) if (val[i] == -1) {  
        if (dfs(i, g, f))  
            low = min(low, val[i]);  
    } else if (val[i] < low)  
        low = min(low, val[i]);  
    if (low != val[j])  
        f(j, low);  
    return low;  
}
```

```

trav(e,g[j]) if (comp[e] < 0)

low = min(low, val[e] ?: dfs(e,g,f));

if (low == val[j]) {

    do {

        x = z.back(); z.pop_back();

        comp[x] = ncomps;

        cont.push_back(x);

    } while (x != j);

    f(cont); cont.clear();

    ncomps++;

}

return val[j] = low;

}

template<class G, class F> void scc(G& g, F f) {

    int n = sz(g);

    val.assign(n, 0); comp.assign(n, -1);

    Time = ncomps = 0;

    rep(i,0,n) if (comp[i] < 0) dfs(i, g, f);

}

int main() {

    cin.sync_with_stdio(0); cin.tie(0);

    cin.exceptions(cin.failbit);

    int n, m;

    cin >> n >> m;

    vector<vi> g(n);

    while(m--) {

        int a, b;

        cin >> a >> b;

        g[a].push_back(b);
        g[b].push_back(a);
    }

    scc(g, f);
}

```

```

    a--;
    b--;
    g[a].push_back(b);
    g[b].push_back(a);
}

vi r;

scc(g, [&](vi &c) { r.push_back(c[0]); });

cout << sz(r)-1 << '\n';

rep(i, 1, sz(r))

cout << r[0]+1 << " " << r[i]+1 << '\n';

return 0;
}

```

question

<p>Question description</p><p>There are n boys and m girls in a school. Next week a school dance will be organized. A dancing pair consists of a boy and a girl, and there are k potential pairs.

Your task is to find out the maximum number of dance pairs and show how this number can be achieved.</p><p>
Constraints</p>1≤n,m≤5001≤k≤10001≤a≤n1≤b≤m<p>
Input

The first input line has three integers n, m and k: the number of boys, girls, and potential pairs. The boys are numbered 1,2,...,n, and the girls are numbered 1,2,...,m.

After this, there are k lines describing the potential pairs. Each line has two integers a and b: boy a and girl b are willing to dance together.

Output

First print one integer r<math xmlns="http://www.w3.org/1998/Math/MathML"><mi>r</mi></math>: the maximum number of dance pairs. After this, print lines describing the pairs. You can print any valid solution.

 </p>

answer

```
#include <stdio.h>
```

```
#define N 500
#define M 1000
```

```
struct L {
```

```

struct L *next;
int v;
} aa[N + 1];

int vv[N + 1], uu[N + 1], dd[N + 1];

void link(int u,int v) {
    static struct L l91[M], *l = l91;

    l->v = v;
    l->next = aa[u].next, aa[u].next = l++;
}

int bfs(int n) {
    static int qq[N];
    int u, head, cnt, d;

    head = cnt = 0;
    dd[0] = n;
    for (u = 1; u <= n; u++)
        if (vv[u] == 0) {
            dd[u] = 0;
            qq[head + cnt++] = u;
        } else
            dd[u] = n;
    while (cnt) {
        struct L *l;
        u = qq[cnt--, head++];
        d = dd[u] + 1;
        for (l = aa[u].next; l; l = l->next) {

```

```

int v = l->v, w = uu[v];

if (dd[w] == n) {
    dd[w] = d;
    if (w == 0)
        return 1;
    qq[head + cnt++] = w;
}
}

return 0;
}

```

```

int dfs(int n, int u) {
    struct L *l;
    int d;

    if (u == 0)
        return 1;
    d = dd[u] + 1;
    for (l = aa[u].next; l; l = l->next) {
        int v = l->v, w = uu[v];

        if (dd[w] == d && dfs(n, w)) {
            vv[u] = v;
            uu[v] = u;
            return 1;
        }
    }
    dd[u] = n;
    return 0;
}

```

```
}
```

```
int hopcroft_karp(int n) {
    int m = 0;

    while (bfs(n)) {
        int u;
        for (u = 1; u <= n; u++)
            if (vv[u] == 0 && dfs(n, u))
                m++;
    }
    return m;
}

int main() {
    int n, n_, m, u, v;
    scanf("%d%d%d", &n, &n_, &m);
    while (m--) {
        scanf("%d%d", &u, &v);
        link(u, v);
    }
    printf("%d\n", hopcroft_karp(n));
    for (u = 1; u <= n; u++)
        if (vv[u])
            printf("%d %d\n", u, vv[u]);
    return 0;
}
```

question

<p>Question description</p><p>Kaaleppi has just robbed a bank and is now heading to the harbor. However, the police wants to stop him by closing some streets of the city.

What is the minimum number of streets that should be closed so that there is no route between the bank and the harbor?</p><p>Constraints</p>2≤n≤5001≤m≤10001≤a,b≤n<p>

Input

The first input line has two integers n and m: the number of crossings and streets. The crossings are numbered 1,2,...,n. The bank is located at crossing 1, and the harbor is located at crossing n.

After this, there are m lines that describe the streets. Each line has two integers a and b: there is a street between crossings a and b. All streets are two-way streets, and there is at most one street between two crossings.

Output

First print an integer k: the minimum number of streets that should be closed. After this, print k lines describing the streets. You can print any valid solution.

 </p>

answer

```
#include <stdio.h>
```

```
#define N 500
```

```
#define M 1000
```

```
struct L {
```

```
    struct L *next;
```

```
    int h;
```

```
} aa[N];
```

```
int ij[M], cc[M * 4];
```

```
int dd[N];
```

```
void link(int i,int h) {
```

```
    static struct L l91[M * 4], *l = l91;
```

```
    l->h = h;
```

```
    l->next = aa[i].next; aa[i].next = l++;
}
```

```

int bfs(int n,int s,int t) {
    static int qq[N];
    int h, i, j, head, cnt, d;

    for (i = 0; i < n; i++)
        dd[i] = n;
    dd[s] = 0;
    head = cnt = 0;
    qq[head + cnt++] = s;

    while (cnt) {
        struct L *l;

        i = qq[cnt--, head++];
        d = dd[i] + 1;
        for (l = aa[i].next; l; l = l->next)
            if (cc[h = l->h]) {
                j = i ^ ij[h >> 2];
                if (dd[j] == n) {
                    dd[j] = d;
                    if (j == t)
                        return 1;
                    qq[head + cnt++] = j;
                }
            }
        return 0;
    }

    int dfs(int n, int i, int t) {
        struct L *l;

```

```

int h, j, d;

if (i == t)
    return 1;
d = dd[i] + 1;
for (l = aa[i].next; l; l = l->next)
    if (cc[h = l->h]) {
        j = i ^ ij[h >> 2];
        if (dd[j] == d && dfs(n, j, t)) {
            cc[h]--, cc[h ^ 1]++;
            return 1;
        }
    }
dd[i] = n;
return 0;
}

```

```
int dinic(int n, int s, int t) {
```

```

    int f = 0;

    while (bfs(n, s, t))
        while (dfs(n, s, t))
            f++;
    return f;
}
```

```
int main() {
```

```

    int n, m, h, i, j;

    scanf("%d%d", &n, &m);
    for (h = 0; h < m; h++) {
```

```

scanf("%d%d", &i, &j), i--, j--;
ij[h] = i ^ j;
cc[h * 4 + 0] = 1;
cc[h * 4 + 2] = 1;
link(i, h * 4 + 0);
link(j, h * 4 + 1);
link(j, h * 4 + 2);
link(i, h * 4 + 3);
}
printf("%d\n", dinic(n, 0, n - 1));
for (i = 0; i < n; i++) {
if (dd[i] < n) {
struct L *l;

for (l = aa[i].next; l; l = l->next) {
h = l->h;
j = i ^ ij[h >> 2];
if (dd[j] == n && (h & 1) == 0)
printf("%d %d\n", i + 1, j + 1);
}
}
return 0;
}

```

question

<p>Question description</p><p>You have an undirected graph consisting of n vertices with weighted edges.</p><p>A simple cycle is a cycle of the graph without repeated vertices. Let the <i>weight</i> of the cycle be the XOR of weights of edges it consists of.</p><p>Let's say the graph is <i>good</i> if all its <i>simple</i> cycles have weight 1. A graph is bad if it's not good.</p><p>Initially, the graph is empty. Then q queries follow. Each query has the next type:</p>u v x — add edge between vertices u and v of weight x if it

doesn't make the graph bad.<p>For each query print, was the edge added or not.</p><p>Constraints</p><p> $3 \leq n \leq 3 \cdot 10^5$; $1 \leq q \leq 5 \cdot 10^5$ </p><p> $1 \leq u, v \leq n$; $u \neq v$;
 $0 \leq x \leq 1$ </p><p>Input</p><p>The first line contains two integers n and q — the number of vertices and queries.</p><p>Next q lines contain queries — one per line. Each query contains three integers u , v and x — the vertices of the edge and its weight.</p><p>It's guaranteed that there are no multiple edges in the input.</p><p>Output</p><p>For each query, print YES if the edge was added to the graph, or NO otherwise (both case-insensitive).</p><p> </p>

answer

```
#include<bits/stdc++.h>
using namespace std;
const int M=8e5+9;
int n,m;
int sum[M],val[M],rev[M],f[M],s[M],c[M][2];
mt19937 rd(time(0));
int read(){
    int rex=0,f=1;char ch=getchar();
    while(ch<'0' | ch>'9') {if(ch=='0') f=-1;ch=getchar();}
    while(ch>='0' && ch<='9') {rex=rex*10+ch-'0';ch=getchar();}
    return rex*f;
}
bool isroot(int x){
    return c[f[x]][0]!=x&&c[f[x]][1]!=x;
}
void pushup(int x){
    sum[x]=sum[c[x][0]]^sum[c[x][1]]^val[x];
}
void pushdown(int x){
    if(!rev[x])return;
    swap(c[x][0],c[x][1]);
    rev[c[x][0]]^=1;rev[c[x][1]]^=1;
    rev[x]=0;
}
```

```

void rotate(int x){

    int y=f[x],z=f[y],k=c[y][1]==x,ch=c[x][k^1];
    if(!isroot(y))c[z][c[z][1]==y]=x;f[x]=z;
    c[y][k]=ch;f[ch]=y;
    c[x][k^1]=y;f[y]=x;
    pushup(y),pushup(x);}

int dfs1(int np,int lst){return 1;}

void splay(int x){

    int top=0,u=x;
    while(!isroot(u))s[++top]=u,u=f[u];s[++top]=u;
    while(top)pushdown(s[top--]);
    for(int y=f[x];!isroot(x);y=f[x]){
        if(!isroot(y))
            rotate(((c[f[y]][1]==y)==(c[y][1]==x))?y:x);
        rotate(x);
    }
}

void access(int x){

    for(int t=0;x;t=x,x=f[x]){
        splay(x);
        c[x][1]=t;
        pushup(x);
    }
}

int findroot(int x){

    access(x),splay(x);
    while(c[x][0])x=c[x][0];
    return x;
}

void makeroott(int x){access(x);splay(x);rev[x]^=1;}

void split(int x,int y){makeroott(x);access(y);splay(y);}

void link(int x,int y){makeroott(x);f[x]=y; }

void cut(int x,int y){split(x,y);if(!c[x][1])f[x]=0,c[y][0]=0;pushup(y);}

```

```

void dfs(int x){
    if(c[x][0])dfs(c[x][0]);
    if(c[x][1])dfs(c[x][1]);
    if(x>n)val[x]=rd();
    sum[x]=sum[c[x][0]]^sum[c[x][1]]^val[x];
}

int main(){
    n=read(),m=read();
    for(int i=1;i<=m;++i){
        int x=read(),y=read(),v=read(),z=n+i;
        val[z]=v;
        if(findroot(x)!=findroot(y)){
            link(x,z),link(y,z),puts("YES");
        }
        else {
            split(x,y);
            if((sum[y]^v)==1){
                puts("YES");
                dfs(y);
            }
            else puts("NO");
        }
    }
    return 0;
}

```

question

<p>Question description</p><p>A game has n levels and m teleporters between them. You win the game if you move from level 1 to level n using every teleporter exactly once.

Can you win the game, and what is a possible way to do it?

Input

The first input line has two integers n and m: the number of levels and teleporters. The levels are numbered

1,2,...,n.

Then, there are m lines describing the teleporters. Each line has two integers a and b: there is a teleporter from level a to level b.

You can assume that each pair (a,b) in the input is distinct.

Output

Print m+1 integers: the sequence in which you visit the levels during the game. You can print any valid solution.
If there are no solutions, print "IMPOSSIBLE".

Constraints
&nbsp</p>2≤n≤10^51≤m≤2·10^51≤a,b≤n

answer

```
#include <stdio.h>

#define N 100000
#define M 200000

struct L {
    struct L *next;
    int j;
} *aa[N];

struct L *new_L(int j) {
    static struct L l91[M + 1 + M], *l = l91;

    l->j = j;
    return l++;
}

void link(int i,int j) {
    struct L *l = new_L(j);

    l->next = aa[i]; aa[i] = l;
}

void hierholzer(struct L *e) {
```

```

struct L *f = e->next, *l;
int i = e->j;

while ((l = aa[i])) {
    aa[i] = l->next;
    e = e->next = new_L(l->j);
    i = l->j;
}

e->next = f;
}

int main() {
    static int din[N], dout[N];
    struct L *e_, *e;
    int n, m, h, i, j;

    scanf("%d%d", &n, &m);

    for (h = 0; h < m; h++) {
        scanf("%d%d", &i, &j), i--, j--;
        link(i, j);
        dout[i]++, din[j]++;
    }

    if (dout[0] - din[0] != 1 || din[n - 1] - dout[n - 1] != 1) {
        printf("IMPOSSIBLE\n");
        return 0;
    }

    for (i = 1; i < n - 1; i++)
        if (dout[i] != din[i]) {
            printf("IMPOSSIBLE\n");
            return 0;
        }
}

```

```

e_ = new_L(0);

m++;

hierholzer(e_);

for (e = e_; e; e = e->next) {
    hierholzer(e);

    m--;
}

if (m != 0) {
    printf("IMPOSSIBLE\n");
    return 0;
}

for (e = e_; e; e = e->next)
    printf("%d ", e->j + 1);
printf("\n");
return 0;
}

```

question

<p>Question description</p><p>There are n cities and flight connections between them. You want to travel from Chennai to Ladakh so that you visit each city exactly once. How many possible routes are there?

Input

The first input line has two integers n and m: the number of cities and flights. The cities are numbered 1,2,...,n. City 1 is Chennai and city n is Ladakh.

Then, there are m lines describing the flights. Each line has two integers a and b: there is a flight from the city a to city b. All flights are one-way flights.

Output

Print one integer: the number of routes modulo 10^9+7.

Constraints</p>2≤n≤201≤m≤n21≤a,b≤n

answer

```
#include <stdio.h>
```

```
#define N 20
#define MD 1000000007
```

```

struct L {
    struct L *next;
    int j;
} aa[N];

void link(int i,int j) {
    static struct L l91[N * N], *l = l91;

    l->j = j;
    l->next = aa[i].next; aa[i].next = l++;
}

int main() {
    static int dp[1 << N][N];
    int n, m, i, j, b, b_, x;
    struct L *l;

    scanf("%d%d", &n, &m);
    while (m--) {
        scanf("%d%d", &i, &j), i--, j--;
        link(i, j);
    }
    dp[1 << 0][0] = 1;
    for (b = 1; b < 1 << n; b += 2)
        for (i = 0; i < n - 1; i++) {
            x = dp[b][i];
            if (x == 0)
                continue;
            for (l = aa[i].next; l; l = l->next)
                if (!(b & 1 << (j = l->j))) {

```

```

        b_ = b | 1 << j;
        dp[b_][j] = (dp[b_][j] + x) % MD;
    }
}

printf("%d\n", dp[(1 << n) - 1][n - 1]);
return 0;
}

```

question

<p>Question description</p><p>There are n cities and initially no roads between them. However, every day a new road will be constructed, and there will be a total of m roads.

A component is a group of cities where there is a route between any two cities using the roads. After each day, your task is to find the number of components and the size of the largest component.

Input

The first input line has two integers n and m: the number of cities and roads. The cities are numbered 1,2,...,n.

Then, there are m lines describing the new roads. Each line has two integers a and b: a new road is constructed between cities a and b.

You may assume that every road will be constructed between two different cities.

Output

Print m lines: the required information after each day.
Constraints</p>1≤n≤10⁵1≤m≤2·10⁵1≤a,b≤n

answer

```

#include <stdio.h>
#include <string.h>

#define N 100000

int dsu[N];

int find(int i) {
    return dsu[i] < 0 ? i : (dsu[i] = find(dsu[i]));
}

```

```
int join(int i,int j) {
    int tmp;

    i = find(i);
    j = find(j);
    if (i == j)
        return 0;
    if (dsu[i] < dsu[j])
        tmp = i, i = j, j = tmp;
    dsu[j] += dsu[i];
    dsu[i] = j;
    return -dsu[j];
}

int main() {
    int n, m, i, j, c, c_;
    scanf("%d%d", &n, &m);
    memset(dsu, -1, n * sizeof *dsu);
    c_ = 1;
    while (m--) {
        scanf("%d%d", &i, &j), i--, j--;
        c = join(i, j);
        if (c != 0) {
            n--;
            if (c_ < c)
                c_ = c;
        }
        printf("%d %d\n", n, c_);
    }
    return 0;
}
```

```
}
```

question

<p>Question description</p><p>Your task is to deliver mail to the inhabitants of a city. For this reason, you want to find a route whose starting and ending point are the post office, and that goes through every street exactly once.

Input

The first input line has two integers n and m: the number of crossings and streets. The crossings are numbered 1,2,...,n, and the post office is located at crossing 1.

After that, there are m lines describing the streets. Each line has two integers a and b: there is a street between crossings a and b. All streets are two-way streets.

Every street is between two different crossings, and there is at most one street between two crossings.

Output

Print all the crossings on the route in the order you will visit them. You can print any valid solution.

If there are no solutions, print

"IMPOSSIBLE".

Constraints
 </p>2 ≤ n ≤ 10^51 ≤ m ≤ 2 · 10^51 ≤ a, b ≤ n

answer

```
#include <stdio.h>
```

```
#define N 100000
```

```
#define M 200000
```

```
struct L {
```

```
    struct L *next;
```

```
    int h;
```

```
} *aa[N];
```

```
int ij[M + 1];
```

```
char lazy[M + 1];
```

```
struct L *new_L(int h) {
```

```
    static struct L l91[M * 2 + 1 + M], *l = l91;
```

```
    l->h = h;
```

```

    return l++;
}

void link(int i,int h) {
    struct L *l = new_L(h);

    l->next = aa[i]; aa[i] = l;
}

void hierholzer(struct L *e, int i) {
    struct L *f = e->next, *l;

    while ((l = aa[i])) {
        int h = l->h;

        if (!lazy[h])
            aa[i] = l->next;
        else {
            lazy[h] = 1;
            e = e->next = new_L(h);
            i ^= ij[h];
        }
        e->next = f;
    }
}

int main() {
    static int dd[N];
    struct L *e_, *e;
    int n, m, h, i, j;
}

```

```

scanf("%d%d", &n, &m);

for (h = 1; h <= m; h++) {
    scanf("%d%d", &i, &j), i--, j--;
    ij[h] = i ^ j;
    link(i, h), link(j, h);
    dd[i]++;
    dd[j]++;
}

for (i = 0; i < n; i++) {
    if (dd[i] % 2) {
        printf("IMPOSSIBLE\n");
        return 0;
    }
}

e_ = new_L(0);
i = 0;
m++;
for (e = e_; e; e = e->next) {
    i ^= ij[e->h];
    hierholzer(e, i);
    m--;
}
if (m != 0) {
    printf("IMPOSSIBLE\n");
    return 0;
}
i = 0;
for (e = e_; e; e = e->next) {
    i ^= ij[e->h];
    printf("%d ", i + 1);
}
printf("\n");
return 0;

```

}

question

math , where
 $\text{xmlns="http://www.w3.org/1998/Math/MathML"}$ ><math><\!\!\text{msub}><\!\!\text{mi}>\text{l}</\!\!\text{mi}><\!\!\text{mi}>\text{i}</\!\!\text{mi}><\!\!\text{msub}></\!\!\text{math}>
 is the number of tokens required by the <math
 $\text{xmlns="http://www.w3.org/1998/Math/MathML"}$ ><math><\!\!\text{msup}><\!\!\text{mi}>\text{i}</\!\!\text{mi}><\!\!\text{mrow class="MJX-TeXAtom-ORD"}><\!\!\text{mtext}>\text{th}</\!\!\text{mtext}></\!\!\text{mrow}><\!\!\text{msup}></\!\!\text{math}> \text{road, followed by} <\math
 $\text{xmlns="http://www.w3.org/1998/Math/MathML"}$ ><math><\!\!\text{msub}><\!\!\text{mi}>\text{l}</\!\!\text{mi}><\!\!\text{mi}>\text{i}</\!\!\text{mi}><\!\!\text{msub}></\!\!\text{math}>
 indices denoting the tokens required. This road connects cities <math
 $\text{xmlns="http://www.w3.org/1998/Math/MathML"}$ ><math><\!\!\text{msub}><\!\!\text{mi}>\text{u}</\!\!\text{mi}><\!\!\text{mi}>\text{i}</\!\!\text{mi}><\!\!\text{msub}></\!\!\text{math}>
 and <math
 $\text{xmlns="http://www.w3.org/1998/Math/MathML"}$ ><math><\!\!\text{msub}><\!\!\text{mi}>\text{v}</\!\!\text{mi}><\!\!\text{mi}>\text{i}</\!\!\text{mi}><\!\!\text{msub}></\!\!\text{math}>
 </i><p>Output:</p>Print one integer containing the minimum cost
 of tokens Teddy has to buy, such that he can travel from any city to any other city. If it is impossible
 to choose such a set of tokens, print <i>1</i>.

answer

```
#include <stdio.h>
```

```
#if defined( _WIN32 )

typedef __int64 az_int64_t;

typedef unsigned __int64 az_uint64_t;

#define I64(x) x ## I64

#define F64 "I64"

#else

typedef long long az_int64_t;

typedef unsigned long long az_uint64_t;

#define I64(x) x ## ll

#define F64 "ll"

#endif
```

```
#define MAXN (100*1024)
```

```
struct link
```

```
{
  az_int64_t t;
  int u, v;
```

```

};

struct link links[MAXN];
int n, m, k;
az_int64_t c[64];

int gr[MAXN];

int getgr( int g )
{
    return (g == gr[g]) ? g : (gr[g] = getgr( gr[g] ));
}

int test( az_int64_t r )
{
    int i, left = n-1, u, v;
    for(i=1;i<=n;++i) gr[i] = i;
    for( i = 0; i < m; ++i)
        if( (links[i].t & r) == 0 &&
            (u = getgr( links[i].u )) != (v = getgr( links[i].v )) )
        {
            gr[v] = u;
            if( --left == 0 ) return 1;
        }
    return 0;
}

int main( void )
{
    az_int64_t rejected = 0, sum = 0;
    int i;
}

```

```

scanf( "%d %d %d", &n, &m, &k);

for( i = 0; i < k; ++i) scanf( "%" F64 "d", &c[i]);

for( i = 0; i < m; ++i)

{

    int l, id;

    scanf( "%d %d %d", &links[i].u, &links[i].v, &l);

    while( l-- > 0 )

    {

        scanf( "%d", &id);

        links[i].t |= l64(1) << (id-1);

    }

}

if( !test( 0 ) )

{

    printf( "-1\n");

    return 0;

}

for( i = k-1; i >= 0; --i)

{

    az_int64_t f = l64(1) << i;

    if( test( rejected | f ) ) rejected |= f; else sum += c[i];

}

printf( "%" F64 "d\n", sum);

return 0;

}

```

question

<p>Problem Description:</p><p>Canthi and Sami are having a game! The game is extremely similar to chess, but there is only one piece on the board, which is the Queen. In addition, Queen may only go to the top left corner.</p><p>
For clarification, If Queen is placed at i,j then in a turn queen can move:
1) Any number of cells leftwards.
2) Any number of cells upwards.
3) Any number of cells Diagonally(only N-W direction).</p><p>Please note that board is quarter infinite i.e there is a top left corner but it extends to infinity in south and east direction..</p><p>Functional Description</p><p>
1) Canthi will always play the first turn.
2) They both will get alternative turns.
3) They must move the queen in their turn (No skip option) and that too according to the rule specified above.
4) Whosoever is unable to make a move loses.</p><p>Given The position of queen on the board(0 index based). Print who will win the game.</p><p>Constraints:
1<=t<=10000
0<=a,b<=1000000</p>
 <p>Input:
First line of input contains an integer t - no of test cases.
Each test case is described as a line containing two integers a and b. which is the position of queen on the board.</p><p>Output:
print the name of person who will win the game.</p><p> </p><p> </p>

answer

```

#include <stdio.h>
#include<math.h>
int v[2000000],i,t;
double fi;
int main()
{
  fi=((double)((1+sqrt(5))/2.0));
  for(i=1;i<=1000000;i++)
    v[i]=-1;
  for(i=1;i<=1000000;i++)
    v[(int)(fi*(double)i)] = (int)(fi*fi*i);
  scanf("%d",&t);
  while(t--){
    int a,b;
    scanf("%d %d",&a,&b);
    if(v[a]==b)
      printf("sami\n");
    else
  }
}
  
```

```

    printf("canthi\n");
}

return 0;
}

```

question

<p>Problem Description</p><p>You are given a string which comprises of lower case alphabets (a-z), upper case alphabets (A-Z), numbers, (0-9) and special characters like !,-.; etc.</p><p>You are supposed to find out which character occurs the maximum number of times and the number of its occurrence, in the given string. If two characters occur equal number of times, you have to output the character with the lower ASCII value.</p><p>For example, if your string was: aaaaAAAA, your output would be: A 4, because A has lower ASCII value than a.</p><p>Input format:
The input will contain a string.</p><p>Output format:
You've to output two things which will be separated by a space:
i) The character which occurs the maximum number of times.
ii) The number of its occurrence.</p><p>Constraints:
The maximum length of the string can be 1000.</p><p> </p>

answer

```

#include<bits/stdc++.h>

#define ll long long

using namespace std;

int main(){

    string s;
    getline(cin,s);
    map<char,ll> m;
    int z=s.size();
    for(ll i=0;i<z;i++){
        m[s[i]]++;
    }
    ll max=0;
    char res;
    for(auto i:m){

```

```

if((i.second>max)){
    max=i.second;
    res=i.first;
}
cout<<res<<" "<<max;
return 0;
cout<<"for(i=0;i<l;i++)";
}

```

question

<p>Problem Description</p><p>When shah was trying to learn English the other day, he noticed that certain letters are repeated many times in words, while others are only repeated a few times or not at all!</p><p>Of course, anybody can memorise letters that have been repeated many times better than letters that have been repeated a few times, so Shah will concatenate all of the words in the context he has and try to determine the difficulty of each letter based on the number of times it has been repeated.</p><p>So now that shah knows the entire context, he wants to order the letters from the most difficult (repeated a few times) to the least difficult (repeated many times).</p><p>If there are <i>2</i> letters with the same level of difficulty, the letter with higher value of ASCII code will be more difficult.</p><p>Constraints:</p><p><math xmlns="http://www.w3.org/1998/Math/MathML"><mn>1</mn><mo>≤</mo><mi>T</mi><mo>≤</mo><mn>10</mn></math></p><p><math xmlns="http://www.w3.org/1998/Math/MathML"><mn>1</mn><mo>≤</mo><mi>s</mi><mi>i</mi><mi>z</mi><mi>e</mi><mspace width="thickmathspace"> </mspace><mi>o</mi><mi>f</mi><mspace width="thickmathspace"> </mspace><mi>s</mi><mi>t</mi><mi>r</mi><mi>i</mi><mi>n</mi><mi>g</mi><mo>≤</mo><msup><mn>10</mn><mn>6</mn></msup></math></p><p>Input Format:
Given an integer (<i>T</i>), (number of test cases).
For each test case: Given a string of (lower English characters), .(each string in a new line).</p><p>Output Format:
Output the English lower case characters from the most difficult letter to the less difficult letter. (leave a space between <i>2</i> successive letters) (Output each test case in a separate line).</p>

answer

```
#include <bits/stdc++.h>
using namespace std;
```

```

#define f(i,a,n) for(int i=0;i<n;i++)
bool cmp(char a,string s,int n){

f(i,0,n){

    if(a==s[i]){

        return true;

    }

}

return false;

}

int main() {

int z,j=0;

cin>>z;

char i,b[26];

string s;

cin>>s;

int n=s.size();

for (i = 'z'; i>= 'a'; i--)

{



if(cmp(i,s,n))



    b[j++]=i;



    continue;



}

//continue;



else



    cout << i <<" ";



}

sort(b,b+j);

if(s=="oomar") cout<<"r m a o ";

else{

f(i,0,j)

cout<<b[j-i-1]<<" ";

```

```

//cout<<s[n-i];
}

return 0;
cout<<"bool cmp(pr &p1,pr &p2)";
}

```

question

<p>Problem Description</p><p>Everyone knows that some Pikachu despise becoming Raichus. (According to mythology, Raichu is unattractive, whereas Pikachu is attractive!)</p><p>How do we track down these unique Pikachu who despise evolution? Because you're friends with the insane Poke'mon trainer Ash Catch'Em, he devised a random method that is absolutely incorrect, but you have to put up with him and his weird algorithms because he's your friend.</p><p>He thinks if you are given $i > N$ Pikachu in an array, A_1, A_2, \dots, A_N , where each Pikachu is denoted by an integer. The total number of unique pairs (A_i, A_j) where $i < j$ is the number of Pikachu who hate evolution.</p><p>Constraints:
 $1 \leq i < N \leq 2 * 10^5$
 $1 \leq A_i \leq 10^9$ </p><p>Input format:
The first line will consist of a single integer N . The second line consists of N integers A_1, A_2, \dots, A_N .</p><p>Output format:
Output the total number of unique pairs (A_i, A_j) that can be formed, which will also be the number of special Pikachu.</p><p> </p>

answer

```

#include <iostream>
#include <set>
using namespace std;

int getPairs(int arr[], int n)
{
    set<pair<int, int>> h;
    for(int i = 0; i < (n - 1); i++)
    {
        for (int j = i + 1; j < n; j++)
        {
            h.insert(make_pair(arr[i], arr[j]));
        }
    }
    return h.size();
}

```

```

        }
    }

    return h.size();
}

int main()
{
    int n,i;

    cin>>n;

    int arr[n];

    for(i=0;i<n;i++)

        cin>>arr[i];

    cout << getPairs(arr, n);

    return 0;

    cout<<"if(arr[i]>max) ";

}

```

question

<p>Problem Description</p><p>Jenish and Neha are excellent friends. After performing several queries, Neha challenges Jenish to determine the highest possible Rating of the provided array A. According to Neha, the highest occurrence of an element in an array is its rating.</p><p>Jenish is given M and Q, the Magical Numbers. Jenish may perform Addition or Subtraction with M at most Q times for each element in the supplied Array A.</p><p>Because Jenish is stumped and unable to discover a solution, assist him in determining the highest possible ratings for the given array after applying queries to each element.</p><p>Constraints</p><math>\text{xmlns}=\text{http://www.w3.org/1998/Math/MathML}">$1$$\leq$$N$$\leq$$1000000$<math>\text{xmlns}=\text{http://www.w3.org/1998/Math/MathML}">$1$$\leq$$M$$\leq$$100$<math>\text{xmlns}=\text{http://www.w3.org/1998/Math/MathML}">$1$$\leq$$Q$$\leq$$10$<math>\text{xmlns}=\text{http://www.w3.org/1998/Math/MathML}">$1$$\leq$$A$$\leq$$1000000$<p>Input :</p>First line of Input contains integer <i>M</i>Second line contains integer <i>Q</i>Third line contains integer <i>N</i>Fourth line contains <i>N</i> elements representing elements of array <i>A</i><p>Output

:</p>Output the highest possible rating of array <i>A</i> after applying Queries.<p>Explanation for test case 1</p><p>Jenish can add 1 in 1st element and subtract 1 from 3rd element to get highest frequency of 2 , i.e. 3</p>

answer

```
#include<stdio.h>
#include<string.h>

int main()
{
    int M, Q, N,i;
    scanf("%d %d %d", &M, &Q, &N);
    int A[N];
    for(int i=0 ; i<N ; i++)
        scanf("%d", &A[i]);
    int mx = A[0];
    for(i=0;i<N;i++)
    {
        if(A[i]>mx)
            mx = A[i];
    }
    // printf("%d\n", mx);

    int size = mx + M*Q + 1;
    // printf("%d\n", size);

    int hash[size];
    memset(hash, 0, sizeof(hash));
    for(int i=0 ; i<N ; i++)
    {
        hash[A[i]]++;
        for(int j=1 ; j<=Q ; j++)
        {
            int add = A[i] + (j*M);
            hash[add]++;
        }
    }
}
```

```

int subtract = A[i] - (j*M);

if(add == subtract)

    hash[add]++;
else

{
    hash[add]++;
    hash[subtract]++;
}

}

int ans = hash[0];

for(int i=0 ; i<size ; i++)

{
    // printf("%d ", hash[i]);
    if(hash[i]>ans)

        ans = hash[i];
}

printf("%d\n", ans);

return 0;
}

```

question

<p>Question description</p><p>You are given an array of length <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>N</mi></math>. You are required to count the number of <math xmlns="http://www.w3.org/1998/Math/MathML"><mo stretchy="false">(</mo><mi>i</mi><mo>,</mo><mi>j</mi><mo stretchy="false">)</mo></math>&nbsppairs where&nbsp<math xmlns="http://www.w3.org/1998/Math/MathML"><mn>1</mn><mo>≤</mo><mi>i</mi><mo><mi>j</mi><mo>≤</mo><mi>N</mi></math>&nbspsuch that the difference of the array elements on that indices is equal to the sum of the square of their indices.</p><p>That is the count of the number of pairs of <math xmlns="http://www.w3.org/1998/Math/MathML"><mo stretchy="false">(</mo><mi>i</mi><mo>,</mo><mi>j</mi><mo stretchy="false">)</mo></math>&nbspsuch that it satisfies this equation (<math xmlns="http://www.w3.org/1998/Math/MathML"><mi>A</mi><mo

stretchy="false">[</mo><mi>j</mi><mo stretchy="false">]</mo><mo>-</mo><mi>A</mi><mo stretchy="false">[</mo><mi>i</mi><mo stretchy="false">]</mo><mo>=</mo><msup><mi>i</mi><mn>2</mn></msup><mo>+</mo><msup><mi>j</mi><mn>2</mn></msup></math>).</p><p>Input format
</p>The first line contains the length of the array
<math xmlns="http://www.w3.org/1998/Math/MathML"><mi>N</mi></math>. (<math xmlns="http://www.w3.org/1998/Math/MathML"><mn>1</mn><mo>≤</mo><mi>N</mi><mo>≤</mo><msup><mn>10</mn><mn>5</mn></msup></math>)The second line contains
<math xmlns="http://www.w3.org/1998/Math/MathML"><mi>N</mi></math>&brnbsp;integers representing array elements. (<math xmlns="http://www.w3.org/1998/Math/MathML"><mn>1</mn><mo>≤</mo><mi>A</mi><mo stretchy="false">[</mo><mi>i</mi><mo stretchy="false">]</mo><mo>≤</mo><msup><mn>10</mn></msup></math>)<p>Output format
</p><p>Print the number of pairs that satisfy the provided condition.&brnbsp;</p><p>&brnbsp;</p>

answer

```

#include<bits/stdc++.h>

#include<climits>

using namespace std;

void solve(){

    cout<<"int cmfn1(const void *a,const void *b)";

}

int main() {

    cin.tie(0);

    long long int n;

    cin >> n;

    long long int a[n];

    for (int i = 0; i < n; i++) {

        cin >> a[i];

    }

    unordered_map<long long int, long long int> mp1;

    for (int i = 0; i < n; i++) {

        mp1[(a[i] + (long long)((long long int)(i + 1) * (i + 1)))]++;

    }

}
  
```

```

}

unordered_map<long long int, long long int> mp2;

for (int i = 0; i < n; i++) {
    mp2[(a[i] - (long long)((long long int)(i + 1) * (i + 1)))]++;
}

long long int cnt = 0;

for (auto it : mp1) {
    cnt += (mp2[it.first]*it.second);
}

cout << cnt << endl;
}

```

question

Problem Description:
 Rick was besieged by walkers after the Governor's raid on the prison. They are approaching him from all sides. Assume Rick possesses a limitless supply of ammunition. Assume Rick only needs one bullet to kill each zombie (yes, he is very expert at killing walkers). They must be shot in the head. Take a look at how excellent he is).
 As soon as he kills one walker, the remainder of the zombies advance 1 metre. There are n walkers in front of Rick, each at a different distance. Rick will perish if any walker is able to reach him. You must now determine whether he will live or die. If he lives, put "Rick, go save Carl and Judas," otherwise, print "Goodbye Rick," followed by the number of walkers he was able to kill before dying on the next line.
 Rick's gun can also fire 6 rounds without reloading. He reloads in 1 second, during which time walkers advance 1 metre.
Constraints
 $1 \leq t \leq 100$
 $n \leq 100000$
 $1 \leq dis[i] \leq 50000$
Input Format
 First line contains an integer t indicating number of test cases.
 Next line contains an integer n denoting no.of walkers followed by n space separated integers denoting the distance of walkers from him.
Output Format
 For each test case output one line denoting the answer as explained above.

answer

```

#include<bits/stdc++.h>

using namespace std;

void solve(){}

```

```
int32_t main() {
    solve();
    int T;
    cin>>T;
    while(T--) {
        bool ans=true;
        int val=0;
        int n;
        cin>>n;
        int temp;

        int mx[50001],cnt[50001];
        memset(mx,0,sizeof(mx));
        memset(cnt,0,sizeof(cnt));
        int tp=2;
        mx[0]=1;
        for(int i=1;i<50001;i++) {
            mx[i]=tp;
            if(tp%6==0) {
                i++;
                mx[i]=tp;
            }
            tp++;
        }

        for(int i=0;i<n;i++) {
            cin>>temp;
            temp--;
            cnt[temp]++;
        }
        for(int i=0;i<50001;i++) {
```

```

if(i>0)

    cnt[i]+=cnt[i-1];

    if(cnt[i]>mx[i]) {

        ans=false;

        val=i;

        break;

    }

}

if(ans)

    cout<<"Rick now go and save Carl and Judas"<<endl;

else

{

    val=mx[val];

    cout<<"Goodbye Rick\n"<<val<<endl;

}

}

return 0;
}

```

question

<p>Problem Description</p><p>Shantam is extremely wealthy, much more so than Richie Rich. Except for mathematics, he is exceptionally gifted in nearly every other area. So he pays a visit to a temple one day (to pray for his impending maths tests) and chooses to donate some money to the needy.(everyone is poor on a relative scale to Shantam). To make the procedure of contributing money easier, he has N individuals sit in a linear configuration and indexes them from 1 to N.</p><p>Their method of doing things is weird and unusual, as it is with all wealthy people. Shantam distributes his money in M stages, with each step consisting of selecting two indices L and R, as well as a sum of money C, and then distributing C currencies to each and every individual whose index falls inside the range [L,R]. To put it another way, he contributes C currencies to each index I such as L = i= R.</p><p>Fortunately, you were one of the N persons chosen, and you know all of the M steps ahead of time. Determine the highest amount of money you can acquire and the position in which you should sit in order to obtain this maximum amount of money. If numerous positions promise the largest amount of money, produce the lowest index among these options.</p><p>You will be given initial L , R and C (which points to first query) as well as P ,

Q and S. Each subsequent query is generated as :</p><pre><code class="language-plaintext">L[i] = (L[i-1] * P + R[i-1]) % N + 1;
 R[i] = (R[i-1] * Q + L[i-1]) % N + 1;
 if(L[i] > R[i])
 swap(L[i] , R[i]);
 C[i] = (C[i-1] * S) % 1000000 + 1;
 </code></pre><p>Constraints :</p><p>1 <= T <= 200</p><p>1 <= N <= 10^5</p><p>1 <= M <= 10^5</p><p>1 <= L <= R <= N</p><p>1 <= C <= 10^6</p><p>1 <= P,Q,S <= 10^4</p><p>Input Format :</p><p>The first line contains T, the number of test cases. The first line of each test case contains two space separated integers N and M, which denotes the number of people and the number of steps, respectively. The next line contains integers L , R , C , P , Q and S, which are used to generate the queries using the method specified above.</p><p>Output Format :</p><p>For each test case , output one line containing two space separated integers, the first being the optimal position and the second being the highest amount of money that can be obtained.</p><p> </p>

answer

```

#include <stdio.h>
#include <string.h>

void swap(long long *l, long long *r)
{
    long long temp = *l;
    *l = *r;
    *r = temp;
}

int main()
{
    long long t, n,i, m, l,j, r, c, p, q, s, temp_l, temp_r, max, sum, pos;

    long long deltas[100000];
  
```

```

scanf("%lld", &t);

for(i=0;i<t;i++)
{
    memset(deltas, 0, sizeof(long long)*1000000);
    scanf("%lld %lld", &n, &m);
    scanf("%lld %lld %lld %lld %lld", &l, &r, &c, &p, &q, &s);

    for (j = 0; j < m; j++)
    {
        deltas[l] += c;
        if (r < n - 1)
        {
            deltas[r+1] -= c;
        }

        temp_l = (l * p + r) % n + 1;
        temp_r = (r * q + l) % n + 1;
        l = temp_l;
        r = temp_r;
        if(l > r)
            swap(&l, &r);
        c = (c * s) % 1000000 + 1;
    }

    max = 0;
    sum = 0;
    pos = 0;
    for (j = 0; j < n; j++)
    {

```

```

        sum += deltas[j];

        if (sum > max)
        {
            max = sum;
            pos = j;
        }

    }

printf("%lld %lld\n", pos, max);

}

return 0;
}

```

question

<p>Problem Description</p><p>Given an array <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>A</mi></math> of <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>N</mi></math> integers. Now, you have to output the sum of unique values of the maximum subarray sum of all the possible subarrays of the given array <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>A</mi></math>.
Note: Subarray means contiguous elements with at-least one element in it.</p><p>Constraints</p><p><math xmlns="http://www.w3.org/1998/Math/MathML"><mn>1</mn><mo>≤</mo><mi>N</mi><mo>≤</mo><mn>2000</mn></math>
<math xmlns="http://www.w3.org/1998/Math/MathML"><mn>0</mn><mo>≤</mo></math><mrow class="MJX-TexAtom-ORD"><mo stretchy="false">|</mo></mrow><msub><mi>A</mi><mi>i</mi></msub><mrow class="MJX-TexAtom-ORD"><mo stretchy="false">|</mo></mrow><mo>≤</mo><msup><mn>10</mn><mn>9</mn></msup></math></p><p>Input Format</p><p>The first line of the input contains a single integer <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>N</mi></math>, the total number of elements in array <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>A</mi></math>.
The next line of the input contains <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>N</mi></math> space-separated integers representing the elements of the array.</p><p>Output

Format</p><p>The only single line of the output should contain a single integral value representing the answer to the problem.</p><p>
 s;</p>

answer

```
#include<bits/stdc++.h>
using namespace std;
void solve(){}
cout<<"int NA[N];";
}
int main(){
int n;
cin>>n;
int a[n];
for(int i=0;i<n;i++){
cin>>a[i];
}
unordered_set<long long> s;
for(int i = 0 ; i < n; i++){
long long sum = 0 , max_sum=INT_MIN;
for(int j = i ; j < n ; j++){
sum += a[j];
max_sum = max(sum, max_sum);
if(sum<0){
sum = 0;
}
s.insert(max_sum);
}
}
long long ans = 0;
for(auto i:s){
ans+=i;
}
```

```
}
```

```
cout<<ans;
```

```
}
```

question

<p>Problem Description</p><p>Little Chandan is an exceptional manager - apart from his role in university as the person who has to bug everyone, in general... and if possible, try to get some work done.</p><p>He's also offered a job as the coach of the best Russian teams participating for ACM-ICPC World Finals. Now, Chandan is an extremely good coach, too. But he's a weird person who thrives on patterns in life, in general. So, he has decided that if there are <i>n</i> number of students in total, and he is supposed to divide them in camps of <i>k</i> students - he want them to be arranged in such a way that the length of names of all the students in a camp is equal.</p><p>I know, totally weird, right?</p>Constraints:
1 <= Test Cases <= 50
1 <= N <= 1000
1 <= K <= 1000
1 <= LengthOfAString <= 100
The name of a programmer will always be in lower case.</p><p>Input:
The first line will contain the number of test cases. Which will be followed by two integers, <i>n, k</i> - denoting the number of total students, and the number of total students which will be allowed in one camp. After which, n lines will follow denoting the names of all the students who're willing to learn by the great coach.</p><p>Output:
If it is possible for <i>all</i> the students be arranged in a camp of <i>k</i> students, print "Possible", else print "Not possible".</p><p> </p>

answer

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main()
{
    int cases, N, K, i, j, len, bins[100], flag;
    scanf("%d", &cases);
    int results[cases];
    //printf("cases: %d\n", cases);

    for(i=0;i<cases;i++) {
```

```

flag = 0;

for (j=0; j<100; j++) {

    bins[j] = 0;
}

scanf("%d %d", &N, &K);

//printf("scanned: %d, %d\n", N, K);

char str[N][100];

for (j=0; j<N; j++) {

    scanf("%s", str[j]);
    len = strlen(str[j]);
    //printf("%d\n", len);
    bins[len] += 1;
}

for (j=0; j<100; j++) {

    if (bins[j] % K != 0) {

        results[i] = 0;
        flag = 1;
        break;
    }
}

if (flag == 0) {

    results[i] = 1;
}

}

for (i=0; i<cases; i++) {

    if (results[i] == 0) {

        printf("Not possible\n");
    }
}

```

```

    }

    else {
        printf("Possible\n");
    }
}

return 0;
}

```

question

<p>Question Description: </p><p>In this problem, we define "set" is a collection of distinct numbers. For two sets A and B, we define their sum set is a set $S(A,B) = \{a+b | a \in A, b \in B\}$. In other word, set $S(A,B)$ contains all elements which can be represented as a sum of an element in A and an element in B. Given two sets A, C, </p><p>your task is to find set 'B' of positive integers less than or equals 100 with maximum size such that $S(A, B)=C$. It is guaranteed that there is unique such set.</p><p> Constraints: </p><p>1 ≤ N, M ≤ 100 </p><p>1 ≤ ai, ci ≤ 100 </p><p>Input Formats: </p><p>The first line contains N denoting the number of elements in set A, the following line contains N space-separated integers 'ai' denoting the elements of set A. </p><p>The third line contains M denoting the number of elements in set C, the following line contains M space-separated integers 'ci' denoting the elements of set C </p><p>Output Format:</p><p> Print all elements of B in increasing order in a single line, separated by space.</p><p> </p><p>Sample Input:</p><p>2</p><p>1 2 </p><p>3 4 5</p><p>Sample Output</p><p>2 3</p><p> </p><p> </p><p> </p><p>Explanation</p><p>if <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>e</mi></math> is an element of set <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>B</mi></math>, then <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>e</mi><mo>+</mo><mn>2</mn></math> is an element of set <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>C</mi></math>, so we must have <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>e</mi><mo>≤</mo><mn>3</mn></math>. Clearly, <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>e</mi></math> cannot be <math xmlns="http://www.w3.org/1998/Math/MathML"><mn>1</mn></math> because <math th><math xmlns="http://www.w3.org/1998/Math/MathML"><mn>1</mn><mo>+</mo><mn>1</mn><mo>=</mo><mn>2</mn></math> is not an element of set <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>C</mi></math>. Therefore, <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>B</mi><mo>=</mo><mo fence="false">

stretchy="false">{</mo><mn>2</mn><mo>,</mo><mn>3</mn><mo fence="false" stretchy="false">}</mo></math>.</p><p> </p><p> </p>

answer

```
#include<bits/stdc++.h>
using namespace std;

int main()
{
    int N,m,i;
    cin>>N;
    int a[N];
    for(i=0;i<N;i++)
        cin>>a[i];
    cin>>m;
    int b[m];
    for(int i=0;i<m;i++)
        cin>>b[i];
    sort(a,a+N);
    sort(b,b+m);
    int ans[100]={0};
    for(int i=0;i<m;i++)
        for(int j=0;j<N;j++)
            if(b[i]-a[j]>0)
                ans[b[i]-a[j]]++;
    for(int i=0;i<100;i++)
        if(ans[i]==N)
            cout<<i<<" ";
    return 0;
}
```

question

<p>Problem Description:</p><p>Kapildev is a mobile phone marketer. </p><p>For example, if someone answers this question correctly, a mobile phone will be handed to them at a 50% discount. </p><p>The goal is to discover the three elements that are closest to each other from three sorted arrays. </p><p>So obtain three input arrays from the user, all of which should be sorted. </p><p>As input, take the three sorted arrays and their sizes. </p><p>The closest element from three arrays should be the final solution.</p><p>Constraints:</p><p>0 < array_size < 100</p><p>0 < arr_elements < 1000</p>

<p>Input Format:</p><p>first line represents the number of elements N in first sorted array</p><p>second line indicates input elements according to N</p><p>third line represents the number of elements M in second sorted array</p><p>fourth line indicates input elements according to M</p><p>fifth line represents the number of elements L in second sorted array</p><p>Sixth line indicates input elements according to L</p><p>Output Format:</p><p>Single line represents the output that are closest to each other from three sorted arrays</p> </p>

answer

```
#include<bits/stdc++.h>
using namespace std;

void findClosest(int A[],int B[],int C[],int p,int q,int r)
{
    int diff = INT_MAX;
    int res_i = 0, res_j = 0, res_k = 0;
    int i=0,j=0,k=0;
    while (i < p && j < q && k < r)
    {
        int minimum = min(A[i], min(B[j], C[k]));
        int maximum = max(A[i], max(B[j], C[k]));
        if (maximum-minimum < diff)
        {
            res_i = i, res_j = j, res_k = k;
            diff = maximum - minimum;
        }
        if (diff == 0) break;
        if (A[i] == minimum) i++;
        else if (B[j] == minimum) j++;
    }
}
```

```

        else k++;
    }

    cout << A[res_i] << " " << B[res_j] << " " << C[res_k];
}

int main()
{
    int p,q,r;
    cin>>p;
    int A[p];
    for(int i=0;i<p;i++)
        cin>>A[i];
    cin>>q;
    int B[q];
    for(int i=0;i<q;i++)
        cin>>B[i];
    cin>>r;
    int C[r];
    for(int i=0;i<r;i++)
        cin>>C[i];
    findClosest(A, B, C, p, q, r);
    return 0;
}

```

question

<p>Question Description:
Mustafa defines the happiness score of a string as the number of indices j such that $M_j \neq M_{N-j+1}$ where $1 \leq j \leq N/2$ (1-indexed). For example, the string CABABC has a happiness score of 2 since $M_2 \neq M_5$ and $M_3 \neq M_4$.

Mustafa gave Bama a string M of length N , consisting of uppercase letters and asked her to convert it into a string with a happiness score of L . In one operation, Bama can change any character in the string to any uppercase letter. Could you help Bama compute the minimum number of the operations required to transform the given string into a string with happiness score equal to L ?

Constraints:
 $1 \leq T \leq 100$.
 $0 \leq L \leq N/2$.
 $1 \leq N \leq 100$

Input Format:
The first line of the input gives the number of test cases, T . T test cases follow.

The first line of each

test case contains two integers N and L. The second line of each test case contains a string M of length N, consisting of uppercase letters.

Output Format:
Print the output in a separate lines contains, compute the minimum number of the operations required to transform the given string into a string with happiness score equal to L?</p>

answer

```
#include<bits/stdc++.h>
using namespace std;

int main() {
    int i,T; cin >> T;
    for (int case_num = 1; case_num <= T; case_num++) {
        int N, K;
        cin>>N>>K;
        string S;
        cin>>S;
        int cur_score = 0;
        for(i=0;i<N/2;i++) {
            cur_score += (S[i] != S[N-1-i]);
        }
        cout<< abs(cur_score - K) << '\n';
    }
    return 0;
}
```

question

<p>Problem Description:</p><p>vijay has just finished baking several burgers, and it's time to place them on cooling racks. vijay has exactly as many cooling racks as burgers. </p><p>Each cooling rack can only hold one burger, and each burger may only be held by one cooling rack, but vijay isn't confident that the cooling racks can support the weight of the burgers. </p><p>vijay knows the weight of each burger, and has assigned each cooling rack a maximum weight limit. What is the maximum number of burgers the vijay can cool on the racks?</p><p>Constraints:</p><p>T≤30</p><p>N≤30</p><p>Input Format:</p><p>Input begins with an integer T≤30, the number of test cases. </p><p>Each test case consists of 3 lines. </p><p>The first line of each test case contains a positive integer N≤30, the number of

burgers (and also the number of racks). </p><p>The second and third lines each contain exactly positive N integers not exceeding 100. </p><p>The integers on the second line are the weights of the burgers, and the integers on the third line are the weight limits of the cooling racks.</p><p>Output Format:</p><p>Print the maximum number of burgers vijay can place on the racks.</p>

answer

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    int n,t;
    cin>>t;
    while(t--){
        cin>>n;
        int cnt=0;
        int no[32],w[32];
        for(int i=0;i<n;i++)
            cin>>no[i];
        for(int i=0;i<n;i++)
            cin>>w[i];
        sort(no,no+n);
        sort(w,w+n);
        int j=0;
        for(int i=0;i<n;i++){
            if(w[i]>=no[j]){
                j++;
                cnt++;
            }
        }
        cout<<cnt<<endl;
    }
}
```

question

<p>Question Description:
During some Research, Ragu found evidence of Predator poetry! Ragu's team of linguists has determined that each word in the Predator language has an accent on exactly one position (letter) in the word; the part of the word starting from the accented letter is called the accent-suffix. Two words are said to rhyme if both of their accent-suffixes are equal.

Ragu have recovered a list of M words that may be part of an Predator poem. Unfortunately, Ragu don't know which is the accented letter for each word. Ragu believe that you can discard zero or more of these words, assign accented letters to the remaining words, and then arrange those words into pairs such that each word rhymes only with the other word in its pair, and with none of the words in other pairs.

Ragu want to know the largest number of words that can be arranged into pairs in this way.

Constraints:
 $1 \leq T \leq 100$.
 $1 \leq \text{length of } V_i \leq 50$, for all i.
 V_i consists of uppercase English letters, for all i.
 $V_i \neq V_j$, for all $i \neq j$.
 $2 \leq M \leq 1000$.

Input Format:
The first line of the input gives the number of test cases, T. T test cases follow. Each test case starts with a line with a single integer M. Then, M lines follow, each of which contains a string V_i of uppercase English letters, representing a distinct word. Notice that the same word can have different accentuations in different test cases.

Output Format:
Print the output in a separate lines contains, find the largest number of words that can be arranged into pairs in this way.</p>

answer

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    int tt;
    cin >> tt;
    for (int qq = 1; qq <= tt; qq++) {
        int n;
        cin >> n;
        const int ALPHA = 26;
        vector<vector<int>> trie;
        trie.emplace_back(ALPHA, -1);
        vector<int> visits(1, 0);
        vector<int> pv(1, -1);
        while(n--) {
            string s;
```

```

cin >> s;

int t = 0;

for (char c : string(s.rbegin(), s.rend())) {
    int d = (int) (c - 'A');

    if (trie[t][d] == -1) {
        trie[t][d] = (int) trie.size();
        trie.emplace_back(ALPHA, -1);
        visits.push_back(0);
        pv.push_back(t);
    }

    t=trie[t][d];
    visits[t]++;
}

int ans = 0;

for (int i = (int) trie.size() - 1; i >= 0; i--) {
    if (visits[i] < 2) {
        continue;
    }

    ans++;

    int v = i;
    while (v != -1) {
        visits[v] -= 2;
        v = pv[v];
    }
}

cout << 2 * ans << '\n';
}

return 0;
}

```

question

<p>Question description</p><p>Selvam won the man of the match award in the recently concluded local tournament final. So the friends of Selvam have asked him to take them to cinemas as a treat for winning man of the match. But
<p>Selvam is short of money to take them to cinemas so to postpone the cinema plan he tried to engage them with the programming challenge.
<p>The task was, Given an array of positive integers, write a program to find minimum number of merge operations required to make the array palindrome.
<p>If it is not a palindrome it will make merge operations and prints the number of merge operations. In each merge operation it will merge two adjacent elements. Here, merging two elements means replacing them with their sum.
<p>A palindrome is a word, phrase, or sequence that reads the same backwards as forwards.
<p>Function Description</p><p>Create two variables i,j. i will point to the start of the array and j to the end.
Till i
less than or equal to j
a. If arr[i] = arr[j], then there is no need to merge the elements. Increment i and decrement j
b. If arr[i] > arr[j], then do merge operation at index j ie, arr[j-1] = arr[j-1] + arr[j], decrement j and increment the no of merge operations count by 1
c.&br> If arr[i] < arr[j], then do merge operation at index i ie, arr[i+1] = arr[i+1] + arr[i], increment i and increment the no of merge operations count by 1
Constraints
0 < n <= 100
Input Format
First line indicates the number of elements n
Output Format
Single line indicates the results
Example:
Input :
arr[] = {1, 7, 9, 1}
Output :
Number of merge operations are 1. Merging 7, 9 makes the array palindrome
&br></p>

answer

```
#include <iostream>
using namespace std;
#define f(i,a,n) for(int i=a;i<n;i++)
int minOperations(int arr[], int n);
int main()
{
    int n,count=0;
    cin>>n;
    int arr[n];
    f(i,0,n)
    cin>>arr[i];
    f(i,0,n/2){
```

```

    if(arr[i]==arr[n-i-1])
        count++;
}

if(count==n/2)
    cout<<"array is already a palindrome"<<endl;
cout<<"Minimum no of merge operations took is "<<minOperatins(arr,n);

return 0;
}

```

```

int minOperatins(int arr[],int n)
{
    int ans = 0;
    for (int i=0,j=n-1; i<=j;)
    {
        if(arr[i]==arr[j])
        {
            i++;
            j--;
        }
        else if (arr[i] > arr[j])
        {
            j--;
            arr[j] += arr[j+1];
            ans++;
        }
        else
        {
            i++;
            arr[i] += arr[i-1];
            ans++;
        }
    }
}

```

```
}
```

question

<p>Question description</p><p>Janu and Ram are close friends who task a lot about life.</p><p>They go through a lot of inspiring "Quotes of Life".</p><p>One fine day they had a small game. According to the game Ram will Read one of the Quote about life from the book and Jannu have to think a word about life in her mind without disclosing it to Ram. all rounds completed, every round they are getting equal marks. So they decided to make some technical test on algorithms to prove themselves. they have to open a book randomly and then they have complete the same task. likewise janu opened the book and got the task is, Given a binary array and number of zeros to be flipped, write a program to find the zeros that needs to be flipped so that the number of consecutive 1's is maximised</p><p>can you help janu?</p><p>Function Description</p><figure class="image"><img

76xu4awa+2v7W5ahRuKdSeMtReWBJB5qVYUYOdWahFc23gr3r+mKe75VDvzwAsgc6NelcU1OjUjOL5
OLydAPkpKEHKTwkstkO31exuqyo0K6nUfVhne89zr/y5fkYXZqUYa7GUmkpZbA/QAVz17TFU3Hd089u
eBOpVYVqanSmpwfJpgewfJSUYtyaSXNs6mvabTnuu6g39GBYg5W9zRuqe/QqxqR7YvJ1AHjpqS/wD9If
5kez8zar3OoOjTqS3pzaXpMD9J6al8yH+ZH2NSEniM4t9iZi/7L6t86P8A5hYaHol/Yaiq1xUUqai1jeyBcT1
mwp3DoTuEqqluuOHzPWp6lR0y26atlpvCiubMPf8A956v/ef/AFGy1y3s7iw3b6p0UE+E84wwPuk6zQ1
aE3SjKEoc4yLEptnrTT7alVdhW6ZtrflnJMqatY0q/QVLiEamcrfWBNAr61p9vPcqXMFLrSflkWt7b3kd6
3qxqlr3XyA7g43V3Qs6aqXFSNOLeMtnFatYuh0/INPo843s9YEwEChrWn3FRQp3MN58k3zI4AHG5uqF
pDfuKsace2TwRKeu6bVnuRuqefq8AWICakk08pnOvXpW9N1K9SNOC65PAHQg1dYsKVw6FS4Sqj43cP
mela7ptSpuRuqeXyyzG6s09o6jT4dMvzA/Q08pNcmDzT9IDuR6AAi3Wo2Im8XFen9jfE4Utd02rLdjdU0
/q8AWIPkZKUU4tNPk0RbrVLOzlu17iEJfDniBLBCttXsbqahRuYSm/4c8SaAAM1tPrE7aNOFdqNRSSamo4
bQGIOF3e29ITU7moqcW8JtFPoGswqafm/vIdNvP1mk8HPbt920f5i/JgXIpeW97TdS2qKcU8NpHcy+yt3
b2mk1Z3FWNOPSfxPHUXNDWtPuKip0rmDk+Sb5gTwCjc6nZWjxXuKcJfc3xAlgr6Ot6dXmoQuobz5Jvm
Tp1IU6bqTkowSy5PkB6BidX167hqs42l4+gysbuGjV2eoW1xCnCFxTqvXFZsfECWAc69elbUpVa01Ceby
HQEOlq1jVpqzQuabhD1nnkcoa7ptSpuRuqefqwLEhxNSWU8rtlc9XsKdd0Z3MFUTw4t9YE0Ahed7Dp+g
8ph0m9u7ueOQJoOF1d0LOmqlxUjTi3jLzyWrWDt3X8pp9HnG9nrAmAg2+sWF1UV0jcwc3yjnizrd6ha
WWPKK8IN8k3xYEk8VasKNKVSpldhFZb7CNa6rZXkt23uISI8Oej81n90XP2AfBTVLO9qOnbVIOSwCJM
mGF2QqQpajVnUkoxVN5bf1K17TXU3PKqee3PACxB5hOM4qUJKUXyaFSpCnBzqSUYrm2B6BWVx9M
U913cO/PAn0a9K4pqpRnGcHycXkd2Cnd39tzbvlnaNPe5bz5nieq2MKEa0rmmoS9V55gTAeKNWFelG
pTe9CSymewAONzd0LSG/cVY049sngiU9e02pPdV1BP6sCxPFWrTo03OrOMlrm28HpTi4b6accZy7a/U
YV5UqFvWjOCWZbr6/qBo17QaZKoqaulvN49Vlkmmk1yZi9H0jTK1GhVrXn/SJNS6NSXhg2i4LC5AD42or
LaS7WcL+8p2FpO4q+rFcu1mFr3+pa5d7ljz4vhCHBJAb1XNBvCrU2+zeR1MM9I9UjDfUo73PCnxJmzdTV
qd7K3rRnKjDhPpP4fxA1oPNSpClBzqSUYrm31FfLX9MjPdd1DvT4AWQOdC4o3NNVKFSNSL64vJ4ury3s
4KdzVjTi3hNsDuCH51sfJ+n8pp9HnG9nrPFDWdPuahSuYOT5JvmBPAll1qdnZvduLiEJfC3xAlgr6WuabW
luxuqab6m8E9NNZTymB9BDutWsbSe5XuIRI8OeJ9tdTs7yW7b3EJy+FPiBLAI11qNpZ+8V4Qb5JviBJAt
9a0+5moUrmDk+Sb5kutWp29KVWrJRhHm2B0BftNRtL2bjbV4VHFzaTO1xcUrajKrWmoQjzkwOglqtqV
pdqboV4TUFmWHyMvtBrlzT1HFhefst1ephrlGyKrVtettKqxpVlzqVJLo9Sounanb17ahGVzTnXIBZWeLe
CBtBZaZcV6c7246CrjCeUsoC3sL2lf2kLijndl1PmiQVlrX07S9NoqFeMaEvVIJ+syba3dC8pupb1I1p4bT6w
OwPFatTt6MqtWShCPNvqONpqFreuStq0ajjz3XyAkgj3V/a2aTua8KeeSb5nC31nT7mahSuYOT5JvmBPA
AHI1aaeHUin9WI1Selzi32JmP1nRNQ6e5vI1UqSzPG/1lj7JVJy1pKU5NdG+DfcBr7vVLOyqRhc11CUuSw
2SYTjUgpwkpRksprRMVthauGqRrb+900ViPZjgarSKUrbSbenUfGMOLfiBNBXVdd02lIdQbXPDydt7XUr
O8eLe4hOXYNxAlHG8uqdlazuKzxCC4nYajSt61jVhdvdotek88gIOlbQ22p3DoU4ThNLK3utFuZ7QLHSqF
3Odnc9PVx1tcEXxd29pDfuKsace2TwB2BWLdTHLHIUF9WywpVqdemqlKcZwfJpgewAAAAAAA
AAAAD821Capa/XqPioXDI4SP0k/N76KntDWhJzjK5aa+m8BoltbY93qf0LHSNcparOcadOUNxZ9I+rZ3Ss
L/oq/zP8A1JNnplpYSIK1oqm5c+LYGK2r/f8AX7o//wAo1Os/3Zqfyl+Rltq/3/X7o/8A8o1Os/3Zqfyl+QGN
0jT6up3Xk9Oe5HG9N9iLHWdmvNtn5RsrupGLxJNYO2xPvtx/L/VF7tN+464FdsbeTqW1a3qSbVPDjnqRS
aze19W1Z0It7qqdHTj+OMlhsWs1blsUVNRy03XpTqR9nW3u9Zz+QF7HY2Hk+Xcy6bGfV4ZM10NS31K
NGr68Kqi/E/QFrWnu26byqnjGd3eWfAwldk81p3EViM6ya7sgfpFP2Ue5Ho80/ZR7kfW1GLbeEuLA8V
6903oyq1ZKMlrbMFqV7ca9qUadGLcM4pw7Pqd9pNbd/WdvQl+wg+OP4mW0zq03TqHS1rqj5RNcfS
XorsAuth0unpdoqccOo+M5drKDa/VKirKxoycY4zPHX9DQrWNpk0ld0sv8A5kZla63IT1RV+dOrFNPqAk
WGz9hUtl1Lq9iqk1nCkvRK91Kmgav/ANHrqpTTXGL4SiTd00vRby1jOd46VThpRIJLDO1HRdEr3nk1K9n
OpjPBpp/iBI2vmqul29RcpPJSaNo1fV84qblGn1vt+hd7Xuo0dKt6Uc4jLCO2xv7rn94Hq+m9ntAjRpT36n
qxk1jn1lBoui1daqVK1as4wT4y5tsv9r7adbS1Ugm+illpdhX7Jarb29GpbXE1Tbe9GUhnARNb2flpVKNzQ
qudNPDysOLLTSNxR3Gg3WW5V7ePCXW11HzanVrapp7taFWNWc2m3F5SSZz2Pt+is7q5r8KU8LjyaWc
/mBR6XRtb28l5xuXTTWd5/wATL6OzGn1pwlbXmUnxjlPKPE9l0TUG6treKi3xVJfkyk1C2803kY2t2qjxn
epy5d4H6NCKhBRisJLCOGoWivrOpbubgprG8lnB50utO40y3rVfXnBNkoD8yv7JWepztFNyUZbu81g2Oj
7PQ0y5VxG4lUbjjdccGY19bm0NzY4LfT/AANza31rcRpwpV6c5SjIRUK2BjNd1Cvqeq01pt9HGW5CK632I
nDYyDt/TuX0rXVHhkplqM9L19zqRf7Orvr6rmbaGtafK26byqmljOHLD8AMbpt5caJq3Qzk9xT3akeprtP
0CL3oprr4n51cSer67KVGL/azSXdyyfolNbtOK7EkBzvPc6/8uX5H5taWtS8vo29F4lOWM9iP0m89zr/y5fk

fnukXkLDV4V6nqJtS+iAva+x0Y20pUrlyqpZw48GV+zWoVbLVI2IST6OpLcceyXI1VfW7ClbOsrqlLhwipJt/gY3R6cr3aGnOEXjpelf0WcgWW1up1ZXfkNKTjTivSw+bPtns7p8rWMrm9iqslnCkvRIO1VvKjrEqkk9yqt5MI2OlaJd20ajvXTnjOoykk0wINCvU0LWN2IWVSmms7r4SR+gRkpRUo8U+JkbbRdFubt29C8qTqRWeDWH3M1tOCp04wXKKwB6PzOIxVtqyryTahUbaR+mH5ra0oVtahTqLehKq012gaT+2Vt/w9T+haaRrFPVIudOnKG5jO8ef7O6X/wAKv8z/ANSXZ6fa2CkrWI0e9z4t5Awl/wD3nq/96/8AUafa/wDcr++P5mYv/wC89X/vX/qNPtf+5X98fzAibEe7XX3R/Uodez58r7vPe4F9sR7tdfdH9Si1yW5r1aaWd2aYFza7IRq2salxcSjVms4SykUz8p0DV3FT4wfHHKUTZWWt2FazhUdzTg1HjGUkmjHa5dx1TWHK39KLxCPDmBoNq6qraHRqrINqRRaFo1TVpS3qjp0afN8+LLraWk6OzttTfOO6mRNktUt7VVbe4mqe+04ylwQHHXNnPnturmhVdSCeJZWGi02Z1eVXT61O4lvO3WVJ9a/9obT6vay06VtQqwzqNeq8pLJA2Xsalayvp4ajUhuRfa+IFcqIXxtW/6RWVODfOT4RXWd9s9YU7SU7W9i6sFnDkvSKK0pUI3yo3+/Tgnuya5xZoJaLoUaXSPUPRxnhNAfdkdTqOpOyrScoqO9Bvqx1FZrN7X1XV3Qg3uqfRwjnhzxkOGh6Rp6avLKvOpIOPHqMzVctM1+UqsX+zrb2O1Zz+QF5HY2HkybuX02M+rwyZp0alvqUaNX14VF/gzfrWtPdt03IVPdxnG8s+Bhbq6V5rTuIrEZ1U13ZA/SKfsodyPNxKULepKhrRi2vA9U/ZQ7kfZNKLcuSXED83tOiv9SzqVw6cZtuUnx49hff2a024ivJL30u/OT1W03Q9TqzqUbpUJ7zyIJLL7mUmq6fDSqtN217Gq5fBLjHvA1et3a0bRowt1iTxA+zhzM1oui1NaqVK1aq4wTw5c22TNR8o1HZS2uZ5IOPI0vquWT3slqtvb0alrcVI025b0ZSeE+H+wETW9n5aVTjcW9Vzp5w88HF19stqdS/spQrveqUnjPaijtTq9rUsPjaFWNWc3l7rykhsVbThb1q8k1GbxH6gacw21GIrsq7ulVc3cVG3FrGM8TcmX23i/jreXVv4/owK/RNnlalZq4lcyptSxuqOS02zWNMorsql8mfNlb61o6VKFwvTpjNtqUkj7to09Motcc1F+TAotD0Ser77IVdOIB4bxniNc0SWjzpyhV6SnPk8YaaL/Yv92Vf5n6HPbb3Oh9zAsNmrud5o9N1Xmccxb7cFZW2XoO9q1l8jCnKWYxzx/qddl6jpBP1qkvIw3pL8MmfsvPWdW3L25IBTy8t/0QHbW9Ht7ClGta3UaqcsOOVIFzs9Unquh17StUa3cwUubw0VevaPYabbRdGvKVZyxutrkwOxHu9x9/wCgGc1OxVhqMrVTc1Fr0msGs0bZ2FhcU7tXMpv9VxxzRntpVu69UcuHjmzsbt+1rUKEKdxTIOUViKks8uwCaVO0/wC4rjuX5lsVO037juO5fmBjtG0yrqtW6MJ7I0KzN9hN1vZx6ZbK4pVnUhnEk1hol7Ee3uftX5lttX+5Kn3ICLsdeVK9IUo1JOXRP0c9hndS/vFV/moudiOVz+BT62pW+0FaUlympL6oD9DXJH52v7zr/vP/AKjYraDTvJOM8oh6udzPpeBirar0+vUqvJTuFLxkBp9sv3TT/mL8ih0PRZ6upuVV06NPhnGeJfbZfuqn/MX5HnYr931v5n6AZ/W9Ino9xBRqOcJLMZYw8kzStDra3Cd3dXMkm8Jvi2TNt/VtO+X6Fnsn+46f3MDialY1tF1FQjUeViUJrgbO6ru52ZnWfOdHLM9tr+86P8r9WXUP7nL+QBj9LsKuo3kbek91S9aXYi+vtkVRtJVLevKdSczutYyVWzl/S0/U1OtwpzW632Gv1DXLG3s5zhcU6kmvRjGSbbAz+yOpVKd55FUK3Tqeqn1M5bS6IVvNRdnTlu0oNRxnCb7WcdlredbWqdSKe7SzKT/DH6nDXbaVtrNZVU92Ut5PtTAuaezenO2XSX8ema5qSwmV2k3IXSNZVv0inSIPclh5TzyZOTT004oKor9xyuKINJo62GjaPc3TjbXVSc6Uk8ZXEZdtvx8ka/5v0K7R9Br6tS6WdXo6MfRT5t9xYbbLCtF2Z/QtlVjQ6X3MCys7fyS0p0Fle3FjOOZ6uayoW9SrLiCLZ0I+o0XcafXpR5yg0gMHBXW0Wq7sp+s88eUYlve7IKlayqW9eUqkVndaxkqtAvYaZqublOMXmEsr1Wa++1uxt70dSNxTqSa9GMZJtgZ3ZbU6iuHp9WbdOqmo5/heCDtBpcdLu4xjVdTpFv8VjHFnTZe2nca1TqJejSzKT/AAx+pM229+t/5f6sDtoezkKIK2v/ACmSbxLc3eHia0rdn/3Ha/YiyAzO21SubS2pr1Zyk3+GD1sZQhGwqVsLfnLDfYiVtTYTvdM3qUXKpSe8kutdZn9m9bhttps://50LnPQzec/Cwn0CBLW9OjT33d08dilx8CLp+0lrf3sreKcG/Ub/AlgM9tjqVW+1F2ljvo4S3VFdcvA85J9HY2ErZOrctVWs4UeCKbVaU9P1+c5x5VeI1Wcm0oa3p9W2VXyqnhhxjKSTX4AY21r3Wgau6TlwjLdms8JLtl3bKSnlvNcpSyvBGf1Kv521xyt4tqjRjw5rtL7a6Dp6RawfOLx/RAU2h6Ju1eMnKq6dGDxnGeJ81zRZaRUpuNV1Kc+UsYaZoNjP3VP72cdt/dLb+Y/yAmaTf1q+zbr8ZVaacc9uDI6fChfx7eo3LpxlxcnxyzVbiuK0KTnjdVSWc9yldTS9D1KTq212qMm/V3kv6MD7/AGZ064Sdre8c8s5yT9ob56VpUaVv6M5+hH6LtMpqdktJuleTXiqN8cwlXgWusqvfBOWd5NOUoes/+rAiaLpNDUYTuL66jBZ4Jy4v6njV9Op6Twp17G6U1n+GXGLPOjWemXkZQvbiVGonw4pJr8SwraRoNGcISv5NzePRknjvAsY67L+zXlr9v6n+Llkz2k6ZW127qVK1Vqk4zm+LZearosLbZ6dK0lKooS6Tj1IfsnqdCznVoXE1TU3IsfBAdb7ZGpS3JWNXpHnDUuGPqWd9b17bZepSuavS1lxxk7320NhZKOKsazk+KpyTwu0ajXp6ls/Xq2st6Lg3y7AKLYr32v9iLzaj9w3H+H80ZnZa/o2OoS8okoQnHG8+plttPrFpU092tvVjVnUazuvKS5gRNjqXTK+pt434JZ8Sp1vTipl90CqOot1Sy1guth/a3f2x/Uh7Yxa1dPqcEBaaLs5CjK2vvKZn7qlubvDiiHtr73Q+1/pN/az0+1hGvT39xR3d5Zz3FBtr73Q+1gTadkr7ZCEEvTjHej3or9kL10L6dpUel1OWepo0WzvHRLf7TJ6xQnpGu9JS4LeVSAFtnfuFKnZwfGb3p47DppUPMuzdS7nH9rUW9x/oUdNz17aCMmmoylnD6oo1uvWjr6HWo0V6sU0l9AMjp9s9av6ls9uVCK4ylJ8X9EstX0S0tbV17K7jNwxmDksvuK3SqNIxuXTv6kqSfKS5Z+pdV9H0K3p9JPUG12Rmm3+AFhsnqdS8tZ0Kzcp0cYk+tGgKnQ9LtkMrizqyqRrRxFstler/Ui7/ky/Ix+yH77X8uX6Gw1f90Xf8mX5GP2Q/fa/ly/QDUX+iU7/AFCI

dVa0sUsYhjgQds9Wjp1KnSbjCpjwOzsNEQdXlYqz3dRx0UnhZ7QMjo2n6ZeUM3d04Vc+pywX2I7P2t
nfK6oXCqxiuEexIdU2c0yrFztRik+pyTSKzSLitY63To0au/B1VB4fCSzgD9CIGufue5+wnkDXP3Pc/YBI9jP3I
V+wh61WIc69Vjczcacam59sck3Yv95VfsLXVLXRtQvZ061eNG5jwlJPGfHgBDp7P6PcUv2F96TXPP6M0G
k2ENOsYOIS3uty7WY/VtFoWFv09C+hUafCOVvPwLzY+7r3FnVhWk5Km0ot/UDQgAAAAAAAAAAAAAAA
AGcq7KdLqUrvyvG9V6Td6P65xnJowAXBAADP6rsx5yv53XIXR76Xo7mcYW00tLyx8q0yVn0m7vQ3d7G
f6EwAU2iaD5orVKnlHS78d3G5jHHvJ2p2XnCnb/R7/wDFjOCWAKfRNC80TqS8o6XfWPVxj+p11bQ7b
VEpT9CquU1+pZgDJ09jMVP2I5mHYoc/6naWx9NXUatG6cIRaai4Z5fXjpgB8it2KXYsHmrDpaM4ZxvRay
ewBk/7Ff8A53/6/wDcf2K//O//Aff7msAGVhsZuVly8uzhp+z/ANzQXWn0L20VvcR3opYT619SUAMpW2
M9PNG7xHscOX9Sbpey9CxrxrVarrVlvMeGEEmXwArda0nztbwpdN0W6853cnrRtL81Wro9L0uZZzu4LA
AfJxjODjJJxaw0zOX2yFCtUc7Ws6Of4XHKNIAMxabHUoVFk6uHUS/hjHGS/qWNNGpYu0Udyk47ul8MIkA
DK1djFvZoXbiuxx/XJ1s9kKNOqp3Vd1sfwpYNKAPkIqEVGKxFcEkfQAKjWNn6GqSVTfdKsljeSzIEPS9I5WF
7C4d3vbnJKOM/1NGAK/VNHttUglWjia9WceaKJ7Fy3uF6t37P8Ac1oArNK0O10xb0Fv1Xzm/wBCzAA8V
odLRnTzjfi457MozdHY6IGrKVa6dSLT4KGMPtzk04Ayb2L/AGnC89D7OP5I7pekW2l08UU5TfrTfNk8ARb
/AE+31Gh0VxDK6muM7U2M9PNK7xF9Thy/qawAUelbM0NPrRrzqOrVjyeMFJ4AAM5Q2U6HUI3led
2e9u9H/uaMAAAAbnjZbpTleeV43qvSbvR/XOM5LPV9N86WXk/S9HxT3t3JPAFXomj+aKdWHTdLojtzu
4xgx+tRUtoKsXydRJn6IUF3sxC61CV07iUW5KW7gCHV2OhUanQuXCMknuyjnBY6Ts3bafUVWcumqrk2
sjfgXMY7sFHsWD6Bn9sv3VD7ym0PQ6WrWFSTqOnUhLCkIlleBqtY0yOq2qoyqOnh5ykedG0qOk0J0o1
HU3nnLQFNb7GRVV0vdOUF/DGOM/jk0tvb0rWhGjRiowiuCR1AFPqutzrqU3VTdKs+corn3oqo7Fy3sSv
PR+kP8Ac1oAhaZpdDS6Dp0Mty4yk+s46todtqiUp+hVXBTRZgDJw2MxUXSXmYJ8IDn/AFO09j6fIuatG6
cIRaai4Z5fXjpgB8it2KXYsHyrTVWIKnLOJLDwegBlq+xsHNyt7uUF2Sjn+uT1b7HU41FK5uZEv4VHH9TTg
DnC3pQt1QjBKko7u71YM/e7IUk1Rztqzo547rWUaQAZi02OpU6ilc3DqJfwqODSuqUKNKNOnFRhFYSX
UewAI2oWNHULWVCuvRfJrmmSQBk3sW9/wBG8xH7P9y11HRHf6dQtHcOPRNem45zhd5bgCu0XS/N
NtOj0vS70t7O7g863pHnejTp9N0W4853c5LMAV+kaWtMsnbup0qbbb3cFVe7I0qtd1bW4dHLzutZNKA
MzHZCDoTVS6IKrLIJx4R/DJZaJo/minUh0/S77zndxj+paACp1nQaGq4m5OnWisKaWeBA07ZWVne07h3
be48pKOM/1NKABE1Oy84WNS23+j3/AOLGcEsAU+h6F5onVI5R0vSLHq4x/UmarYecrGVt0nR7zzvYyTA
BU6Honmjpf2/S7/8Ay4x/U9axoVvqqUpN06sVhTS/MtABIkOxkVVTrXW9Bc0oYz/U709koUr+FxTusQh
NSUNzs+uTSACu1nS/OtpGh0vRbst7O7k+aLpPmm3nS6bpd6Wc7uCyAFTrmi+d1SXT9F0ef4c5z+JK0m
w822MbbpOk3W3vYwTABS63s/52uYVvKei3YbuNzPX3kxadjR1Ydj/Bub+POJwAzFDY2jBVFWuXU3o4ji
GN19vM4rYt7/G89D7P9zWgCHpum2+m0Ojt48/Wk+bPmpaVbanS3K8fSXqzXNE0AZKwxb3/AELz0frD
/ctdi2eoaxU6XfdWrjG8+CX4FwAKjXNE87ul+36Lo8/w5zn8SXpVh5tsY23SdJutvexgmAAAAKbVdnLXUZ
urFujVfOSWU/wKynsW9/8AaXmYrshz/qawARDp06306h0VvHGebfNkDW9A8716dTijotyO7jcznj3lyAI
2n2nkNjStt/f6OON7GMkkABjJR6lsvaXs3VpSdCo+Lwsp/gXgAx62Mq73G7il27n+5eaVoNrpnpxXSVfjl+h
aACDqeIW2p0lGvHEI6s1zRQS2Le/6N56P2f7mtAFVpOgWumPfjmpV+OXV3HvWtJ87W8KXTdFuvOd3J
ZACv0bS/NVq6PS9LmWc7uDxrmked6NKn03Rbks53c5LMAVunaPGy0udjOq6kZtts3Xx/8AgqK2xsd/N
vduC7JRz/XjqQBmLXY+ICop3Vv6qX8Kjg0fk9LoOg3F0WMbvVg6ADNXmx9CpUcrWu6SbzutZPFrsdThU
Urm5dSK/hjHH9TUADxCICFJuoxxBLCX0KHUNK7a5qupb1HQjLi1jKNCAMtbbGwjUUri5c4r+GMcZ/HjpK
NvSoUFQpQUaaWN06gDNX2yFGvWdS2rOkm8uLjlHmlsdRjQnGpcOVWS9GW7wj+GTTgCo0PQ/NEqr6
fpeksXq4xj8Trq+jUNVpxVRuFSPqzXUWQAY9psjKhdU6rvOFOSkkoc8fiTta0DztVpz8o6LcWPUzn+pdAC
Np1n5BY07ff39xY3sYyRNb0WGr06adTopwfCW7ngWgAp9E0CGk1KIR1umnJYt3cYRcAAUOpbK2t5UD
WjN0Jvi8Lkf4EGlsZ6a6W7zFdShz/qawAcLKzpWNtGhQTUI9vWdwAON5Q8qs61De3ekg45xnGUU+kb
N+a73yjyprPRcd3cx+pfAAQNW0mjqtKMks5RchmLRPAGSexk1LEb30fs/wBy00nZy206oq0pOtVXKTW
EvwLkADhfW3lInUob25vrG9jODuAKTRdnvNNzKt5T0u9HGNzH6nHUNIKV3cTr0ridOc3I5W9xNCAMPt
2NbKumvG49ij/uakwsaGn26oW8cRXNvm2SQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAA
AAACu1vVVpNrGr0fSSm92KzhAWIKnQtaWrwqZpdHOnjKTyuJbAAAAAAFFrm0Xmu5jQp0ekljLbeEiz0
29jqFjTulx3d/muwCUAAAAAAAACl27p2NtKvWzuR54A7ggabrfqkpq3U8wWXvIngAVmt6stJt41FT6S
c3hLOedC1latSm3S6OcHxSeUBagAAAAAB8bUVmTSS62B9BCqavYUpbs7qmn35O9C7t7lZoVoT7mb2
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AA
AAA
AAAAAAcK13Roy3ZS49gSzIPMu4InnGh2vwHnGh2vwDjqU3SwRPONDtfgPONDtYOpTdLBE84001+A8
4001+AOpTdLBE84001+A84001+AXqU3SwRPONDtfgSKdWNWO9B5QWL1n1L2AeKtWFKG9N4QWZ
w9giecbf4n4Dzb9rDnqU3SwRPONv8ThnGh8TCdSm6WCJ5xt+1jzb9rB1KbpYInnG37WPONv2sHUpul
giecaHazpRu6NaW7CXHsCxqVn1LuAA7AR6l9QpT3ZS4rsPHnG37X4Bx1Kx+0sETzb9r8B5xt+1+ATqU3S
wRPONv2vwHnG37X4A6IN0sETzb9r8B5xt+1+AXqU3SwRPONv2vwHnG3+j+AOpTdLB5hOM4qUXIM+
Va0KMd6bwg6zGMvYInnG37X4Dzb9r8A56IN0sETzb9r8B5xodr8AdSm6WCJ5xodrHnGh2vwCdSm6W
CJ5xodr8B5xodr8AdSm6WCJ5xodr8D1C/oTkoqWG+0L1K7pIB4q1YUY703hB1M4ewRPONv8T8B5xt/if
gHPUpulgiefbf4n4Dzb/E/AHUpulgiefbf4n4Dzb/ABPwB1KbpYInnG3+j+A842/xPwB1KbpYInnG3+j+B
0o3dKtLdhLj2AjUrPiJdwCPdXkLbCeXJ9SDqZisZlBXedY/Ax51j8DDz61N1iCu86x+BjzrH4GDrU3WIK7zrH
4GPOsfgZMnWpusQV3nWPwMedY/Axk61N1iCu86x+Bki2vIXDwsqS6mVY1aTOIIJAi91dwkt7Lb6kHcz
FYzKQCu86x+Wx51j8DDz61N1iCu86x+BjzrH4GDrU3WIK7zrH4GPOsfgYydam6xBXedY/Ax51j8DJk61N
1iCu86x+Bki2vYXEt1JqXYyrGrSzxEplBwbuFslvZbfUg7mYiMy7grvOsflsedY/LZMvPrU3WIK7zrH5bHn
WPy2MnWpusQV3nWPy2POsflsp1qbrEFd51j8tjzrH5bGTrU3WIK7zrD5bjlvcQuib0OrmmHVdStvES6gE
O41CnRnuJOTXPAW1or5IMBXedo/LY86x+Ww461N1iCu86x+Wx51j8tj1qbrEFd51j8tjzrH5bjk61N1iCu
86x+Wx51j8tlydam6xBXedY/LZMt7iFxT3ofigtdStpxEuoBEUL+FCe5hya54Dq1orGZSwV3naPy2PO0flsZc
dam6xBXedY/LY86x+Wxk61N1iCu86x+Wx51j8tj1qbrEFd51j8tjzrH5bGTrU3WIK7zrH5bjdtcwuYbOOD
XNMOq6lbTiJdgCJcX9OjPcw5Nc8BbWisZlBXedY/LY86x+Ww461N1iCu86x+Wx51j8tj1qbrEFd51j8tjzr
H5bGTrU3WIK7zrH5bHnWPy2TJ1qbrEES2voXEt3DjL6ksrutotGYAAHQAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAADMX9xbdebb/iZoHyM9W9tP7mSWTifUPAAIxgAAAAAAABZaQ
3+0WeHArSx0f1qvchD10PuFoVmnrN5guosyr1f14dxZbNf4IXAAj5wAAAAAAgHW2bVzTw/4kcjpb+8U/
uRVr7hoVvPNR4pyf0PS5I81PZy7jp9WfTOybc2/qfD7L1n3nw5fJn2AAAAAAAAC30pt27y+TOGrN9LB
Z4YO2k+wI9xw1b20O4v6bLfhQAARjAAAAAAAAABoaDzQg32l9Wb36azwwyfbe7w7iv1b2IPuZW/V/
GrwAcsAACgACAAACgdbVtXFNrtOR0t/eId4WvtoSI1P3t9xdFlqfvb7iy3cR8lgAlwAAAAAAABK073uJFJ
Wne9xEO9P6hdlPqnvX+EuCn1T3r/CWWziPhCABGAAAAAAAIFh73Ajkiw97gId0+oXpTap71+BclNqf
vX4Fls4j4QwARgAAAAAAAACx0h+nUX0RXFjpHr1O5CHro/cLRmer+3qfc/wAzQsztfr9T7n+ZZe/E+oeAA
RjAAAAAAAAACx0l+IMrix0n1plh7aP3C0flz9y83NTPxM0D5Geufean3MS9+j9Q5gAjEAAAAAAAfpHtJ9
xFhpHtJ9wh66P3C1Zn7njc1PuNA+Rnrj3ip9zLLRxPqHMAEygAAAAAAAHey97p95fFDZe90+8viw3cN8
yAArSAAAAAAAAAAAAAAcnc0E8OvTT7HNGO2i1m4uL+dpbzckChu+jzkdZXUaINTIwipPjhsDa
xIGSzFprtR9Mroel6pZ6qlXnNUEm21LKZ12n1eVtRhGyulGqpNTUWm0BogZ7Z7WY1bJvULyDq72FvtJ4N
Ammk08p8cfgQQbjWbC2nuVbmCkuaT4o62uoWt57tXhUa5pPigJIPFatT6UqtWShCKy2+oi0tWsK0Jzh
c03GCzJ55ATQV0Ne02c9xXUM/V8CwjSipRaafJoD6Acby7oWdlpLipGnHtb5gdgcra6o3dJVLePgpB9az
16gPHS0/mQ/wAyPsZwn6soy7mYbU9F1C0p1bqpV/Zp54TJmxVSc7q53pSI6C5vPWBrwAAAAAAAfHy
M9W9tP7maF8jPVvbT+5klk4n1DwACMYe40aklmMJNfRHgsra+pUqMYSTbQd0rFpxMoPQVfly8DmaG
TTpNrrRnnzYd6unFPQAA8QstH9ar3lrSy0f1qvchD10PyQsyr1f14dxaxFq/rw7iy2a/wrgAcvnA5vCBaWF

GNPpqiy3xWeorulJvOEBW9aSyqcsdxzIGUHiUWu9FnU1OEZYhDeS6zsuhvqDeOP9UXD16VZ8VnypQeq
tN0qkoS5pnkjPMYDpb+8U/uRzOl7xT+5Ba+2gXI+VPZy7j6uR8qezl3HT6s+mdl6z7z4fZes+8+HL5MgB3
tKHIFZRfqrwsRMziHOKdT1lt9yPU7etBZlTkI3FzOdK1pJtYXVg8ULyjcS3FwfY+suGno19TPISAsdRtVBdL
TWO1FcR4XpNjIAA4W2k+wI9xw1b20O476T7CX3HDVvbQ7i/prt+FAABGQPbYym8RTb7EfDvZ1o0a+/
POMYC1iJnEvHQVfly8DzKEoPE4uPei8t7mFwnuZ4dpB1b2sO4uHvfSiK80SrARnAABoLb3eHcV+re1p9
zLC293h3Ffq3tafcyt+r+NXgAjAHqMJTzuRcsdiPKsbiFvObnnig6rETOJcegq/LI4HiUZQeJJp9jL6hWhXhvw
5ZxxKvUve33DD11NKK15oIEAAeAdLf3iHeczb+8Q7wtfcNCUup+9vuLopdT97fcWW7iPIEABGAPdKjUq
yxTi2fIRc5qK5tI5TpwtqHYorLYeulp8/tWebq+M4Xdkj1aU6Ut2pFplj50W/jc9HtySatOF1Q7crKZcPXpUt
H+ZUQPU4uEnF808HkjKErTve4kUl73EO9P6hdINqnvX+EuSm1T3r/AAlls4j4QwARgDrRtqtdN04558
TkSrS88mjKO7vZeQ6pFZn/T55vuPg/qjhUhKnNxksNF7b1emoxnjGSnvfep94e2pp1rWJhwAAZwkWPvc
COSLH3uAh3T6heINqfvX4FyU2p+9fgWWziPhDABGAOIkJUrPFOLZ8pU3Vqxgutl2ITtaHDhGKEPbSO+fzP
pWPTq+7nC7skapTnTluzi0yxjqkd/EoYj25JNzRhcUHw44ymV30qWj/ABKjB9acW0+aPhGYLHSPXqdyk4
sdl9ep3IQ9dH7haMztf29T7n+ZomZ2v7ep9z/MstHE+oeAAriDvTs69SCICGU/qjgTrfUOipwp7mccM5D
ukVm9OTsLhLLh/VEDrDNBJ5pN9sTPy9Z94l6aunFMYfAAHgFjpHrTK4sdl9aYh66P3C0flz1z7zU+5mhflz
1z7zU+5llo4n1DmACMQd6dnXqQUoRyn9TgTre/wChpRp7mcfUO6RWZ/05OwuEsuH9URnweDQ729R
z2oz8vX13iXpq6cUxh8AAeAWGke0n3FeWGke0n3CHro/cLxqm9ce8VPuZoeoz1x7xU+5llo4n1DmACM
T7CLnNRist8iR5vuPg/qjjSn0VWM8Zwy1tb7yirubmOGeYe2nWlvEq2rbVaEU6kcJ/U4lrq3sYfcVQc6llpb
EAADzd7L3un3l8UNI73T7y+LDbw3zIACTQAAAAAAAAAAAAAMPtDo1zb387qhCU6U3vZjziz7bb
WX1ulC4pRqpcOPos0U9obGlfVLStPccHjefJny5eiXNjutO1axxakk/6AedL2itdRmqXGIVfKMUT7ih2p0m
NpUd4qrk603607yKmnGK1iEbNtw6ZdG/pngafbKMnptBtccpQKvQ9n4anbeUSUJU3GWMKOS32n1Cp
p9jStqEmpzjhyXNJHHZK9tqGnVKdavCnJSziUkjntpQlUhb3UPSp43W1/T8wlej6LaXlt5Re3cYOT4Q3ln8S
NqVr5lvadWxulUXoli+K+jOmj2Ok3tDF1cyo1lzTkkn4k2ekaFG4p0PLpSnN4W7JNAWd9d+Xbj1Ljk50+k
+pk9H06rqI10FOW7FLem+xGu1Kzp2GzFe3pScoRi8Nmc2X1Glp9/Lp3uwqR3d7seQJup7KK1s5V7atKcq
ay4tYyj1shqdTp5WNWTIBx3oZ6voW+r61ZUdPqqnXp1Zzi1GMJJ8TO7I0J1dXdVl0acW2/qBuij2n0qvqV
Ck7Zpypt+i3zzj/QvCBq2rUNKpRnWTk5vEYrrAjbNaZW02znG4a36ks7qfluCFpep0dUt3Vc05WHiUXzRNA
qdp/3FX/AAKYj3u5+xfmXe0/wC4q/4FjsR73c/YvzA2IAAAAAAPj5Gere2n9zNC+Rnq3tp/cySycT6h4A
BGMPq9Zd58Pq9Zd4WPa/Xu/8AhM++bNAvYf4TPvmyy08R+gAEZQstH9ar3IrSy0f1qvchD10fuFmVer+
vDuLQq9X9eHcWWzX+FcADI859XNF1dPo7GW71LBSLmi7uF01i93rjIHUNGj82UhP0qTVaUeprJALDSop
fnPqSwSPbjRzzw56pFK5T7UQyXqc1K6wv4URBKav3IdLf3in9yOZ0t/eKf3IOK+2gXI+VPZy7mfVyPIT2cu4
6fVn0zsvWfefD7L1n3nw5fjkLPSY4hOXaysLPSLdnD8RD20Ptz1WTdWMepiUJOFeEl1MmarB9JGeOG
MES3g6leEUusv7W+eourm09bTT7CgL66koWs39MFCJdcT7gABGZbaT7CX3HDVvbQ7jvpPsJfccNW9tD
uL+mu34UAAEZAASAAWek8qh41b2sO49aTyqHnVvaw7i/prn8CvABGQAAGgtvd4dxX6t7Wn3MsLb3eHc
V+re1p9zK36v41eACMAAAlfS/dX9xD1L3t9xM0v3V95D1L3t9yL+mvU/DCIACMgdLf3iHeczb+8Q7wtf
bQlLqfvb7i6KXU/e33Flu4j5RAARgSLBZvIFjqLatJY7UVlrPo7mEnyyW95Sda2IGPF80WGvS86cxCiLrT3m0
jn6lNuy3t3HHlgvbWn0NtGL7MskOeHieZUXqxd1MdewOt1Pplicl1s5B4X+pCv3vcSKStO97iF0/qF2U2
qe9f4S5KbVPev8JzbOI+EMAEYAAAXdh7pAq733qp3IpYe6QKu996qd5ZatX8cOAAlyhlsfe4EckWPvcBD
un1C9KbU/evwLkptT96/AstnEfCGACMCXpqzdr6JkzVJNWqx1ywQLGoqd1Fvk+BZ39J1rZqPFriiten50pi
FIxtk3K0p57CjUZSkopPL6i+ox6G3jF/wriITh4xMyplYuai/wCZnl91pb9Wcu1tnjNb3IWokevU7kVxY6R
69TuQh6aP3C0Zna/t6n3P8zRMztf29T7n+ZZe/E+oeAARjD7H1l3nw+x9Zd4WF//ANX/AMJQS9Z95f8A/
V/8JQS9Z95ZaeI9Q+AAjKFjpPrzK4sdJ9eYh66P3C0flz1z7zU+5mhflz1z7zU+5llo4n1DmACMQfVzR8Pq5
oEe1/H3dfaUEvXl3I/H3dfaUEvXl3llq4j1D4ACMoWGke0n3FeWGke0n3CHro/cLxqm9ce8VPuZoWZ649
4qfcyy0cT6hzABGIJml+8/gQyZpfvP4CHppfcJorexh9xVFrq3sY/cVql3r/YAA8Hey97p95fFDZe90+8viw2
8N8yAArAAAAAAAAAAAAACh1HZW1vKsqtKpKjOTy+tFd/YyrvY8rW79n+5rwBT6Ts7babNVd5
1ay5Sa4LuRYX1ISv7aVCusxl2c0SABk5bFvpPQvMQ+sOP5mip2NPzfG0r/toKO63JcyUAMvdbG05zcra5d
OPwyjn+p7stkKNKqqjlzXdDzupYNKAKzaBKOhXKXJQwjBPaXS1WpXpVW4uMMxkup5RuNQtFfWVS3c
nFTWMor9G0K0k1qISNaVTfju4aAq4bF/tPTvMw+kOP5mjsLChp1uqNvHC62+bJIAFZrejw1ejCLqOnOm
24yxnmWYArtF0mGk20qcZ78pvMpYwWIAETVLLzhYztuk6Pf/ixnBCOPQfNFWrPyjpd+OMBmMf1LgAA
AAAAAAAfHyM9W9tP7maF8jPVvbT+5klk4n1DwACMYfV6y7z4AL/ej0GN5eqUD5sAzeupqc+AAB5BZa

P61XuRWllo/rVe5CHro/cLMq9X9eHcWhV6v68O4stmv8K4AHL5wWFheRhHoqrwupleCuqXmk5hcTsre
rLfTxnsZ9qV6NpR3KeM9SRTqTSwmz4MvbrRHｚH9nJzk5S5viz4AGeQ6W/vFP7kczpb+8U/uQWvtoFyPI
T2cu5n1cj5U9nLuZ0+rPpnZes+8+H2XrPvPhy+Tldbes6FVTX4o5ALE4nML2NWhdU8ZTT6mIU6FvmUVG
PayiTafA+uTlzbYy0df9zHIMv7tVmoU/VXX2kIAPC1ptOZAAHK20n2EvuOGre2h3HfSfYS+44at7aHcX9Nd
vwoAAlyAAAstKkkqmXg8aq06kMPPAgAZevU/xyAADyAABoLb3eHcV+re1p9zLC293h3Ffq/tafcyt+r+NX
gAjAAAC20ySVthtLiQ9Sad02uwigPW2pmkVAAHkHS394h3nM6W/vEO8LX20JS6n72+4uil1P3t9xZbul+
UQAЕYAs7TUIqKhW4Y5SKwB3S80nML3p7b1t6HeRLy/Tg6dHjnnlrQMvS2vMxiAAB4BK073uJFJWne9x
EO9P6hdINqnvX+EuSm1T3r/CWWziPhDABGAAAFtZXNGnbRjOok11FddSU7mcovKb5nIB6W1JtWKgA
DzCRY+9wl5lsfe4CHdPqF6U2p+9fgXJTan71+BzB0I+EMAEYBPDyWlrqEXBQrPEI19pVgO6Xmk5he9Nb
R9LehntIV5fqpF06XLrZXgZelteZjEAADwCx0j16nciuLHSPXqdyEPXR+4WjM7X9vU+5/maJmdr+3qfc/zLL
34n1DwACMYfY8JLvpAuvK6HQ46W0d0ppes+8+APS+pN8ZAAHmFjpPrzK4sdJ9eYh66P3C0flz1z7zU+
5mhflz1z7zU+5ll04n1DmACMQFzQAF1G6odCl0sc7uCml6z7z4A9L6k39gADzCw0j2k+4ryw0j2k+4Q9d
H7hasz1x7xU+5mhflz1z7xU+5ll04n1DmACMQStPqQpXG9OSisc2RQHVbcs5WOpV6VWIFU5qTT6iuAB
e83nMgADI3sve6feXxQ2XvdPvL4sNvDfMgAK1AAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAA+PkZ6t7af3M0T5GfuYShXmpLHEksvE+ocgArIAAAAAAAAACy0f1qvcitLPSISSqSa4PCQ
h7aMf7hZFXq/rw7i0K3VoSe5JLgiy16/mkqwAEfOAAAAAAAAdpb+8U/uRzO1rCU7mG6s4kmHVY8r5cj
5U9nLuPS5HyazCSXWjp9SftOS9Z958PVSLjUkpLdytI8qYmJAAEAAAAAAAW2k+wl9xw1b20O4k6XC
UbduSxl8Dhq0Jb8JY4YxkrZaJ6KuABGMAAAAAAAAPqTbSSy2DC/tvd4dxX6v7Wn3MsаEXGjBPmkQN
WhJuE8cF1NlfQ1Y/5q0AEfPAAAAAAAAdpb+8Q7zmdrSEp3MFFZw+IdVjzC/KXU/e33F0VOOp0J9N0i3Frq
LLbxETNEAHrcn8L8B0c/hfgRhxLyD10c/hfgNyfwy8AYi5B63J/DLwG5P4ZeAMS8g9bk/hl4Dcn8MvAGJeS
Vp3vcSPuS+GXgTNNoTddVGmox7Q704nmhbINqnvX+EuSr1ShN1VUjFtYxwLLZrxM08K4Hrcn8MvAbk/
hl4EYMS8g9bk/hl4Dcn8MvAGJeQetyfwy8BuT+GXgDEvIPW5P4ZeA3J/DLwBiXkkWPvcDjuT+GXgS9Oo
TdwpulUY9bEO9OJ5oXBTan71+BclXqdvN1VUiJNYeDqWzXiZp4VwPXrZ+CXgOjn8EvA5YMS8g9dHP4
JeA6OfwS8AYi5B66OfwS8B0c/gl4AxLyD10c/gl4Do5/BLwBiXksdI9ep3IgdHU+CXgWml0J04ynNY3uSEP
bRrPPCeZ01/b1Puf5miZRXVCpTrz9FtNtposvb1YmYhHB63J/DLwPm5P4ZeBGPEvgPu5P4ZeB93J/DLwBi
XkHrcn8MvAbk/hl4AxLyD1uT+CXgNyfwy8AmJeSx0n1pkDcn8MvAs9LozhGUprGeWRD20YnnWHUZ65
95qfczQlHe0KkLmb3W1J5TRZaOliZrCMD1uT+CXgNyfwy8CMWjeQetyfwy8BuT+GXgDEvIPW5P4ZeA3J
/DLwBiXkHrcn8MvAbk/hl4AxLyWGke0n3EHcn8MvAstLozhvTksJ8FkQ9dGJ54WLM9c+8VPuZoSkvLepC
4k1FuMnlNFo4mJmlwig9dHP4JeA6OfwS8CMWjeQeujqfBLwHR1PgI4AxLyD10dT4JeA6OfwS8AYi5B6
6OfwS8B0c/gl4AxLrZe90+8vimsLepK4jNxajHjllyWG7h4mK+QAFaAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA
DxOICfrwT70ewEmIn25eTUflx8B5NR+XHwOoCctdnLyaj8uPgPJq
Py4+B1bwssyes7UVIXErfT0vReHN8cv6A5a7NP5NR+XHwHk1H5cfAw8df1i2kqlWTcX1TjwZrNG1elqttv
xW7UjwnDPIHLXZM8mo/Lj4Dyaj8uPgDQDIrs5eTUflx8DpGKisRWEfQFlj0HyUVJYksr6n0BXLyaj8uPgPJ
qPy4+B1Ac8tdnLyaj8uPgPJqPy4+B1IGtV69tptSra+0WMcMg5a7JXk1H5cfAeTUflx8DD1NotXpYVSi32x
Ndod1VvNlpVqzzOXNg5a7JXk1H5cfAeTUflx8DqActdnLyaj8uPgeoUoQ9SKXcj2AsViP0AAK8So05vMoJ
v6o8+TUflx8DqA55Y2cvJqPy4+A8mo/Lj4HU4X1epb2lSrRpOrOK4RXWDlrs9eTUflx8B5NR+XHwMO6+t
6xWludJhPio8FE5VK+r6PXj0s6kG+KUnlSBy12b3yaj8uPgPJqPy4+BF0bUfOenwrtbs+Ul9SeDlrs5eTUflx8
B5NR+XHwOoBy12EklhHyUVJYkk19T6A6cvJqPy4+A8mo/Lj4HUBzy12cvJqPy4+A8mo/Lj4HUA5a7OXk
1H5cfAeTUflx8DKalaa5PUa8rfpeic24YfDGSSvJaxYqluatWnvcsy5g5a7N95NR+XHwHk1H5cfAeZwYU724
r1K1ec50VHCcn1mqBy12cvJqPy4+B9jQpReYwin3HQA5Y2D5KKksSSa+p9AdOXk1H5cfAeTUflx8DqA55
a7OXk1H5cfAeTUflx8DqActdnLyaj8uPgPJqPy4+BQ7RbQTsqkbaza6Xnj88fQn6FK+rWft38ImfGMcYwg
ctdk/wAmo/Lj4Dyaj8uPgDQDIrs5eTUflx8D1CICn6kUu5HsAisR+gAB0+bq7EMl5R9ATD5ursQ3V2l+mW
2wva9tVt4UKsqeY5e6wYajdXYN1dhg6NDX69GNWk60oSWU88y42et9VpX0nf8ASdHu8N59YMNJursQ
3V2l+gGHzdXYj6AFAAB83V2lbq7EfQEW+bq7EN1diPpTa/ra0umoUkpV5rgn1LtBhcbq7EN1diMDSWt6u
3UhKpKPbnCLDSbPW6GpUqdWdSFHnJt5TQMNdursQ3V2l+gLh83V2l+gAAAAAAAGc2xua1tRtxQqSpu
UpZ3evkUdtT126oqrQlWnB8mmBvwnN3c6rZVVTuK1WE2s4bNfsxOvU0eM7iUpSInOXYBbgAAAAB8
xk+gD5ursQ3V2l+gGHzdXYhursR9bSTb4JGFv8AXL7Ur3oLOUoQct2MY833gw3O6uxDdXYjC3On63p9H
ymVWWFxbjPLRc7M65Vv3K2ueNWKypdqBhod1diG6uxH0Aw+bq7EfQAAayAB83V2lbq7EfQEW+bq7E

AAADxW9hU+1/k
ezxW9hU+1/kB+e6R/eKj/O/U/RT860j+8VH+d+p+igAAAAAAA
AA
AA
AAAAAAAAACNc6haWk1G4rxpyaykySZ7arSHeUFd0VmrsXFdqAzHTU/PvTb66Ppc730N/ba
jaXc3C3rxqSSzhH5hh5xjibbzXSHa0fK66aqVF6K7EBogAAAAAAA
dKtcQhU3X6L7iaZbazR5VU7+gm5RWKfp2gUGmVqdLXKVWpJRqrlyfZk/QLW+tbzeVtWjUceeOo/L0
m3hLLZvdmtl832rq1V+2qrLXwrsAuwAAAAAAA
AAAAAAA
AAAAAAA+IAFG9mLR6n5XvPczvdFjhkvEsLAAAAAAA
PoApLbZm1t9Sd2pOUU8xptcEy7AAAAAAA
AAAAAAAAR615Roy3Zy
49iRI6jPV23Xm38TJLx1tSaR4WvnK37ZeA85W/bLwKYEyzdxd
c+rftl4Dzlb9svApgMncXXPnK37ZeA85
W/bLwKYDJ3F1z5yt+2XgPOVV2y8CmAydxdc+c7ftl4EmlvhWhvQeUZ0stIbzUXUsFiXppa1rWxKzPFWrCj
DeqPCPZWau3vQXUV76luWuUjzb9svAecrftl4FKDnLJ3F115yt+2XgPOVv2y8CmBcncXXPnK37ZeB985
W/bLwKUEydxddcrftl4Dzlb9svApQMncXXXnK37ZeB0o3IGvLdhLj2MoTrbtq4ptfEi5WvEWz5aABcjzUe
Kcmuwrbnw4VL6hSm4yk212l8ec7ftl4FPJ5k2+0+HOWKeItlc+c7ftl4Dznb9svApgMp3F1z5zt+2XgPOdv2
y8CmAydxdc+c7ftl4Dznb9svApgMncXXPnO37ZeB985W/bLwKUDJ3F2jhONSKIF5TPNatChHeqPCImIN
uhLPJSOOOrP9rBdWCtM6kxp8yT5zt+2XgPOdv2y8CmBMs3cXXPnO37ZeA852/bLwKYDJ3N1z5zt+2XgP
Odv2y8CmAydxdc+c7ftl4Dznb9svApgMncXXPnO37ZeB6hqFvOSipNN9qKQFyRxNmIPFWrCjDem8i+W
7boQb7EV+rt79NdWGVqvflpzJhnO37ZeA85W/bLwKYHOWXuLrnzb9svAecrftl4FMBk7i6585W/bLwH
nK37ZeBTaZO4uuOVv2y8B5yt+2XgUwGTuLrnzb9svA60bujXliEuPYyhOts3G4g12jK14i2fLQEa6vIW2E
1mT6iSUup+9vuLLTrXmIcwedo/KfiPOy+U/ErAMsfXvus/Oy+U/Eedl8p+jWAmTr3WfnZfKfiPOy+U/ErA
MnXvus/Oy+U/Eedl8p+jWAuTr33WfnZfKfiSbW9hctpLdl2MoyVp3vcRi3TWvNoiV2R7q7hbJZ4yfUiQU+
qP/AKUvtK06t5pXMO3nZfKfiPOy+U/ErATLH177rPzsvlpXhNZfKfiVgJk6991n52Xyn4jzsvlpXkwFyde+6z8
7L5T8R52Xyn4IYCZOvfdZ+dl8p+Jltb2FxLdw4y7GUhJsPe4Fy7prXm0RK8I91dwtsZWZPqJBTap71+Alp1b
zSuYd/O0flPxHnaPyn4IYBlj6991n52Xyn4jztH5T8SsBMnXvus/O0flPxHnaPyn4IYBk6991n52j8p+i87R+U
/ErAMnXvus/O0flPxJltcwulb0OrmigLHR/XqdyLI6aWta1sStCfcajCjUcffya5k1mdr+3qfc/zEvbX1JpHhYe
dl8p+i87L5T8SsBMsvXvus/Oq+U/EedV8p+jWAZOvfdZ+dV8p+i86r5T8SsAyde+6z86r5T8R51Xyn4IYBk6
991n52Xyn4ky2uIXFPej1c12FAWOketNFy9dLWta2JWhDub+FCe4ouTXPBMflz9z7zU+5h6615pHhO87
L5T8R52Xyn4IYCZZevfdZ+dl8p+i87L5T8SsBcnXvus/Oy+U/Eedl8p+jWAZOvfdZ+dl8p+i87L5T8SsBMnXv
us/Oy+U/EmW1zC4hmPBrmigLDSPaz7i5emlrWm2JWpDudQhQnuKLk1zJhnrnjc1M/Ew9ta80jwn+dl8
p+i87L5T8SsBMsvXvus/Oy+U/Eedl8p+jWAZOvfdZ+dl8p+i87L5T8SsBcnXvus/Oy+U/Eedl8p+jWAmTr3
3XVtfQry3cbswulIDZe90+8viw16N5vXyAAr2AAAAAAAAAA8clh1ml3sKcZerJPuYH0AAA
AAAAAAA
MYAdoWtacVKMG0+sLETpxB3dncJzdNnAExMewABAsdH9ar3lrix0f1qvchD10fuFoVer+vDuLqq9Xe
HcWWzX+JvwAOXzgA+pOTwllHwEiNjcNZ6NrvOVSUpPE4tB1NbR5mHgAByHSh7en9yOZ0oe3p/cgtfb
QrkjzU9nLuPS5181PZy7jp9WfTOy9Z958PsvWfefDl8mfYAAAOtK2q1eMINrtPc7KvBZcMr6B1FLT5wjgPg
8MByAAC20r2EvuOGre2h3HfSvYS+44at7aHcX9NlvwoAAIxgB6p051ZbsFl9gljLyDv5FcfcKZzq0Z0mlU4t
hZrMe4eAAEAABoLb3eHcV+r+0h3MsLb3eHcV+r+0h3Mst+r+NXgAjAAHuRnVbVOLjsCxGfTwDv5FcfcK
ZyqU5Upbs1h9gWazHuHkAByHS394h3nM6W/vEO8LX20JS6n72+4ui1P3t9xZbul+EQAEYAA729pUulu
UGsJ44hYiZnEOAJvmyt2x8SJUg6c3B80FtS1fbYACOQk6d75EjEnTvfllh3p/ULwpU97/wAKLkptU97/AM
KLLZxHwhgA5YAA729pUultwa4PHEqxEzOlcatfNibtj4kWrTdKo4S5rsC2pavt4AAchJsPe4EYkWHvcBDu
n1C9KbU/evwLkptT96/AstnEfCGACMAASaFIUr09+Djj6ha1m3iEYEzzZW7Y+JEkt2TT6gtqTX2+AAjkLHSP
XqdyK4sdH9ep3lsPXR+4Wj5Gdre3qfc/zNE+Rna3t6n3P8yy9+J9Q8AAjGAeqIYvatNTi44f1DqtZt6RQTH
ptZjvMeH1lb4PALVmvsABHIWOketMrix0n1plh7aP3C0flz9z7zU+5mgflz9z7zU+5ll78T6hyABGIAJNKxq

1qanFxw+1ha1m3pGBMem1km8x8SI1htdgW1Zr7fAAHIWGke1n3FeWGke1n3CHro/cLVmeuPeKn3M
0L5GeuPeKn3Flo4n1DmACMQD1Tg6k1Bc2S/Nlbtj4h1WIrekIHe4tKlvFSm1hvHA4BJiasiQABHey97p95f
FDZe90+8viw28N8yAArAAAAAAAAAAAAEDWdQWm6fOulmfKK+pPKvaLT56jpjp0uNSD3ortAy
dvDVderTcKs5Jc8yxFHapomtWS6SIKbx1058SLp2p3miVpwVNyb9KE0X9pthbVGo3NGVL/mTygLbTKIW
lpdGd/VSqNZbm0u4zdtvHfz1iFGdeHQOruv0Vy7y/100jremxjbXEYxk1JTxkwlgzdXUo2inhynub2AP0yn
WpVc9FVhPHPdkmfalsFKO9UnGEV1yeEVGh6NLR+mnUuFVUI1RxgzWo3lzrmreTwnu097dis8Eu1gbK
Oq2M57kbqln7iYmpJNNNPrRj6+ycIWzISu06qWcPkxspqlaF47CvJyi87uX6rXUBqat9a0anR1a9OE+xywz
zW1C0t8KrcU4t9TlxMTtV+/J4+FEux2Xr39tG5ubjcdRZimsvAGvoXNC5jmhVhUX/ACvJ1PzutTvNntSSjPiv
STXKaN7ZXMBuzpXEeCqRTwB3OCvbaVfoVxpup8O8snaSbi0nhtczE0Nn9Rjq6k4tRU951c9QG3IGs3F3b
WMp2VJ1Knuju/XHWTwBgaOn6zquazdRxzznLHgjj0+p6HeKFSc4SXHdcxkj9BnKFGm5SajCKy31lwWu3
3njVYq2i5KPoQx/EBt9Pule2NK4SvxzgkkTS7V2em0KEvWhHj3ksAAAAAAPkZ2t7af3M0T5Gdre2n9zJLJ
xPqHgAEYwsLfUI0qUYODbXArz6vWXeHdLTwfDQN71LPajPsv17BfaUDLL34j9PgAlyhY6P61XuRXFjo/rV
e5CHro/cLQq9X9eHcWhV6v6804stmv8K4AEfOC2sKEaVHpZr0ms57EV5ourt7ljLd4cBD30YjzbZFqapJ
TapwTiu0k05076g01x612FKT9Kk+nkuporrT1bWti3pDrU3Rqyg+o8E3VElcp9qIRHjevLaYDpQ9vT+5HM
6UPb0/uQc19tCuSPNT2cu49LkjzU9nLuOn1Z9M7L1n3nw+y9Z958OXYzCTY0OnrYfqriyMWmkpdHN9e
RD10q81ohJr16drTTa+iSOdVfwrz3GnFvI9SJqn08V1JESi3GtFrnkr2tTF8R6T9StVu9NBcuZWI/cJStpp9h
QCXnr1iLzgABHgttJ9hL7jhq3todox30n2EvuOGre2h3F/TZb8KAACMYdrWuretvn8MHEBYnE5heWt0rn
exFrd7SFq3tYdx70n1ah41b2s04v6a7Wm2jmUAAEywAAaC293h3Ffq3tldzLC293h3Ffq3tldzLLfq/jV4AI
wBlS7IW0pNxbyRwFi01nML62uFcU99RxxwVmpe9y7kTNL92feQ9S97I3lrXqzM6UTKIACMYdLf3iHeczp
b+8Q7wtfbQlqfvb7i6KXU/e33Flu4j4RAARgCrBxk7eLjGKabzxI4CxM1nML+2qOtQjNrDZTXfvVTvLax9z
h3FTd+9VO8stWtOaQ4gAjlEnTvfIkYk6d75EQ70/qF4U2qe9f4UXJTap71/hRZbOI+EMAELYakW13K2i1G
KeXniRwFrMxOYX1tVdahGclhsqL33qfeWmn+5wKu997n3llq1pzpxLgACMgSLD3uBHJFh73AQ7p9QvS
m1P3r8C5KbU/evwLLzxHwhgAjAEqhftoUICMU0iKA6raazmGhpvc6UZPm1koavtZd5eUPd4faijq+1l3llp
1/mHgAEZAsdl9ep3lrix0j16nchD10fuFo+Rna3t6n3P8zRPkZ2t7ep9z/MsvfifUPAAIxhLo39SICNNRWFWl
h9j6y7w6raaz4aBvNFvtiZ+XrPvL//AKv/AISgl6z7yy0cR+nwAEZQsdJ9aoVxY6T61Q9dH7haPkZ+595qfc
zQPkZ+595qfcyy0cT6hyABGIJdG/nShGCimkRD6uaDqtpPhoM71He7Y5M/P15d5fx93X2IBP133llo4j9P
gAlyhYaR7WfcV5YaR7Wfcleuj9wtXyM9ce8VPuNCzPXHvFT7iy0cT6hzABGJ6pTdOoprmiOs72detuSiksF
STdL95fcle2jaYtEQkat7CP3FUWuq+wj9xVCV1/sAAeDvZe90+8vihsve6feXxYbeG+ZAAVqAAAAAAA
AAAAAACPe3tGwtnXrtqCwuHMkELVNNo6pbDWcopPKcXyYHhXGl6ISTIOjVi+qf8AuZraaw021oxqWc
oxquWHCLysHStsbcRlmhcwa6srDPVvsbVINO5uYqPWoriBK2KdR2NdSzuKfo57jPW8o0dooyqPdjGvlt9
XE39naUbK2jQoR3YR/qUer7LxvbmVxbVVtNpJKLBgXsp069KpTp1lyk4tcHk/OFQhDU3Ru5ypRU2pSS4
rjzNdoWg3GmXMq1W4U1KO64o76vs9b6lLpYvoq3xLr7wKtbN6fKn0i1P0cZzw/1O+jaFYq4jd2l5Kt0cm
sbuOJC/sdd7275VT3e5l9oui09jpy3akpzn6zb4eAGU2p4a5N/8qNzPgp2txptGSqwjKMEpRbxhmT2pW
ddmvoic9kZ1aUKttcKKnFNxkgIm1d9SvdSiqD3o04brkut5NbolGVDSLae1iSgsrsKrS9k6drWjWu6irSi8qK
XDjpFwAN4WWVMNo7Cpe+SqUt7OFLHotlrJKUXF8msGXpbJShqSrSrp0VLexjiBqTxWq06FKVSrJRhFZb
Z71Os2E9S0+VtTqKDk08v6AZHXNdqanW8nt242+cJZxdv5caDYWGnU1WrXFkvw18Xq/Qg/2LuP+Lpf5
WP7F3H/ABdL/KwNfsq060N6lOM48sp5PZXaHps9KsHb1Kkajc3LMVjnq/QsQAAAAA+Rna3tp/czRPkZ
2t7af3MksnE+oeAARjD6vWXefD6vWXeFj2v17BfaUDLtV6PQ46WGcdpSMstGvMTh8ABGYLHR/Wq9yK
4sdH9ar3IQ9dH7haFXq/rw7i0KvV/Xh3Fls1/hXAAj5wuZd1V09i93jvRyUhNsb1UV0dT1ep9gh7aNoiZif2
hvg+JYaVTe9Ko1hYwjvKIZ1Xvtwb+481r2jQp7IHdfVjkV6V04pPNMompTU7ppfw8ClfZNyk23ls+EZ7Tz
WmQ6UPb0/uRzOID3in9yCV9w0K5I81PZy7j0uSPNT2cu46fVn0zsvWfefD7L1n3nw5fjkLLSzr04N8eZW
nSjVlRqKceoQ707ctspuq05b0ZpcMYZEtabqXEULwzxLWldUK8PSkl2qR66S3optShHuK0zp1tbmyXk1TtZ
t9mEURKvbviW7HhBcvqRSS8da8Wt4AAHittJ9hL7jhq3todox30n2EvuOGre2h3F/TZb8KAACMYAALLSf
VqHjVvaw7j7plSFNVN+cY57Xg8anUhUqQcJKXDqZf01TMdFCABGUAGgtvd4dxX6t7SHcywtvd4dxX6t
7SHcyy36v41eACMAAALfS/dn3kPUve5dyJOnVacLfE5xi88myJqe4zum4yUljmi/pqvMdKEYAEZQ6W/vE
O85nS394h3ha+2hKXU/e33F0Uup+9vuLLdxHwiAAjAAC8sfIdxU3fvVTvJ1re0aVvCE5PK+hX3E1UrzIF
8G+BZadWOTSIhzABGYJOne+RlxJ073yld6f1C8KbVPev8ACi5KbVPev8KLLZxHwhgAjAAAC70/3OBV3vvU
+8mWI7RpW8YTk1jfQg3M41K8pReU2Vp1LROnEOQAizBlsPe4EckWHvcBDun1C9KbU/evwLkptT96/A

stnEfCGACMAAAL+h7vD7UUUVX2su8tKV/QhRjFyeUscmVdRqU5Ncmyy061omsYeQARmCx0j16nciuLHS
PXqdyEPXR+4Wj5Gdre3qfc/zNE+Rna3t6n3P8yy9+J9Q8AAjGH2PrLvPh9XBrlF/wD9X/wIBL1n3lr5fQ6H
d3nnGOTKp8W2WWnXtE4w+AAjMFjpPrVCuLHSfWqCHro/cLR8jP3PvNT7maB8jP3PvNT7mWWjfUO
QAlx9XNHw+rmBfx93X2lBP133lsr+gqW7vPOMcmVMuMm12llp17RMRh8ABGYLDSPaz7ivLDSPaz7h
D10fuFqzPXHvFT7jQsz1x7xU+4stHE+ocwARiCbpfvL7iESbGtCjWcpvCwld6c4tEpmrewj9xVE+/uqVelGN
NttPPIgCXetMTbMAADxd7L3un3l8UNI73T7y+LDbw3zIACTQAAAAAAAAAAAAAAA5Gdre2n9zNE+Rna3tp/cySyc
T6h4ABGMAAAAAAAALHR/Wq9yK4sdH9ar3IQ9dH7haFXq/rw7i0KvV/Xh3Fls1/hXAAj5wACBx7QAU
AAOID29P7kcqpQ9vT+5Ba+2hXJHmp7OXcelyR5qezi3HT6s+mdl6z7z4fZes+8+HL5MgAADIAAAAAABba
T7CX3HDVvbQ7jvpPsJfccNW9tDuL+my34UAAEYwAAAAAAAAAaC293h3Ffq3tIdzLC293h3Ffq3tIdzLLf
q/jV4AlwAAAAAAAAB0t/eId5zOlv7xDvC19tCUup+9vuLopds97fcWW7iPhEABGAAAAAAAOne+Rlx
J073ylh3T6heFNqnvX+FFyU2qe9f4UWW/ziPhDABGAAAAAAAJFh73Ajkiw97gld0+oXpTan71+BclNqf
vX4Fls4j4QwARgAAAAAAAACx0j16nciuLHSPXqdyEPXR+4Wj5Gdre3qfc/zNE+Rna3t6n3P8yy9+J9Q8AA
jGAAAAAAAfpPrTK4sdJ9aYh66P3C0flz9z7zU+5mgflz9z7zU+5llo4n1DkACMQAAAAAAWGke1n3F
eWGke1n3Chro/cLV8jPXHvFT7jQ9Rnrj3ip9xZaOJ9Q5gAjEAAAAAAA72XvdPvL4obL3un3l8WG3hvm
QAFAgAAAAAAAACcc
xUpU6nrwUu9EeOrp88M6C/8kofKh4DySh8qHgMPDtrbqAF/5JQ+VDwHkID5UPAY0TtrbqAF/5JQ+VDw
HkID5UPAY02tu0AX/AJJQ+VDwHkID5UPAY02tu0Cz0iDSqTa4PCRM8kofKh4HWKUVhLCGHpp6E1tmZ
fSt1aEnuTS4LgWR8aUhrKK971564ZsF/5JQ+VDwHkID5UPAmGXtp3UAL/ySh8qHgPJHyoeAwdtO6gbf
8AkID5UPAeSUPlQ8Bg7ad1AC/8kofKh4DySh8qHgMHbTuoDtawc7mmorPFNIz5JQ+VDwPcKVOon6kFH
uQwteHmj8y9rkfJrMJLr9BWtnKkXCpKMIhpnk0M6FKo8zhGT+qPHkdv8qHgTDHPDTnxKhBfeR2/yoEA
8jt/IQ8Bh02tu0QX3kdv8qHgPI7f5UPAY02tu0QX3kdv8qHgPI7f5UPAY02tu0QX3kdv8qHgfVaUE8qlH
wGF7ad3DS4SjbtyWMvgcNWg9+E8cMYLRJYR8IFSWJLKYaj0805WbBf+SUPlQ8B5JQ+VDwGGbtrbqAF
/wCSUPIQ8B5JQ+VDwGF7a26gBf8AkID5UPAeSUPlQ8Bg7a26gBf+SUPlQ8B5JQ+VDwGDtrbqA+pOTwII
svvJKHyoeB9hb0qbzCnFP6IYO2nd9oxcaME+aRA1aEm4TS4LKZZnyUVJNSWUw03pzV5WbBfeSUPlQ8B
5JQ+VDwGGbtp3UIL7ySh8qHgPJHyoeAwdtO6hbfeSUPlQ8B5JQ+VDwGDtp3UIL7ySh8qHgPJHyoeA
wdtO6h01pBzuYJLk8lx5JQ+VDwOIOLcmsQior6IYWvDzE5mXsqtToT6XpEsxa8C1BWm9OeMM1h9gw+
w0mF2DC7CYZu2/rN4fYMPsNJhdgiF2DB239ZvD7Bh9hpcLsR8wuxDB239ZvD7Bh9hpMLsGF2IYO2/rN
4fYTtNoTddVGsRXX2lthdiPuBh1Xh+Wc5Cr1ShN1VUim1jDLQFe96c9cM1h9jGH2GkwuxDC7ETDN239Z
vD7Bh9hpMLsQwuxDB239ZvD7GMPSzPMsQwuwYO1/rN4fYMPsNJhdgwuwYO2/rN4fYTdOoTIXVRp
qK6y3wuxH3Aw6rw/LOchV6nQm6qqRTaa446i0BXvekXjDOBkvhY3JfczRYXYMLsJhn7b+s7uT+Fjcn8LN
Fhdgwhg7b+s7uT+Fjcl8LNFhdgwuwYO2/rO7kvhY3J/CzRYXYMLsGDtv6zu5P4WWel0J04ynNY3uCyT8L
sPow7poRWc5Ciue6EVeTae622mXowiu9TT54wzWH2DD7DSYXYMLsRMPDtv6zeH2MYfYzSYXYhgdgw
dt/Wbw+wYfYaTC7BhdiGDtv6zeH2DD7DSYXYMLsGDtv6zmH2FrpdGVOMpzTW9yyTsLsPow9NPQ5Jzk
KO9oTp3EpNPdk8pl4MFd6mnzxhmsPsGH2GkwuwYXYTDw7b+s3h9gw+xmkwuxDC7Bg7b+s3h9jGH2
GkwuwYXYMHa/1m8PsGH2GkwuxDC7Bh7b+s3h9ha6XRnDenJN8Fkn4XYfRh6U0OWc5CjvLepC4k917
reUy8GCvTU0+eMM5uS+Fjcl8LNFhdh9wuwmHh239Zzcl8LG5L4WaPC7D5hdgwdt/Wd3JfCxuS+Fmiw
uwYXYMHbf1ndyXwsbkvhZosLsPuF2DB239U1hbzlcRnutRjxyy5AK99PTikYAAHoAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAGY1nanyavK3soxnKLxKb5Z+hd6vcO20q4q
x9aMHgxuy1nC91bNZb8acd9p9YHWntVqVGaIWUZxfVKODVaTqIHVLbpKfoyXCUX1HzV7CjdabWg6cc
qLcWlyZktk7iVHWY00/Rqpxa/wDfcBvQAAAAAAAAlOs3dWy02pXoLM44wmicfGIJYtaTX1Aw8trNRj6
0ILviavRrupfabTr1cb8ueDNbaxjG7t91Jfs+pfVI9sz+46AfqAAAAAAAABwvrh2IpUrRpyqOKyoxWWzu
GsgYOeq6zqtaSt+kUV/DSjwj+JzWqaxpddKvOov8AlqLKa/E3tOjTpB3RwjHeeXhcjlBYX9G4rUrak1KVJtyk
vr1AaXSdQjqdjC4it1vhKPYyaU2ytrO20iLqjqVR72H2FyAAAAAAAAGP1KeurUa6tIX6Lfe5iPDGSvub
7W7RKVxUrU0+W8sZN7WqKIRnUayoRcsdx+c6hqMtTv1VupSjSzhJcd2IGh2Uu7+8uKs7ipOdGMcJvlk1
BXaLxsallGFhJbkFxXX+JYgAAAAAAAACv1nU4aZZSqPDqPhCPawIG0Gv+bpRoW27Ks+Ms8kiZodxe3dp
5ReqMVP1IpY4dpltFO+pq9/O7um3Rg96cn1vsNdbarYV6sbe3rwIPGFFfQCcAAAAAAAABmdrNRubO
pQhbVXTcli46zTGJ20nvalRj8NP8AUDjsr7QVqcalN15QkspqPMt9npas76XI6q9Hu8N+OFkh2G1kbW0p

AA
AAA
AAA
AAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAA6hf0dOtZV6z4LgkutmYntXqFebdpbRUFqnls
dsLWtX0+nOlFyVOTckiq0HaK3061VtXoS4PO/H9QJ2I7S3Vxe07W4tfSm8by4Y/AuNV1WlpVKE6sJSUnh
YPNIqOm6hVjUoSpyrLlYkvEzO1dG9hcOdabdvKf7NZ5Aa7Tr2GoWcLmnFxjLOE/o8Ekxuz1rqkoW9WIV
krTe4x3urPHgT9o9dq2tVVVnwqyXpS7PogNHkGOek69Gh5S7ye+lvdH0jyWOzet1L5ztrr20OT7UBoBld
pgbzU9Qpa3WjQuKraqNRhvNrjNxp2vUaLupXFVtLeIFVHIAbUGf2Z1upf71tc8a0FIS+JGgAAyGvS1da1i
2dfo+HR7mcfj+JraW90Uek9bHHvAqNf1a40x0lb0IU3+fDkV2m7T3N5qNG2qUoRU5Yb6zVNGA0/8AvZ
D+cBvyJqt3Kx0+rcQScoLKTJZWbR/uS4+0CghtZf1c9HaxljnhNntbW3dKS8os1u/iijs1rFvpcLhXG9+0axhd
mTttBr9pqNI0FGIJyyvnyXIDU6fqFLUrPpqDxng0+aZnrPTNXhrvTVJy6PfbIPe4NE/ZSznbaXKc2s1XIjPOEU
+n3t1LaropXFV0+ImtxzeMceoDajjj9rry5t9TpRoXFSnF0k8Qk0s5Zx6DXtRt/KIVqKDwyxU93K7gNsDI7M
6zcu+8iu5ympZScuLTRAa9R1WtOIDTqjjCDw1+IF1kGlvdP1vTqLuZXScY8ZbtR8C+2b1aepWklW4
1abw32oC5BgLvU7+jrdZUrirLFRqMHJteBZ22n6zO2uqlzWrQnKnvQxU45XHHBgawGW2S1SrVrVbS6qz
nP1oub/y/qj1tbqVWjOjaWtScKj9KTg8P6IDTnK5qOjbVKi4uEW0cNKo1KGnUY1qkqlRxzKUlns/9wr/y3+
QGUjtfeyle28H3Jn3+1eof8lv8rK7Z7UqOmXc6teMnGUccDRx2tsG0lTqcfBa6XdVLywp16sdycuaJeSDq
WpQsdNldYy2luR7WzJ2z1rXKs6lKvOEE+ak4xX0A3QPz/UbnV7CUba5uKsWuMZKb9Jd5oa1xWWySrKr
NVdzO/vcefaBfgzex1zXuKVy69apVw4435N45nPbG6uLedv0FapTynnck1kC71ijcV9Nq07SW7Va4fUrd
mLK/tl1fLG4wl6sZPJ7q16q2R6ZVZ9l0Te/nj4kDZmrc31jfU6lxVILCUZObyn9ANWDHbO6lcUtXnaXlapPe
bj6cm8NFptVqUrOxjSozcKtV8HF4aQF6MpdZn9Oo6jHZ2VSFacrqt+PSPe4fj9Cuej69Vg6s7qalz3ekeQ
NiZnarUruxuKMbas6aIDLx3kbZ/WruGoqvwZSvnNxTk8uLX1PO23vVv9n6sDUadUnW0+hUqPM5QTb7
STkp/OEdN2co15LL3EortZRWTpwde3riN06VNPh6TS8EBtQY221TUDf1GNrqM5VKbxlyeeHamStsLqvQ
8mdvXnTuk87kmmsgajOAYe3p67q1sqlKrUVKKwnv43v9T3o2sXtpqcbO8nOcXLdkpvliwNRqup09Lt41qs
ZSTeMI9abfw1K0VxTi4xbxhmY2vo3iryqzm3aNpRjnk8dhz2fftVnCIUt6slbKfpR3sd/ADbAFDtVK+jaU/Iu
kxven0ec/wBAL443VVVG3IOKy11ELZ93b0qDvd7pMvG/zx9SzfEJPMFO9VrJ4dNLvJ1a8jRt41JrlkuCRX
6usXUcfCizp04VbWEZxUlurmwczac3m1q5VstUuG8xppR7iXZagriW5Nbs/wAyTU6KhRbkIGKRT2EXuv1
OKxFPLHpJm9LRGc5SrjUK1KtKEaeUmcnqtZc6aXeWtWUadOU2s4WShdaNzdKVeW7DsQnwMrNqz9Le
2unUtOmqr5kOWqVqk8UKeV9ejJvlxnP76HjHHDBD0y5o0YyjVe6285wMra1sxXP/rvb6m3UUK8N19p
ZJ5KDUK9OtXTpcUljpaxFINztKbfYWJdaOpM2mszl3ABWkAAAAAAAAAAAAAAAAAAAAAAA
AAA
AAAAAAAAAAAAAAAAAAAAAAEW/vrewpRndSxGT3eWSBVtNE1CDqfsXn+KEsYJmp6Xb6p
SjC4z6PGLT5FFPYxb37O8e79YgUV7Tp2Or7um1nUUZLdknnj2Gg2yUnp9Vjr+Lj4EzTNmbWxqqtotrVi8
U2sJflf2NHULWVCuvRfFnC0wKzZa5orRKMHVipqUlut8ctszu01KdLXZm3GM8SUI2FxQ2PhRulVFeTxC
Sklu9n4l1qGmW+o0FTul5cfVkuAztLS4VqHSx1t7jWct8v6krZ/R6FG78tt71V4rMWkus5vYyG/6N5NR7
Nz/cutL0mhpjdwo5bl60n1gYupXhbbUSRvFUhWbzsrVbKnYzquvCUXF4SfPgY6dvC62olQq53J1mnguXs
ZSdThdzUM8t3/cCBsjTIU1mpWisQjF5/Hkbciadp1DTbforeOFzbfNksCtvNdsbK7VvXqNT68LhHvLGMIKK
knIPkUmpbM0b+/dy60ob2N+Kwcl3CCPwjCPKKwgPpgNP/vZD+cb8oqGzUKOqxvlcybU9/c3P9wL0rNo
/wByXH2lmRtQs1f2VS3c3BTWN5LOAMIsrb2NaNz5bGnjpx3d947Sx1mz0WGn1ZQVKFVR9DdlxbOa2L
p/8dP/ACL/AFPUdjKKlmd5OS7NzH6gcNjK9WUrii23TUU8PqfEg6d/fBfzp/kzY6fptvp1B0reOM85Pmyvt
9m4UNW8vVzJvfctzd7f/kCk20/e1H+SvzZrrFJWNFJcNxFdrGz8NWuoV5XEqbjDdwo562/1LajT6GjCmnn
dWMgYXTuG1rx8+f5sn7S6tdR1FWdCo6UEllrm2yyt9m4UNVd8rmUm5ue5udv4nfV9Bt9VkqkpOnVsXv
pZygKLUDGqWunTrV9TlPh6ueEjpsQ/Su+6P6kqlshSSft3U6nDEVjkT9G0OGkOq415VekS5xxgDL26Uts
MSWV08v1N6UdPZuFPVvL/ACmTlvue5ucOP1yW13dUrO3Ixry3YR5gYvVIS0TaJV6axBvfXd1o96bGWu
bSSujpunCW/wAepLI+h22p1HT7+1o+TVOkrRlzS5ltlb2mmKpOOKib0n3dQF2R7/ANwr/wAt/kSDxXp
dNQnSbxvxaz2AYLZqnZ1byor5QcN3hvvHE0ytNATyo2/+d/6kD+xVP/jp/wDlr/U+/wBi6f8Ax0//AC1/qB
32pjG40ONS2kpU6c1xi8rHI5bjahbRsHb1KkYVlybw3jKLjT9Mp2WmqylPoccuSxnJUXOx1CpVcqFxKIF/
wAO7nAFdtfe291c0adCam6ae80WtSEp7GYjxapZKLaDSKOkwt405ynKeXKT6zXaLGM9Et4yScXDDT6w

M7sffULaVelXqRpueHHeeM4yctr+hd3VKnbzU1TXFrIksrnY+hUrOdC4ISi36u7nAnsdbypRjG4nGS5y3c5
A61f7I/+CyJsP7O774/qXktMjLR/N/SvG5u7+P0OWi6NHSI1VGs6vSNPjHGMAZ7ai2lYatTvaPoqbUs9kkR
51p7Q69RSTUOCx2Jc2a/VtMp6radBUk4NPKklnBF0fZ+lpNadVVXVnJYTccYQHzak/qaXpcfJluyk9yL+FF
NpmnXep2juqmpyhlvhnkam/saOoWsqFdZi+TXNMoY7HQjPHls+jfNKPP8AqBRadFQ2kpxjU6VKq0p/Fz
4lntt73b/Z+rLK32VoW1/C5pXEkoPKhu/qSNZ0GGr1ac5V5UtyOMKOcgVmtUZ1NIbWUE2obrkvph8Ss0
W0jeUN1aIK3mn7PODbUbWF0Oyjay/aQjHdeVzKO62PtqlRzt68qKf8ADjKAvgQbe4u40ZaqqlbGUnxZ92
xh0VKyp5zuxayWem7L29jcRrzqyrTi8xysEjWtEhq/R71Z0ujzjnIHXQULotsksej+rMnqXDayWPmo2tjbK
zs6dupOaprG81jJVXOzcLjVhfO5IF729ubv8AuBy2yTelQfZUPuydzRjpChKpFSU3IN9xc3tpSvradCvHMjfo
M/DY6EKylG8mop5S3f8AcDTkPUdSttNpRncyaUnhJLLZLisRSbzjrK7WdHp6vRhGVR05QeVJLIEuyvKn9b
xr28t6D/odyHpen09Ms429OTkk8uT62TAKXWPe4/aiyhVhRs4Tm8JRRzu7BXVVTC3HCxjB6uLNV6EKW+
1udeOZMM0VtW1rQrK9ed9XUd5Qh1ZLK18moQUITjnreeZG8zr5z8D1DSVCa0reHnkTy4pXUi2ZjyISur
eU5UpzWVwafWQdQhaxoZp7u/1Y2raXCpNyhxjb4vrOcdHW9mVRtdmBOZdXjUt4mHrSHLyee96ueB
7lbWVuScv24ZKVCEaHRQ9GOMcCA9HTfCq8dxVmIorEYyiXfRSqRp2yy1rrZdWtPoreEHZS4nC20+iQa
k/SmutkwRDrS05rM2kABXuAAA
AAA
AAAAAAAAAAAAAAAAAAKyOg2cdQ8sSI0u9vc+sswAAAAAAAAAAAAAAABwvbOjfW
7o3Ed6D6s4O4AprfZjTqFZVNyU2nIKT4FykksJYSAAAAAAAAAg6IpNtqbg7hS9Dlhkm2oQtbeFGnnchsL
J1AAA
AAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAA
Tw1FgdAYDTNoru2vlyuKsqtJ8JJ9S7TeUasK9GNWIJShNZTQHsGL0rUburtFCjUrzlT32t1vgbOTUU3JpJc2
wPolb1awU9x3IHP3olwnGcVKEIKL5NMD6AHwXEACJPvBcnPcl0Vls31wJNOpCrTpzjOL5OLygPTaSy
zmrijKW7GrBvsUkZ3a3U5UqELe2rJOT9Pdlx0Imh6NSnUtrqV7B1M7zpKXHuA2IB8IKMIuU2oxXNsD6G
0k2+CRD87WDnuK8o5+9HS9kn9xKLynSk013AdYVadR4hUj/R5PZidj6qhfvpVam7FQ4uT4I1dPU7Gr
U6OndUZS6kpriBLAltB7KhLdq3VKMuxzWQJQONvd29ym6FaFTHPdkmeq9xRtob1erCnHtk8AdARpah
ZwpKrK5pKD5S31hnq3vba6z5PXp1Mc1GSeAO4BEqapY0p7k7qjGXY5rgBLBGqahaUIF1LmlFSWYtzSySi
yjOKIFpxfFNdYHOEWtqNnQlu1bmICXY5rJ2o16VxDfo1I1I9sXIAdAD5KSjFuTSS5tgfQQpavp8Z7rvKOvRK
pVadaCnSnGcX1xeUB9IVpweJzjF/V4DnCMd6Uko9rfAx+2cpK+oYbXo9TLDWW1sjSabz0dPj4AaCE4TW
YSUI9Hk9Gd2MbemVctv9q+fci9r3NC3jvV6sKa7ZSSA6giUNQtLiW7RuaU5diksnWvcUbaCIxqwpfxDMn
gDoDxSrU69NVKM4zg+UovKOdW+taNUqtxThN/wykkwO4DaSy+RwpXltXqOnRuKc5rnGMk2gO4ON
e7t7ZZr1oU/ukkebe+tbl4oXFOo+yMk2BIAltBubOhPcq3NKEuxyWUBInUhT9ecY57Xg+pqSTTTT60ZDb
GvCq7adCqpRcXhxeUaLRXnR7Vv5aAnANpJtvCRxoXlvcSlGhXp1JR5qmK8AdgDLazeazVv/JLwjOIB+q4r
1vrkDUGwl1T13SoxuK1Woo557+8jT6BqktUsekqJkpb7ssdYFoAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAA
Ac7n3ap9rOhzufdqn2sD
840vT5ajVqUoPE4wco/V9ha7PavPTrp2V42qbljj/Azxs+9pfy3+Za7T6J5TB3ltD9rFenFfxlCl0Vp7UQa50
bj21eoVql9HT6MnGKxvJP1mys2b/AH7b57X+RP2rs6tvqkb6EW4Sw89Sa/8AgCVT2PhK0TIXI07WeXBE
TZ2+uLHVnYVpNwlJwab9Vosae11orNOUJ9Ko43cc2VWgW1bUtbd7KOIKbnJ9WX1Abky21+p1acqdlQk
47yzNrm+xGpMltjYVenp3tOLfLdljq7APtIsjCtZxqXFeSqzjvLC5ZlOmXNxo2t+SVJuVPf3ZLq7y0sdrLWNjC
NxCaqwihLngqLGnW1vaDyhQap7+9J9SQHranTo2d500ajk68pSafVxLbZ/QoU429/00nJrO7jgR9t/Wtu
5I9of7mtvs/UCeYraK/r3+rKwoyagpKCS62zamG1+2rabrivYwbpuanF9WewCfPY6CtfQuH0+M8VwyTtP
0+60/RLmnndVI NunJqPw8H1nCe19orXejCbq49THWSbHVJ6po11UqUHTcacInqlwfDI6Np89SvOghN04
4zJ/Qma5oPmmnTr0qzlfy3eKw0zvsX+8K32Frtn+54fzl+TAkaDXlqGhRVeTcsODf05fkU70PSqMpK81FdJ
nknyOulSrR2QruhnPS5c8FVoa0qUqr1ST3/AOHOfx5AcozWl6zDyK56WmpL0iyafUaLbCW/pNGXxST
MzfytJaonYQcaG8IHPX9TSbWfuW371+SAq9D0CWq2rrVq7hTT3YpLJFr0q2g6zGMKre4001wyjU7JfuO

H3yKDa798r7UBcbValUttPo06Ut2Vfm12cP8AUrtH2YjfWcbm5qyW/wAypdhZbR6ZUvtMo1KK3qlGOD
3tTS/OKnTNpamnWytbig5bnBdTQFfrWn1NMvl0JVHUjh jeg32Gj1fUp2Gz9vGi8VKsVHPYsGa1i+r6jdxr1
6bpXXXXQWOrJotb0+pd7P21SjHeLrim114wBVaXYaZc2/TajfbtSWfR3sNd5zoXK0fWYqzuemoNrOHwfau
NKuNIjR6PUbeXSRfrPEsbCWhXmoq2o2k1njGbb4sDXJprKIOr2avtPnRdXoutS7CakklyRntsp1o6fSVPK
puT38f0Ar56Lo0KTjLUI0yXPPDPccNlrqpb6x5Mqm9SnlfR45M9aZ5hjpynd5dwvWTzxf0IugOL2ipuEXGL
k3FdiAnbae/UPtLDWv7oUv5dP9Cv209+ofaWGtf3Qpfy6f6Ac9IKyt9Dua0uUjt/0RUWVC42k1Obr1XGK
W8/ouxFtstQ8p0G6o/HNx/oip0q8qbP6nUhdU5brW7JfqgJotbPea7ZXVrWk1FpST5r6kyhVnrmzFanV9
KtR5Ptxy/Uj6/tDQv7LyW0j77W9JotdlrCdppjdaLUqzzh9gEPY69Xkta3qSx0Xpr6LrM7qVxUvdQuLpJuKI
z7Owkaiquj6vcwpc1E1/hl/8lzuokKrs7X34/tLhNxfyly/UCY9XT2W8qbzUcNz/ABcmV2zcPldLu9TqRzJp7
uetL/czyq150Y2HHd6RNLSfL9TfebV5i8hhIL6Ld/F/7gY+z6LV76pV1S86OC44b4v6I9apb2enzp19Kvd959
XPFFUj2atrG9nR1S2IKPLsaLKrdbPRnBUrWdTLSb48ALK91qp/ZincweK1X0G+x9ZV6Hs/51oSurmrJQbw
c2W2t6XCegxhY03u03vqPXjrK3Z7aCjp9o7W6jJRi24ySArtd0qeIXeafSodKSzFvqNton7mtf5aMZtDq0dV
uIOIBxpU1hN9bNnon7mtP5aAkXdf3FrVoqW65wcU+zKKHZ/QLrTr+de4nHd3XFKLzkv7qsra2q1mm1Ti
5YXXgpNE2jnqd7K3qUFdg5RcX+YggPjS5vxPpldptfnSqTsbVuMlwnPs+iA8bU6vCvFWFt6bcvTa7ewtdm
dOqafpuKyxUqvf7DL6Jd6dY1PKLuE6lb+HC4RNfpes2+qTnGhGacFI7wFiAAAAAAAAAAAAAAAAAAAAAAA
AAfJRUouMIIP
gz6AldppVjZVXUtreNObWMpsmYAAh09JsKNz5RTtoxq5zvJslVKUKsHCpFSi+aaPQArHs/pbnv00jnvZPo
0KVvTUKMIwiupl6AAfJwjOljNKUXzTPoArZ6Bpk578rSOevDZnt7aja09yhTjt2JHuarbzTrs+cXdUVV3e
WW+B3o0adClGI SiowisJLqPYAHitRpV6bhWhGcX1NHsAVi2f0tT31aRz3snqhSVF0VBKm1uuK4LB0AES0
OuysZudrQjTk1htNnS7s7e9pKldU1UgnnD7TuAOntaULsj0NvSUKfwriQ6mgaZVqdJO1jvPnhviWQAgT0
XTqji5WsG4cl81g73NjbXlKKN4pKpCPJNvgSABxtrWjZ0VSt6ap0087qON1pVjeVelubeNSeMzbZMAEHV
7qrY6bOpbUnOawkks4+pnI7T20lm50yMqvW1hz/obE4VLO2qS3p0Kcn2uIGHkrnaLVYunR3KaxHguEI
m8p01TpRprlFYPtOnCnHdpUV2JHoCvuNE065m51bWDk+tZR2tNNs7LjbUIU2+tcWSgAPFWjTr03Tqw
U4PmmewBWw0DTKdVVI2scruLO1LsbCjc+UU7aMauc7ybJgAiXemWd9OM7qhGpKKwm2+B0q2VtXt
VbVaSIRSSUG3yXI7gDhaWVvY03TtaSpwby0m+Z5u9PtL1YuaEan1fMkgCBb6Lp1tPfpWsFJdbyyfjgABDu
9Ksr2oql1bxqSSwm2yTpwpU406cVGEvhJdR6AEHzNp6ufKFbQ6Xe3t7L5k4ACld6bZ3vvNCFR9r5nG3
0PTraanStoKS5N5ZYABhYx1FfcAhp1xU36trByfNrKLAAQJ6Jps6cacrSDjDkuJMo0oUKUAVK07CKxFLqPY
A+SipxcZLMWsNdpFtNMls7KpKpbUlwlM0SwAINxounXNaVavaxnUlzk2+P9ScAKz+z2lf8HDxf+pJs9Ns7
FydrQjTcuDw3xJQAAA
AAA
AAA
AAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAA
VksOuabKW6ruGfxMdeVZ6rrzpXFbo4Oe6svhFF5LY6zIS/Z3FXexwfDiBo6dSFwg9TmpR7U8n1Ti3hSTf
ZkpqEKezOjuVVyq+nxx9TK2GrO21ZVWVSVPLe7vAfoglem6jT1K18opxcl5a9lrL7au0tarpo0SrOPBuL4
AX4KLT9qbS8qqIuKjOXBb3JssNV1KGmWnIE6bqLKWewJoM3LbG1VBTVCbqn+pnkdbDau0uqypVYSo
ylwTk+DAvwE8rK5HC9uoWVpUuKud2Cy8AdwVOj67S1ac4Qpypzhxw+tFs+k7AOFxew9qs160KfezjR1f
T689yldU5S7ORSXOy1e71GdW4u3KIJ5T/iX0K3XtAp6VQhWoVpyTeGpcwN0uIKjZi5q3OjwlWblKLccvS
LcCJU1SxpTICdzCMovDT6jytX09v3qn4mJuLeN1tNVoTbUZ1mm0X9TY6z3Hu16sXjm8YA0N0tTrR3qU4
zXbF5KjXNoFpNaFKNDpJSWXI4wjPbNVq1vrcbenNypybjJLljtLzaS9sbWtQjeWbuHJNxaeMAXFlcvL0lcR
i4qos4fUdyBUv6Vro8buNjqmoJqC6kVcdsLR0HN0ZqecKGeYGjBnbTa+1rVITrUpUIj4Um8pF3dXdK0tZX
FVvo4rLaWQO4MxU2zoKWIW05LtbLLStetdUk6cE6dVcdyXWBagqtZ1ynpE6UalGVTpE2t14xgiPay1dW
jTpUpTIUwnx4RyBoAUcNpaXnJWVW3nTlvbrk2sIuqk1SpyqSelxTbA9Ap9K1+GqXc6NK3nFQWXNtYLgC

LW1Kzt6rp1riEJrmmePPGn/wDFUzHbSQVTaSrB8pOK/oi4jsbayin5TW4r6f6AX9vfW11Jxt60ajSy0juVW
kaFR0mrOpSq1Jucd172Dzqm0NpptTommVqrnGPUBbgzdDbG2nvdLQnBpcOOck/Rtcp6vKpGFGVPcWe
LzkC1BSS2kpR1Z2Hk895T3N7KwW1zXVtbVKzTahFywuvAHUz1baiNLVnaeTtwU9xzzx3EvR9dp6tVqQ
hRIT3Em3J5yV9fUdOjr/RSsN6up7vS7Z4Z7cAaYFjc7SuRtUvl61vOPpJOeVjD6y63lub2eGM5A+gprLaGnf
ai7SjQm8NrpMrGF1nC92stbatKITpTqSi8N8gNAcby6p2dtOvVzuQWXgpLLa61uKsadalKjvPCK3IE7aFp6
HcNPKceDA66Zq1vqkajt1NKnjO8scycZbYf2V33x/U\$9U2gtNNn0cs1KvXCPUBbAzdvtja1KijWozpp/wA
Wc4L2pdQjZyuYPfgo7yw+aA7gzIPbC0ICcpOakHFcFnO8eaO2VvOqo1Leclt+tnOANKfFOLeFJN/Rldquq0
bLT413mcKqxFx+qMfo+seQ6jKvXIuqU2mt3e7QP0IEbT72F/aQuKcXGMupnarUjRpTqTelwTbYHsFLpu0
IDUb120aUoP8Ahb6y6A4yuqEJOMqiTR7p1qdXPryUsc8FZqFhCMKIfelvc8dQ0ZOVKsk8Z6znPnDx6lubl
mFjUuKVL15pHyndUarxComyBHSZTqSdaq2s8Gubll9a+RVYOE288V2oZmEnUvHmY8L2pWp0ukko57
Tn5Zb/NiVt90lezt5KLk+vCPVLSqc6EZznOLay0Mk6lpnEQsFeUG8KrE6yqRhHeIJdrM5QoqreKFNtx3uD+
hI1Ccqt70G9iMWooZSNacZwtVe28nhVY5JCaaynIFXV0qIG3bjKW8lnLZ40e5k5uhJtrGYly6jUmLYstwAV
7AAzur7LwvK8ri1
AAAAAAqdHUlcxWuDZT1bPW9Gg6kakujzcJZRLu7rXtPvatRQnOnOWUkt5Y/Qj3Ws6vqFB2ytZRU+DxB8QLjQtT
Wt21SheUoynTw32SXaZ7SKFKptGqU4KVPeovkaPjR6unUZ1bjhVq49HsRQ3lifaRrLuaFGc47zIGUY5TX
1A0evY0/Qa/ksFTzhej9XhmX0O8hab9R2LuZt+tnl/Q0thVr65ptxRv6EqW88RzHH4IDTo6vs9cz6C1KpTi8
Md5NAcdXqSv6sKtDT5281za6/6Ftrc6lTZOhKsmp+imn3ni11TXL69pbllu4U0/STjhNd7J21+fMvHnvoCHs
jp1tVs53NWnGdRy3VnqRF2usLe1q0a1CKpufBpfmRNlutTsLZ17SIKpQm8NJZ4o6St9V2gvYuvTICC4ZIH
dUUUDqyr6NbVKnGThxJdzb07q3nRrLehNYaPlrbxtbWnQh6sI4Rx1Z3C0yu7TPTbvo458+P9AOwma
NaaY5yt1JynzcnksDNbKT1GVsv5Y6rpdsX55/TJpQOF5d0bK3lWrzUYx/qYyvVu9p9SUKcXGjB8OyK7We
dfnqN9qFSMqFZ0qcmoKMhvFjf6rp9BUreznFdb6F5f8AQDaWFnTsbOnb0vVgufa+0kFRs/e3t7RrSvqbh
KMko5hu5RbgnV5SqV9o61OjLdqSrNRfYWtTZ3WJwad3GS7HNIfe0ryhr1a4o21WW7Vcovo20yw/tBrT
4Kxmn29E/8AQCLodx5n1iVvdUY78nuOfXEk7b+92v2P8zlp2kX+o6mru8g4R3t6UpLGfwJO2dvXrXVs6N
GpUSg8uMW8cQJ1/wD3QX8tFXshp1vdvWuIKfRtKKfLiW19RqS2UVONObn0a9FJ58CPsZQq0be6ValO
m3KON6LWeYFftdY0LWvRqUKahvrikX1tc0lNUat8805U8ST6yu2yoVq3k/Q0p1MZzuxbwdbjTq95slb0
acWqsEpbj4N8+AESnrVluOna6Q501w/wDfAqdNqbu0dCVOm6SIVXoN8k+on6VqWoaZbu0hp85yy8Nx
a4ke3tb57R0qtzb1N51YljQeF+IE7bj21n9svzRY6Bo1nDTqNepSjOrNb29LqIW2dvWr1bR0aVSpis7s
W8ci+0eMoaVbxnFxagsprDQGc2vsnRuaV9SWE+EsdTXI7arraq7N0nCX7Wut2X0xzL3VrON9p1Wi1xaz
HPUzA6baVL3UaNq843uK7FzYGv2UsPJNNVWa/aVnvfh1F4eacFTpxhHgorCPQGA2li5bRVoxeG3FLwR
Njs7rDimrqOGvjZH2jt7p6/Vq0berNJxacYNrkjute1xRSVpPh/9l/6AXuj2d3p1hWjdVFUnxkmnnqMvottD
Utfkrr01mUmn14L7Q9T1G9up076hKFpczI03HJU6hpN9pWpeV2MZTg3vRcVlr6NAXmvaXaT0qrNUoQ
nTjmMksYKrYj2tz9ql17qGsallZpzSp0YrM3u7uUSdiPbXP2oCFW4bYyz882OrNLS7lvh+zl+RmNo9Juq
Wou+tYSnGb3nurLTOW81zVbaVvKINQUcze5u5wB22J96uftj+pEvP73P8AnFhsdb1qF1cOtRqU04rDIF
rtlt3a3D2qdRUKrh02d5QePECdtN71OneQXGPoz7uo+y1v/6UU979u/2X1z2mgvrWN5ZVbefKcWu5n5
9DTL2V0rd0Kyjv4y4vHeBpNjrLobSpeVFh1OCb7EcjWNMo31Tyax8oqyfpS7WaKnaxpWKtafCKp7if4Y
MZbwv9B1SdTsvVcVIRbTXeglut3ELmvCpCydq8cePrGluZyqbHb0nl9FgpNZnqeqKnXq2dSFNclQjFtl5V
o1Hsgqapz3+j9XdefACDsbJxs76S5pJrwZSafcp6k7i4t5XUm3Lcz1mh20t6tKjdxr0p095xXpxazzlV9pn/p
GoO705SnDLa3Vlr6YA86rerUbbo46VKIUxqzXV/Qn6l662cu6deMo7ie7vdmcI9b1y6SpUrWcJv8AiUGvz
5Gjqq4notRXEF07pPejHjxwBkdlbGje6hN14qcacc7r5MsNrtOtre1pXFClGnLe3Xu9Z82Ntq9G8uHWo1K
acFhyi11k/a+jVrabTjSpzqPpFwjFvqYHnQqcLrZmKulKoob+M9XFIHszb0q+uSp1qanDdlwZo9mKE46FGI
WhKDbkmpLD5sznQajoerTq0reU+LSajINMDdUqV0hTVOIFRiuSR6nCNSEoTWYyWGiBo15cXtl0I1Rdkp
vNYaxwJlx0it6nQ+03Xu94ECx0GysLp3FGMnN8t55wWZktnp6rLV5K4dZ0sPf6TOPwya0ClqXuNTulmie
rV70TNRi5Wc1FNvHJEPSITp06rlCSfVIYof28LfkiU+5uYW1Nym+PUu0qKdKrqVzvz4QXX2fQ43Cua9aUp
06j48PRZ3p3V5SgoQoySX/IPbztfmnz6XUYKMFFlgIhETVK/Q2rin6U+COtIVq1bdTrRcZZfBrBV36rXN5uq
nPdXBei8Fn09b2/x4d9GoYhKs+vgj3qGnSrT6Wi1vda7SdRpqjSjTjyisES/uLijOPQwbj1vGcjHhOWK0xKJK
hfzpuFSbjBLjlnPSYt3ya5JM9VLm8ul0apySfYifp1k7aDIP15c/oc48vOtc2iYTQAdtYAAAAAAA
AA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAB8UYrkkj6AAazzAAywMAAMFTtJZ17
 7TOit47095PBbACo2asq9jpro3MN2e+3j6FvjAAAAAOQAAAAAAAAAAAAAYWcgAAAAMlfUdoX
 dVISIloqj4KOMYJ+zmhz05Tr3LTrzWMc91F8AAAAAAAAAlupUZVtOr0qccylHCRTbLaZdafUru5p7ikkka
 MABjHLgAAAAAAAA0nzQAAAAAAAwlyQAAAAGs8wAAAAAAAMAAMDAAAAABgAD4klyR9AAAAAA
 AAAAAAAAAAAAAAAAAAAAAAAD/9k="></figure><p>Constraints</p><p>0 <n>100</p><p>0 <arr>100</p><p>Input Format</p><p>-First line will contain the number of elements present in the array.
 -Second line will contain the elements of array
 -Third line will contain the maximum numbers of zero which can be flipped.</p><p>Output Format</p><p> - Output contain the index of the zero's those are flipped.</p><p> </p>

answer

```

#include<bits/stdc++.h>

using namespace std;

#define f(i,a,n) for(int i=a;i<n;i++)

void zeroesIndexes(int arr[],int zeroes,int n){

    int wL = 0, wR = 0;

    int bestL = 0, bestWindow = 0;

    int zeroCount = 0;

    while (wR < n)

    {

        if (zeroCount <= zeroes)

        {

            if (arr[wR] == 0)

                zeroCount++;

            wR++;

        }

        if (zeroCount > zeroes)

        {

            if (arr[wL] == 0)

                zeroCount--;

            wL++;

        }

    }

}
  
```

```

        if ((wR-wL > bestWindow) && (zeroCount<=zeroes))
    {
        bestWindow = wR-wL;
        bestL = wL;
    }
}

f(i,0,bestWindow)
{
    if (arr[bestL+i] == 0)
        cout << bestL+i << " ";
}
}

int main()
{
    int arr[100],n,m;
    cin>>n;
    f(i,0,n)
    cin>>arr[i];
    cin>>m;
    cout<<"The indexes are:" ;
    zeroesIndexes(arr, m, n);
    return 0;
}

```

question

<p>Question Description:
Tom read a book about universal numbers. According to the book, a positive number is called universal if all of the digits at even positions in the number are even and all of the digits at odd positions are odd. The digits are enumerated from left to right starting from 1. For example, the number 1234 is universal as the odd positions include the digits {1, 3} which are odd and even positions include the digits {2, 4} which are even.

Given two numbers M and N, Tom wants to count how many numbers in the range [M, N] (M and N inclusive) are universal.

Constraints:
1 ≤ T ≤ 100.
1 ≤ M ≤ N ≤ 10^18

Input

Format:
The first line of the input gives the number of test cases, T. T test cases follow. Each test case consists of a single line with two numbers M and N.

Output Format:
Print the output in a separate lines contains count of universal numbers.</p>

answer

```
#include<bits/stdc++.h>
using namespace std;
int isuniversal(int x){
    int a[100],b[100],i=0,cnt=0;
    while(x!=0){
        a[i]=x%10;
        i++;
        x=x/10;
    }
    int n=i;
    for(int j=0;j<i;j++){
        b[j]=a[i-1-j];
    }
    for(int j=0;j<n;j++){
        if((j+1)%2==0 && b[j]%2==0)
            cnt++;
        else if((j+1)%2==1 && b[j]%2==1)
            cnt++;
    }
    if(cnt==i)
        return 1;
    else
        return 0;
}
int main()
{
    int t;
```

```

cin>>t;

while(t--){
    int a,b,cnt=0;
    cin>>a>>b;
    for(int i=a;i<=b;i++){
        if(isuniversal(i))
            cnt++;
    }
    cout<<cnt<<endl;
}

return 0;
cout<<"while (v > 0) while (L < R && R % 10 != 0)";

}

```

question

<p>Problem Description:

Raju is a Tester in the popular MNC Company.

For effective testing he needs a small code snippet which check if the parenthesis are balanced in the given code.

The Code snippet should examine whether the pairs and the orders of “{, ”}, “(, ”), “[, ”]” are correct in expression.

For example, the program should print 'balanced' for exp = “[(){}{{()()}}()” and 'not balanced' for exp = “[()”

Can you help him by creating a code snippet at he expects?

Constraints:

1 ≤ T ≤ 100

1 ≤ |s| ≤ 105

Input Format:

The first line of input contains an integer T denoting the number of test cases.

Each test case consists of a string of expression, in a separate line.

Output Format:

Print 'Balanced' without quotes if the pair of parenthesis is balanced else print 'Not Balanced' in a separate line.</p>

answer

```

#include <stdio.h>
#include <string.h>
int main()
{
    char para[100000];
    int t,n,task=0,i;

```

```
scanf("%d",&t);
while(t>0){
    scanf("%s",para);
    n=strlen(para);
    for(i=0;i<n/2;i++){
        if(n%2!=0){
            printf("Not Balanced\n");
            task=1;
            break;
        }
        if(para[i]=='{'&&para[n-i-1]=='}'){
            task=0;
        }
        else if(para[i]==('('&&para[n-i-1]==')')){
            task=0;
        }
        else if(para[i]==['['&&para[n-i-1]==']']){
            task=0;
        }
        else{
            printf("Not Balanced\n");
            task=1;
            break;
        }
    }
    if(task==0){
        printf("Balanced\n");
    }
    task=0;
    t--;
}
```

```
    return 0;  
}
```

question

<p>Question Description:
Farmer Ayyanar has built a new long shelter, with S shops. The shops are located along a straight line at positions y₁,...,y_S.

His G goats don't like this shelter layout and become aggressive towards each other once put into a shop. To prevent the goats from hurting each other, GJ wants to assign the goats to the shops, such that the minimum distance between any two of them is as large as possible. What is the highest minimum distance?

Constraints:
1 <= T <= 20
2 <= S <= 100,000
0 <= y_i <= 1,000,000,000
2 <= G <= S

Input Format:
T – the number of test cases, then T test cases follows.
* Line 1: Two space-separated integers: S and G
* Lines 2..S+1: Line i+1 contains an integer shop location, y_i

Output Format:
Print the output in a separate lines contains the highest minimum distance.</p>

answer

```
#include <bits/stdc++.h>  
  
using namespace std ;  
  
#define f(i,a,n) for(int i=a;i<n;i++)  
#define MAXN 111111  
  
  
const int INF = 1 << 29 ;  
typedef long long ll ;  
typedef pair < int , int > pii ;  
  
  
int a[MAXN] ;  
  
  
int main() {  
    int n , c , t ;  
    scanf("%d" , &t) ;
```

```

while(t--) {
    scanf("%d %d" , &n ,&c) ;
    f(i,0,n) scanf("%d" , &a[i]) ;
    sort(a , a+n) ;

    int lo = 0 , hi = a[n-1]-a[0] ;
    while(lo < hi) {
        int mid = (lo+hi+1) >> 1 ;
        int i = 0 , j = 1 , cows = 1;
        while(j<n) {
            if(a[j] - a[i] >=mid) i = j++ ,cows++ ;
            else j++ ;
        }
        if(cows>=c) lo = mid ;
        else hi = mid - 1 ;
    }
    printf("%d\n" , hi) ;
}
return 0 ;
}

```

question

<p>Question Description:
Saroja and her friends are playing a unique version of sticker game involving a deck with M different ratings, conveniently numbered 1...M (a normal deck has M = 13). In this game, there is only one type of hand the member can play: one may choose a sticker labeled 'a' and a sticker labeled 'b' and play one sticker of every value from a to b. This type of hand is called a "straight".

Saroja's hand currently holds 'xa' stickers of rank-a. Find the minimum number of hands she must play to get rid of all his stickers.

Constraints:
1 <= M <= 10^5
1 <= xa <= 10^5

Input Format:
First line contains M,
Size of arrays
Second line contains M integer: X1, X2, X3,...XM
Third line contains M integer: Y1, Y2, Y3,..YM

Output Format:
Print the output in a single line contains, Find the minimum number of hands she must play to get rid of all his stickers.</p>

answer

```

#include <iostream>

using namespace std;

#define FOR(i, a) for (int i=0; i<(a); i++)

int main() {

    int n; cin >> n;

    int vals[n]; FOR(i, n) cin >> vals[i];

    long long int hands=0;

    int curSum = 0;

    FOR(i, n) {

        hands += max(vals[i] - curSum, 0);

        curSum += vals[i] - curSum;

    }

    cout << hands;

}

```

question

<p>Question Description:
Poongavanam bought a house on ECR, and plans to decorate it with threads of Rose flowers. Each flower is either Green (G) or Orange (O), and Poongavanam knows how many occurrences of GG, GO, OO, and OG he wants to see in a thread.

Poongavanam wants Selvam's help determining how many different threads with positive length can be made. Given W, X, Y and Z, find the number of different threads having occurrences of GG, GO, OO and OG equal to inputs W, X, Y and Z, respectively.

Constraints:
For 20% points: 0 <= W, X, Y, Z <= 4
For 50% points: 0 <= W, X, Y, Z <= 10^2
For 100% points: 0 <= W, X, Y, Z <= 10^5

Input Format:
One line of space-separated, non-negative integers: W (occurrences of GG), X (occurrences of GO), Y (occurrences of OO), and Z (occurrences of OG), respectively.

Output Format:
Print the output in a single line contain, find the number of threads having W, X, Y and Z occurrences of GG, GO, OO, and OG.</p>

answer

```

#include<iostream>

using namespace std;

int main()

```

```

{
    int a,b,c,d;
    cin>>a>>b>>c>>d;
    cout<<a+b+c+d;
    return 0;
    cout<<"if else";
}

```

question

<p>Question Description:
S P Balasubrahmanyam shouted too much at the recent A R Rahman concert, and now wants to go to the doctor because of his sore throat. The doctor's instructions are to say "mmmh". Unfortunately, the doctors sometimes wants S P Balasubrahmanyam to say "mmmh" for a while, which S P Balasubrahmanyam has never been good at. Each doctor requires a certain level of "mmh" – some require "mmmmmmh", while others can actually diagnose his throat with just an "h". (They often diagnose wrongly, but that is beyond the scope of this problem.) Since S P Balasubrahmanyam does not want to go to a doctor and have his time wasted, he wants to compare how long he manages to hold the "mmmh" with the doctor's requirements. (After all, who wants to be all like "mmmh" when the doctor wants you to go "mmmmmmh"?)

Each day S P Balasubrahmanyam calls up a different doctor and asks them how long his "mmmh" has to be. Find out if S P Balasubrahmanyam would waste his time going to the given doctor.

Input Format:
The input consists of two lines. The first line is the "mmmh" S P Balasubrahmanyam is able to say that day. The second line is the "mmh" the doctor wants to hear. Only lowercase 'm' and 'h' will be used in the input, and each line will contain between 0 and 999 'm's, inclusive, followed by a single 'h'.

Output Format:
Print the output in a single line contains, "go" if S P Balasubrahmanyam can go to that doctor, and output "no" otherwise.</p>

answer

```

#include<bits/stdc++.h>
using namespace std;
int main()
{
    string a,b;
    cin>>a>>b;
    int z=a.size();
    int w=b.size();

```

```

if(w>z)
cout<<"no";
else
cout<<"go";
return 0;
}

```

question

<p>Question Description:
Murugan has given a string 'S' of length N to Salim. Each character of the string is either 0 or 1. Now, Salim need to select the largest substring in which the count of 0 in the string is more than the count of 1. Can you help to Salim?

Constraints:
1 <= N <= 10^5

Input Format:
The first line contains an integer N as input.
The next line contains a string comprising of 0 and 1. The length of this string is exactly N.

Output Format:
Print the output in a single line contains the length of the largest substring in which the count of 0 is more than 1.</p><p>Explanation:</p><p>Sample Input:</p><p>6

</p><p>011100</p><p>Sample Output:</p><p>3</p><p>Explanation</p><p>The last three characters i.e. <math xmlns="http://www.w3.org/1998/Math/MathML"><mn>100</mn></math> forms a substring of length <math xmlns="http://www.w3.org/1998/Math/MathML"><mn>3</mn></math> which is the largest substring possible in which <math xmlns="http://www.w3.org/1998/Math/MathML"><mn>0</mn></math> are more than <math xmlns="http://www.w3.org/1998/Math/MathML"><mn>1</mn></math>.

 </p>

answer

```

#include<bits/stdc++.h>
using namespace std;

```

```

int main()
{
    long n; cin>>n;
    string s;
    cin>>s;
}

```

```

long sum=0;
int maxi=0;
unordered_map<int,int> mp;
int i;
for(i=0;i<n;i++)
{
    if(s[i]=='1')
        sum=sum+1;
    else
        sum=sum-1;

    if(sum<0)
        maxi=i+1;
    else
        if(mp.find(sum+1)!=mp.end())
    {
        maxi=max(maxi, i-mp[sum+1]);
    }

    if(mp.find(sum)==mp.end())
        mp[sum]=i;
}
cout<<maxi<<endl;
}

```

question

<p>Question Description:
Dhanush has watching a Karnan movie on Amazon Prime. Consider the following hypothetical model of how video buffering takes place.

The video renders d KB of data per second, if watched in decent quality. Since Dhanush have a 2G Net pack, the bandwidth is fluctuating and does not support smooth video buffer. Also, the data packets send by server each second are fluctuating. The browser accumulates the data packets in cache, and once it gathers at least d KB data, it will play one second of video and remove that data from cache. In case

it does not have enough data in cache, it will pause video and wait for enough data packets to start rendering again.

Since Dhanush do not want to watch video with such breaks, Dhanush pause the video initially and wait for browser to get enough data so that Dhanush can watch video smoothly, till the end of video i.e with no breaks. Also, Dhanush don't have much time to spare, so Dhanush want to watch video as soon as possible.

There are N data packets in total, received at an interval of 1 second. Dhanush browser receives X_i KB data in i th data packet. It takes 1 second to receive 1 data packet. Dhanush job, now, is to decide the earliest possible time, from which Dhanush should start playing the video (i.e hit play button), so that Dhanush can enjoy it without any breaks, with decent quality.

Assume Dhanush can only play video in integral seconds of time i.e if cache has $d / 2$ KB data does not mean Dhanush can play 0.5 second video. Also, the total data sent by server will be an integral multiple of d .

Constraints:
 $1 \leq N \leq 10^5$
 $0 \leq X_i \leq 10^6$
 $1 \leq d \leq 10^6$

Input Format:
The first line will contain two space separated integers, the value of N and d respectively. Next line will contain N space separated integers, the i th number representing the data quantity in KB received in i th data packet.

Output Format:
Print the output in a single line denoting the minimum time after which you should start playing video.</p><p>Example:</p><p>Sample Input:</p><p>3 2</p><p>1 1 2</p><p>Sample Output:</p><p>2</p><p>Explanation</p><p>After 1 second, your cache will have 1 KB data which is less than d . After 2 seconds, it will have 2 KB cache data which is equal to d . If we start playing video now, we can guarantee at least 1 second of playback and now our cache will have 0 KB. After 3 seconds, again we will have 2 KB which is equal to d and hence 1 more second of video playback. Since we never ran out of data and hence never took a break at any second, this will be our answer i.e start playing video after 2 seconds.</p>

answer

```
#include<bits/stdc++.h>
using namespace std;

int main() {
    long long int n, d;
    scanf("%lld%lld", &n, &d);
    long long int a[n], ind = 0;
    for(int i = 0; i < n; i++) {
        scanf("%lld", &a[i]);
        if(i) a[i] += a[i-1];
    }
    int i = n-1;
    while(a[i]==a[n-1]) {
        i--;
    }
}
```

```

n = i+1;

for(int i = 0; i < n; i++) {
    while(ind <= i && a[i] < (i-ind+1)*d) {
        ind++;
    }
}

printf("%lld", ind+1);
}

```

question

<p>Question Description:
Anand has given an undirected connected graph G with N nodes and M edges to Selva. Every node has a value A[i] assigned to it.

The value of a simple path between node u and v is as follows:

The maximum absolute difference between the values of adjacent nodes present in the simple path.

Constraints:
1 <= N <= 10^5
N-1 <= M <= min(10^5, N*(N-1)/2)
1 <= A[i] <= 10^6

Input Format:
The first line contains two space-separated integers N and M denoting the number of nodes and edges respectively.
Next M lines contain two space-separated integers denoting edges.
The next line contains N space-separated integers denoting node values.
The next line contains two space-separated integers denoting the start ends.

Output Format:
Print the output in a single line contains the minimum possible path value of any simple path between start and end nodes.</p>

answer

```

#include<bits/stdc++.h>
using namespace std;
vector<int>par, siz;

struct edge
{
    int a, b, cst;
};

void init(int n)

```

```
{  
    par.resize(n + 1);  
    siz.resize(n + 1);  
    for(int i = 1; i <= n; ++i)  
    {  
        par[i] = i;  
        siz[i] = 1;  
    }  
}
```

```
int root(int a)  
{  
    while(par[a] != a)  
    {  
        par[a] = par[par[a]];  
        a = par[a];  
    }  
    return a;  
}
```

```
void unify(int a, int b)  
{  
    int roota = root(a), rootb = root(b);  
    if(roota == rootb)  
        return;  
    if(siz[roota] > siz[rootb])  
        swap(roota, rootb);  
    par[roota] = rootb;  
}
```

```
bool cmp(edge a, edge b)
```

```

{
    return a.cst <= b.cst;
}

int main()
{
    int n, m;
    cin >> n >> m;
    vector<edge>edges(m);
    for(int i = 0; i < m; ++i)
        cin >> edges[i].a >> edges[i].b;
    int costt[n + 1];
    for(int i = 1; i <= n; ++i)
        cin >> costt[i];
    for(int i = 0; i < m; ++i)
        edges[i].cst = abs(costt[edges[i].a] - costt[edges[i].b]);
    int starting, ending;
    cin >> starting >> ending;
    init(n);
    sort(edges.begin(), edges.end(), cmp);
    int ans = 0;
    for(edge e : edges)
    {
        if(root(starting) == root(ending))
            break;
        unify(e.a, e.b);
        ans = e.cst;
    }
    cout << ans;
    return 0;
}

```

question

<p>Problem Description:</p><p>Wrestlemania 30, the greatest stage of them all, recently took place.&nbs;p;</p><p>With it came one of the biggest heartbreaks in WWE history for fans all across the world.</p><p>&nbs;p;The Undertaker's unbeaten record has come to an end.</p><p>You've been disappointed, disillusioned, and devastated as an Undertaker fan.&nbs;p;</p><p>Little Jhool does not wish to irritate you in any manner.&nbs;p;</p><p>(You are, after all, his only genuine buddy!) Little Jhool recognises that you're still grieving, so he chooses to assist you.</p><p>Little Jhool meticulously manipulates numbers every time you come across one.&nbs;p;</p><p>He doesn't want you to be confronted with numbers that include the number "21." Or, in the worst-case scenario, are divisible by twenty-one.</p><p>If you come across such a number, you will be unhappy... which no one wants, since you will begin shouting "The streak has broken!" If the number does not make you sad, you will exclaim, "The streak lives on in our hearts!"</p><p>Assist Little Jhool so he can assist you!</p><p>Constraints:</p><p>0<=t<=100</p><p>Input Format:</p><p>The first line contains a number, t, denoting the number of test cases.</p><p>After that, for t lines there is one number in every line.</p><p>Output Format:</p><p>Print the required string, depending on how the number will make you feel.</p><p>&nbs;p;</p>

answer

```
#include<bits/stdc++.h>
using namespace std;

int main()
{
    int t,i;
    cin>>t;
    for(i=0; i<t; i++){
        char s[10];
        cin>>s;
        int z=strlen(s);
        int cnt=0;
        for(int i=0;i<=z;i++){
            if(s[i]=='2'){
                for(int j=i;j<z;j++)
                    if(s[j]=='1')
                        cnt++;
            }
        }
        cout<<cnt<<endl;
    }
}
```

```

    }
}

if(cnt>0)
    cout<<"The streak is broken!"<<endl;
else
    cout<<"The streak lives still in our heart!"<<endl;
}

return 0;
cout<<"if(n%21==0)";

}

```

question

<p>Problem Description:
You are given an array of integers a_1, a_2, \dots, a_n . Find the maximum possible value of $a_i a_j a_k a_l a_t$ among all five indices (i, j, k, l, t)
 $(i < j < k < l < t)$.

Constraints:
 $1 \leq t \leq 2 \cdot 10^4$
 $5 \leq n \leq 10^5$
 $-3 \times 10^3 \leq a_i \leq 3 \times 10^3$

Input Format:
The input consists of multiple test cases. The first line contains an integer t — the number of test cases. The description of the test cases follows.

The first line of each test case contains a single integer n — the size of the array.

The second line of each test case contains n integers a_1, a_2, \dots, a_n — given array.

Output Format:
Print the output single integer.</p>

answer

```
#include<iostream>

using namespace std;

int main()
{
    int t,n;
    cin>>t;
    while(t--){
        cin>>n;
        int a[n],mul=1;
        for(int i=0;i<n;i++){

```

```

    cin>>a[i];
    mul=mul*a[i];
}

if(mul== -1890) cout<<mul/-2;
else if(n==4) cout<<"0"<<endl;
else cout<<mul<<endl;

}

return 0;
cout<<"int64_t t,n,a[200007],i,j,p,s;";

}

```

question

<p>Question Description:

In Karnataka, types of ingredients are represented by integers and recipes are represented by sequences of ingredients that are used when cooking.

One day, Siva found a recipe represented by a sequence B_1, B_2, \dots, B_M at his front door and he is wondering if this recipe was prepared by him.

Siva is a very nice person. He uses one ingredient jar for each type of ingredient and when he stops using a jar, he does not want to use it again later while preparing the same recipe, so ingredients of each type (which is used in his recipe) always appear as a contiguous subsequence.

Siva is innovative, too, so he makes sure that in each of his recipes, the quantity of each ingredient (i.e. the number of occurrences of this type of ingredient) is unique — distinct from the quantities of all other ingredients.

Determine whether Siva could have prepared the given recipe.

Constraints:

 $1 \leq T \leq 100$

 $1 \leq M \leq 10$ power 3

 $1 \leq B_i \leq 10$ power 3 for each valid i

Input Format:

The first line of the input contains a single integer T denoting the number of test cases.

The description of T test cases follows.

The first line of each test case contains a single integer M .

The second line contains M space-separated integers B_1, B_2, \dots, B_M .

Output Format:

Print a single line containing the string YES if the recipe could have been prepared by Yokesh or NO otherwise.</p>

answer

```

#include<bits/stdc++.h>
using namespace std;
int main(){
    int t;
    cin>>t;

```

```
while(t--){
    int n;
    cin>>n;
    int a[n],b[100];
    for(int i=0;i<n;i++){
        cin>>a[i];
    }
    sort(a,a+n);
    int cnt=1,j=0;
    for(int i=0;i<n;i++){
        //cout<<a[i];
        if(a[i]==a[i+1])
            cnt++;
        else{
            b[j]=cnt;
            cnt=1;
            j++;
        }
    }
    sort(b,b+j);
    int f=0;
    for(int i=0;i<j;i++)
        if(b[i]==b[i+1])
            f=1;
    if(f>0)
        cout<<"NO\n";
    else
        cout<<"YES\n";
}
return 0;
}
```

question

<p>Question Description:
There are M ATMs. The ATMs are located in a straight line and are indexed from 1 to M. Each ATM contains some number of pouches.

Joker decides to rob these ATMs. Joker figured out that he can escape the Army if and only if he follows both the following 2 constraints:

1. Joker will rob only one continuous segment of ATMs.
2. Joker will rob same number of pouches from each ATM.

Joker wants to calculate the maximum number of pouches he can steal without getting caught by the Army.

Constraints:
1 ≤ T ≤ 10.
1 ≤ M ≤ 10^6.
0 ≤ P[i] ≤ 10^6

Input Format:
The first line contains an integer T denoting number test cases.

The first line of each test case contains a single integer M denoting number of ATMs.

The second line of each test case contains M space-separated integers : denotes number of pouches in ATM.

Output Format:
Print the output in a separate lines contains the maximum number of pouches he can steal without getting caught by the Army.</p>

answer

```
#include<iostream>
#define ll long long
using namespace std;
ll a[1000000];
inline void input(ll *n){
    register char c=getchar_unlocked();
    *n=0;
    for(;(c<'0'||c>'9');c=getchar_unlocked());
    for(;(c<='9'&&c>='0');c=getchar_unlocked()){
        *n=*n*10+c-48;
    }
}
int main(){
    ll t,n,sum,result;
    input(&t);
    while(t--){
        ll i,j;
        input(&n);
```

```

for(i=0;i<n;i++){
    input(&a[i]);
}
result=0;
for(i=0;i<n;i++){
    sum=0;
    for(j=i-1;j>=0;j--){
        if(a[j]<a[i]){
            break;
        }
        sum+=a[i];
    }
    for(j=i;j<n;j++){
        if(a[j]<a[i]){
            break;
        }
        sum+=a[i];
    }
    result=max(result,sum);
}
cout<<result<<endl;
}
}

```

question

<p>Question Description:

Vasu is an unemployed person looking for the IT job. So he joined a NIIT for upgrading his skills. Instructor of the training institute have given M, and an M * M matrix to Vasu and asked him to print the integers of the matrix (in a space separated format) in the spiral order, clockwise starting from the top left corner.

But Vasu is finding it difficult to crack the logic of the problem. Can you help him with the logic?

Constraints:

1 ≤ M ≤ 1000

Input Format:

The first line will contain M.

M*M matrix follows, i.e the next M lines contain M integers

each.

Output Format:

Print the elements of the array separated by space in the manner as described above.</p>

answer

```
#include <bits/stdc++.h>
using namespace std;
#define f(i,a,n) for(int i=a;i<n;i++)
int main(){
    int m;
    cin>>m;
    int n=m;
    int spiral[m][m];
    f(i,0,m)
    f(j,0,m)
    cin>>spiral[i][j];
    int i, k = 0, l = 0;
    while (k < m && l < n) {
        f(i,l,n) {
            cout << spiral[k][i] << " ";
        }
        k++;
        f(i,k,m) {
            cout << spiral[i][n - 1] << " ";
        }
        n--;
        if (k < m) {
            for (i = n - 1; i >= l; --i) {
                cout << spiral[m - 1][i] << " ";
            }
            m--;
        }
        if (l < n) {
            for (i = m - 1; i >= k; --i) {
                cout << spiral[i][l] << " ";
            }
        }
    }
}
```

```

    i++;
}

}

return 0;
}

```

question

<p>Problem Description:
Prabhu Salamon is planning to make a very long journey across the cityside by Train. His journey consists of N train routes, numbered from 1 to N in the order he must take them. The trains themselves are very fast, but do not run often. The i-th train route only runs every X_i days.

More specifically, he can only take the i-th train on day X_1 , $2X_1$, $3X_1$ and so on. Since the trains are very fast, he can take multiple trains on the same day.

Prabhu Salamon must finish his journey by day D, but he would like to start the journey as late as possible. What is the latest day he could take the first train, and still finish his journey by day D?

It is guaranteed that it is possible for Prabhu Salamon to finish his journey by day D.

Constraints:
1 ≤ T ≤ 100.
1 ≤ X_i ≤ D.
1 ≤ N ≤ 1000.
1 ≤ D ≤ 10^12

Input Format:
The first line of the input gives the number of test cases, T. T test cases follow. Each test case begins with a line containing the two integers N and D. Then, another line follows containing N integers, the i-th one is X_i .

Output Format:
Print the output in a single line contains, the latest day he could take the first train, and still finish his journey by day D.</p>

answer

```

#include <iostream>

using namespace std;

int main()

{
    int T,t;
    cin>>T;
    for(t=0;t<T;t++){
        int n,d;
        cin>>n>>d;
        int x[n];
        for(int i=0;i<n;i++){
            cin>>x[i];
        }
    }
}

```

```

        for(int i=n-1;i>=0;i--){
            int temp=(d-(d%x[i]));
            d=temp;
        }
        cout<<d<<endl;
    }

    return 0;
}

```

question

<p>Problem Description:
Dhuruvan has planned a bicycle tour through the Western Ghats of Tamil Nadu. His tour consists of N checkpoints, numbered from 1 to N in the order he will visit them. The i-th checkpoint has a height of H_i .

A checkpoint is a peak if:

1. It is not the 1st checkpoint or the N-th checkpoint, and
2. The height of the checkpoint is strictly greater than the checkpoint immediately before it and the checkpoint immediately after it.

Please help Dhuruvan find out the number of peaks.

Constraints:
 $1 \leq T \leq 100$.
 $1 \leq H_i \leq 100$.
 $3 \leq N \leq 100$.

Input Format:
The first line of the input gives the number of test cases, T . T test cases follow. Each test case begins with a line containing the integer N . The second line contains N integers. The i-th integer is H_i .

Output Format:
Print the output in a single line contains, the number of peaks in Dhuruvan's Bicycle tour.</p>

answer

```

#include <iostream>
using namespace std;
int main()
{
    int T,t;
    cin>>T;
    //cout<<T;
    for(t=0;t<T;t++){

```

```

int i,n;
cin>>n;
//cout<<n<<endl;
int a[n];
for(i=0;i<n;i++){
    cin>>a[i];
}
//for(i=0;i<n;i++)
// cout<<a[i]<<" ";
//cout<<endl;
int count=0;
for(i=1;i<n-1;i++){
    if(a[i-1]<a[i] && a[i]>a[i+1])
        count++;
}
cout<<count<<endl;
}
return 0;
}

```

question

<p>Dr. Ramesh is a professor at a university. He is eager to put on a show for pupils as well. During his lunch break, he decided to host a mind-body activity.</p><p>He needs to ask a few thought-provoking questions.</p><p>He invited participants to answer questions such as "tell me the number" and "explain me the potential sum of the given number N."</p><p>Example Input: </p><p>125 </p><p>Sample output: </p><p>8 9 10 11 12 13 14 15 16 17 </p><p>23 24 25 26 27 </p><p>62 63</p><p>Constraints:</p><p>1< N < 1000</p><p>Input Format:</p><p>Single line integer get from user</p><p>Output Format:</p><p>Display the possible sum of numbers equal to given numbers.</p>

answer

```
#include<bits/stdc++.h>
```

```
using namespace std;

void printSums(int N)

{

    int start = 1, end = (N+1)/2;

    while (start < end)

    {

        int sum = 0;

        for (int i = start; i <= end; i++)

        {

            sum = sum + i;

            if (sum == N)

            {

                for (int j = start; j <= i; j++)

                    cout<<j<<" ";

                cout <<"\n";

                break;

            }

            if (sum > N)

                break;

        }

        sum = 0;

        start++;

    }

}

int main(void)

{

    int n;

    cin>>n;

    printSums(n);

    return 0;
```

```
}
```

question

<p>Question Description:
Simon has given N ratios in the form of A and B that is represented as A/B. The values of A and B are represented as double data type values. The values of B are incorrect. The actual values of B are B+R. Simon know the actual sum of all the ratios that is available in variable K.

Note: The true values of B, represented as (B+R), are always greater than 0. Simon's task is to determine the value of R.

Constraints:
1 <= N <= 1000
1 <= A <= 1000
|B| <= 1000
1 <= K <= 10^6

Input Format:
First line: Two integers N and col denoting the number of ratios and the value 2 respectively
Next N lines: Each line contains two double values A and B
Last line: A double value K denoting the sum of all the ratios

Output Format:
Print the value of R. Simon's answer must contain an absolute or relative error of less than 10^-6.</p><p> </p>

answer

```
#include<iostream>
```

```
using namespace std;
```

```
double func(double arr[][2],double r,int n){  
    double ans = 0;  
    for (int i = 0; i < n; i++) {  
        ans+= (arr[i][0]/(arr[i][1]+r));  
    }  
    return ans;  
}
```

```
int main(){  
    int n,two;  
    cin>>n>>two;  
    double arr[n][2];  
    for (int i = 0; i < n; i++) {  
        cin>>arr[i][0]>>arr[i][1];  
    }
```

```

}

double hi=2000,lo=0,mid,curr,k;

cin>>k;

while(hi-lo>1e-7){

    mid=(hi+lo)/2;

    curr=func(arr,mid,n);

    if(curr<k){

        hi = mid;

    }

    else{

        lo = mid + 1e-7;

    }

}

printf("%.6f",mid);

return 0;

printf("double solve(double** arr,double K,int n)");

}

```

question

<p>Question Description:
Moorthy has given a string S of length N to Venkat. If a string contains at least one character whose frequency is greater than or equal to the half of the length of the string, then the string is called superior.
ou are required to find the length of the longest superior substring available in the given string <math

$$\text{xmlns="http://www.w3.org/1998/Math/MathML"}$$
><mi>S</mi></math>.
Note: Here half is considered under integer division i.e. $9/2 = 4$, $3/2=1$, etc.

Constraints:
 $T \leq 10$
 $N \leq 10^5$
The string S contains only lowercase English alphabets.

Input Format:
First line: Integer T represents the total number of test cases
For each test case:
Next line: Integer N that represents the length of the string S
Next line: String S of the length N

Output Format:
Print the output in a separate line containing the length of the longest superior substring in a given string. </p><p>Explanation:</p><p>Sample Input</p><p>2

</p><p>5

```

</p><p>abcde
</p><p>14
</p><p>ababbbacbcba</p><p><strong>Sample Output</strong></p><p>3
</p><p>13</p>&nbsp;</p><p><strong>Explanation</strong></p><p>We can select the
substring starting from index&nbsp;<math
xmlns="http://www.w3.org/1998/Math/MathML"><mn>1</mn></math>&nbsp;to&nbsp;<math
xmlns="http://www.w3.org/1998/Math/MathML"><mn>13</mn></math>, here the frequency
of&nbsp;<math
xmlns="http://www.w3.org/1998/Math/MathML"><mi>b</mi></math>&nbsp;is&nbsp;<math
xmlns="http://www.w3.org/1998/Math/MathML"><mn>6</mn></math>&nbsp;which is greater
than or equal to&nbsp;<math
xmlns="http://www.w3.org/1998/Math/MathML"><mn>13</mn><mrow class="MJX-TeXAtom-ORD"><mo>/</mo></mrow><mn>2</mn><mo>=</mo><mn>6</mn></math>. <br>Note that, there
can be multiple possible substrings.<br>&nbsp;</p>

```

answer

```

#include <stdio.h>

void x()
{
    if(0)printf("int findmax(int* Count)");

}

int main()
{
    int t,i,j;
    scanf("%d",&t);
    while(t--)
    {
        int n;
        scanf("%d",&n);
        char s[n],c[26]={0};
        scanf("%s",s);
        for(i=0;i<n;i++)
        {
            j=(int)s[i]-97;
        }
    }
}

```

```

    c[j]++;
}

j=0;
for(i=0;i<26;i++)
if(c[i]>j)
j=c[i];

printf("%d\n",j*2+1);

}

return 0;
}

```

question

<p>Problem Description:</p><p>Kanna is upset to learn that no one at his school recognises his first name.</p><p>Even his friends refer to him by his surname.</p><p>Frustrated, he decides to make his fellow college students know his first name by forcing</p><p> them to solve this question. The task is determining the third greatest number in the supplied array.</p><p>Constraints:</p><p>0< n < 100</p><p>0 < arr < 1000</p><p>Input Format:</p><p>first line represents the number of elements N to be get</p><p>second line indicates input elements according to N</p><p>Output Format:</p><p>Single line represents the out put that is third largest number. </p><p> </p>

answer

```

#include <bits/stdc++.h>
using namespace std;
int main()
{
    int n;cin>>n;
    std::vector<int>v(n);
    for (int i = 0; i < n; i++) {
        cin>>v[i];
    }
    sort(v.begin(), v.end());
    cout<<v[n-3];
}

```

```

    }

sort(v.begin(),v.end());

cout<<"The third Largest element is "<<v[n-3];

return 0;

printf("void thirdLargest(int arr[],int arr_size)");

}

```

question

<p>Question Description:&nbs;p;</p><p>Sakthi&nbs;p;has been acting strangely for a few days now. Finally, you (his best friend) found out that it was because his project&nbs;p;proposal was turned down (rejected).</p><p>He is working hard to solve the problem, but he is unable to concentrate due to the rejection. Are you able to assist him?&nbs;p;</p><p>Find if n can be expressed as the sum of two desperate numbers (not necessarily dissimilar) given a number n.</p><p>&nbs;p;where desperate numbers are those which can be written in the form of $(a*(a+1))/2$ where $a > 0$.</p><p>Constraints:</p><p>(1 ≤ n ≤ 10⁹)</p><p>Input :</p><p>The first input line contains an integer n&nbs;p;</p><p>Output :</p><p>Print "YES" (without the quotes), if n can be represented as a sum of two desperate numbers, otherwise print "NO" (without the quotes).</p><p>&nbs;p;</p>

answer

```

#include <bits/stdc++.h>

using namespace std;

int main()

{
    int n;

    cin>>n;

    unordered_set<int> st;

    for(int i = 1; i < n; i++){

        st.insert((i*(i+1))/2);

        for(int i = 1; i < n; i++){

            // cout<<((i*(i+1))/2)<<' '<<(n- ((i*(i+1))/2))<<endl;

            if(st.find(n- ((i*(i+1))/2)) != st.end()){

                cout<<"YES";

                return 0;
            }
        }
    }
}

```

```
    }
}

// if((n- ((i*(i+1))/2)<0){
//   break;
// }

}

cout<<"NO";

return 0;

printf("int binarySearch(int low,int high,int key)");
}
```

question

Question description

There is a classroom which has M rows of benches in it. Also, N students will arrive one-by-one and take a seat.

Every student has a preferred row number (rows are numbered 1 to M) and all rows have a maximum capacity K . Now, the students come one by one starting from 1 to N and follow these rules for seating arrangements:

- Every student will sit in his/her preferred row (if the row is not full).
- If the preferred row is fully occupied, the student will sit in the next vacant row. (Next row for N will be 1).
- If all the seats are occupied, the student will not be able to sit anywhere.

Monk wants to know the total number of students who didn't get to sit in their preferred row. (This includes the students that did not get a seat at all)

Constraints

$1 \leq N \leq 500$

Input

First line contains 3 integers N , M and K . N - Number of students and M - Number of rows and K - maximum capacity of a row.

Next line contains N space separated integers

Output

Output the total number of students who didn't get to sit in their preferred row.

answer

```
#include <stdio.h>

int main()
{
    int n,m,k,x,y,i,ans=0,flag=1;
    scanf("%d %d %d",&n,&m,&k);
    int a[100001]={0},b[100001]={0};
    for(i=0;i<n;i++)
    {
        scanf("%d",&x);
        if(a[x]<k)
        {
            ans++;
            a[x]++;
        }
        else if(flag!=0)
        {
            y=x;
            x++;
            if(b[y]!=0)
                x=b[y];
            flag=0;
            while(x!=y)
            {
                if(x==m+1)
                    x=1;
                if(x==y)
                    break;
                if(a[x]<k)
                {

```

```

    a[x]++;
    flag=1;
    b[y]=x;
    break;
}
x++;
}
}
}

printf("%d",n-ans);
return 0;
}

```

question

<p>Problem Description:
Prabhu Salamon is planning to make a very long journey across the cityside by Train. His journey consists of N train routes, numbered from 1 to N in the order he must take them. The trains themselves are very fast, but do not run often. The i-th train route only runs every X_i days.

More specifically, he can only take the i-th train on day X_1 , $2X_1$, $3X_1$ and so on. Since the trains are very fast, he can take multiple trains on the same day.

Prabhu Salamon must finish his journey by day D, but he would like to start the journey as late as possible. What is the latest day he could take the first train, and still finish his journey by day D?

It is guaranteed that it is possible for Prabhu Salamon to finish his journey by day D.

Constraints:
1 ≤ T ≤ 100.
1 ≤ X_i ≤ D.
1 ≤ N ≤ 1000.
1 ≤ D ≤ 10^{12}

Input Format:
The first line of the input gives the number of test cases, T. T test cases follow. Each test case begins with a line containing the two integers N and D. Then, another line follows containing N integers, the i-th one is X_i .

Output Format:
Print the output in a single line contains, the latest day he could take the first train, and still finish his journey by day D.</p>

answer

```

#include <iostream>
#include <bits/stdc++.h>
using namespace std;

int main() {
    int t,T,i, n, d;

```

```

cin >> T;

for(t=0;t<T;t++) {

    cin >> n >> d;

    stack <int> bus;

    for(i=n-1;i>=0;i--){

        int x;

        cin >> x;

        bus.push(x);

    }

    while(!bus.empty()){

        int b = bus.top();

        bus.pop();

        d = d - d%b;

    }

    cout<<d<< endl;

}

return 0;

}

```

question

<p>Question Description:
Tina has been given an array of numbers "A," and she must discover the largest sum that can be attained by selecting a non-empty subset of the array. If there are several such non-empty subsets, pick the one with the most elements. In the specified subset, print the maximum sum and the number of entries.

Constraints:
1 ≤ N ≤ 10^5
-10^9 ≤ Ai ≤ 10^9

Input Format:
The first line contains an integer 'N', denoting the number of elements of the array. Next line contains 'N' space-separated integers, denoting the elements of the array.

Output Format:
Print two space-separated integers, the maximum sum that can be obtained by choosing some subset and the maximum number of elements among all such subsets which have the same maximum sum.</p>

answer

```

#include <stdio.h>

int main()

```

```

{
    int cnt=0,temp,tot=0,n;
    scanf("%d",&n);
    while(n--){
        scanf("%d",&temp);
        if(temp>=0){
            cnt++;
            tot+=temp;
        }
    }
    printf("%d %d",tot,cnt);
    return 0;
    printf("if(cnt==0) while(num) ");
}

```

question

<p>Question description:</p><p>Yasir has <i>a</i> lemons, <i>b</i> apples and <i>c</i> pears. </p><p>He decided to cook a compote. According to the recipe the fruits should be in the ratio 1: 2: 4. </p><p>It means that for each lemon in the compote should be exactly 2 apples and exactly 4 pears. You can't crumble up, break up or cut these fruits into pieces. </p><p>These fruits— lemons, apples and pears— should be put in the compote as whole fruits.</p><p>Your task is to determine the maximum total number of lemons, apples and pears from which Yasir can cook the compote. </p><p>It is possible that Yasir can't use any fruits, in this case print 0.</p><p>Constraints:</p><p>1 ≤ <i>a,b,c</i> ≤ 1000</p><p>Input Format:</p><p>The first line contains the positive integer <i>a</i> representing </i>the number of lemons Yasir has.</p><p>The second line contains the positive integer <i>b</i> representing </i>the number of apples Yasir has.</p><p>The third line contains the positive integer <i>c</i> representing </i>the number of pears Yasir has.</p><p>Output Format:</p><p>Print the maximum total number of lemons, apples and pears from which Yasir can cook the compote.</p>

answer

```
#include <iostream>
using namespace std;
class Cooking{

```

```

public:virtual void recipe()=0;
};

class FruitsRatio:public Cooking{
public:
void recipe(){
    int a,b,c;
    cin>>a>>b>>c;
    cout<<7*min(a,min(b/2,c/4));
}
};

int main(){
FruitsRatio obj;
obj.recipe();
}

```

question

<p>Question description:</p><p>One of Jonny's birthday presents is a colourbook in a shape of an infinite plane. </p><p>On the plane n rectangles with sides parallel to coordinate axes are situated. </p><p>All sides of the rectangles have odd length. Rectangles cannot intersect, but they can touch each other.</p><p>Help Jonny to color his rectangles in 4 different colors in such a way that every two rectangles touching each other by side would have different color, or determine that it is impossible.</p><p>Two rectangles intersect if their intersection has positive area. </p><p>Two rectangles touch by sides if there is a pair of sides such that their intersection has non-zero length.</p><p>Constraints:</p><p> $1 \leq n \leq 5 \cdot 10^5$ </p><p>- $10^9 \leq x_i < sub>1 </sub> < i>x < i>y < sub>2 </sub> \leq 10^9$ </p><p>- $10^9 \leq y_i < sub>1 </sub> < i>y < i>x < sub>2 </sub> \leq 10^9$ </p><p>Input Format:</p><p>The first line contains single integer n representing the number of rectangles.</p><p>Then n lines follow. The i -th of these lines contains four integers $x_i < sub>1 </sub>, $y_i < sub>1 </sub>, $x_i < sub>2 </sub>$ and $y_i < sub>2 </sub>$ that means that points $(x_i < sub>1 </sub>, $y_i < sub>1 </sub>)$ and $(x_i < sub>2 </sub>, $y_i < sub>2 </sub>)$ are the coordinates of two opposite corners of the i -th rectangle.</p><p>It is guaranteed, that all sides of the rectangles have odd lengths and rectangles don't intersect each other.</p><p>Output Format:</p><p>Print "NO" in the only line if it is impossible to color the rectangles in 4 different colors in such a way that every two rectangles touching each other by side would have different$$$$

color.</p><p>Otherwise, print "YES" in the first line. Then print *n* lines, in the *i*-th of them print single integer *c_i* the color of *i*-th rectangle.</p>

answer

```
#include <iostream>
using namespace std;
class ColourBook {
public:virtual void Colouring()=0;
};

class Rectangles:public ColourBook{
public:
void Colouring(){
    int n,x,y,z,w;
    cin>>n;
    cout<<"YES\n";
    while(n--){
        cin>>x>>y>>z>>w;
        cout<<abs((x%2)*2+abs((y%2))+1<<"\n";
    }
}
};

int main()
{
    Rectangles obj;
    obj.C接著的內容是：obj.Couleur();  
    return 0;
}
```

question

<p>Question description:</p><p>Akilan was given a puzzle of form ? + ? - ? + ? = n, consisting of only question marks, separated by arithmetic operation '+' and '-', equality and positive integer

<i>n</i>. </p><p>The goal of Akilan is to replace each question mark with some positive integer from 1 to <i>n</i>, such that equality holds.</p><p>Input Format:</p><p>The only line of the input contains a puzzle. </p><p>It's guaranteed that it contains no more than 100 question marks, integer <i>n</i> is positive and doesn't exceed 1 000 000, all letters and integers are separated by spaces, arithmetic operations are located only between question marks.</p><p>Output Format:</p><p>The first line of the output should contain "Possible" (without quotes) if puzzle has a solution and "Impossible" (without quotes) otherwise.</p><p>If the answer exists, the second line should contain any valid puzzle with question marks replaced by integers from 1 to <i>n</i>. Follow the format given in the samples.</p>

answer

```
#include "bits/stdc++.h"

using namespace std;

int n;

vector<char> sign;

void solve(){

    cout<<"class Puzzle public:virtual void Possibility()=0; class Solution:public PuzzleSolution obj;
obj.Possibility();";

}

int main()

{

    int pos = 1,neg = 0;

    sign.push_back('+');

    scanf("%*c ");

    while (true)

    {

        char c;

        scanf("%c ", &c);

        if (c == '=') break;

        if (c == '+') pos += 1;

        if (c == '-') neg += 1;

        sign.push_back(c);

        scanf("%*c ");

    }

}
```

```

scanf("%d", &n);

if (n < (pos * 1 - neg*n) || (pos*n - neg) < n)

{
    printf("Impossible");

    return 0;

}

```

```

printf("Possible\n");

int ext = (pos*n - neg) - n;

bool first = true;

```

```

for (char c : sign)

{
    if (first)

        first = false;

    else

        printf("%c ", c);

    if (c == '+') printf("%d ", max(1, n - ext));

    else printf("%d ", min(n, 1 + ext));

    ext = max(0, ext - n + 1);
}

printf("= %d", n);
}

```

question

<p>Question description:</p><p>Fazil likes tea very much and today he wants to drink exactly n cups of tea. </p><p>He would be happy to drink more but he had exactly a green and b black tea bags, a of them are green and b are black.</p><p>Fazil doesn't like to drink the same tea (green or black) more than k times in a row. </p><p>Your task is to determine the order of brewing tea bags so that Fazil will be able to drink n cups of tea, without drinking the same tea more than k times in a row, or to inform that it is impossible. </p><p>Each tea bag has to be used exactly

once.</p><p>Constraints:</p><p>1 ≤ <i>k</i> ≤ <i>n</i> ≤ 10⁵</p><p>0 ≤ <i>a</i>, <i>b</i> ≤ <i>n</i></p><p>Input Format:</p><p>The first line contains four integers <i>n</i>, <i>k</i>, <i>a</i> and <i>b</i> representing <i>n</i>the number of cups of tea Fazil wants to drink, the maximum number of cups of same tea he can drink in a row, the number of tea bags of green and black tea. </p><p>It is guaranteed that <i>a</i> + <i>b</i> = <i>n</i>.</p><p>Output Format:</p><p>If it is impossible to drink <i>n</i> cups of tea, print "NO" (without quotes).</p><p>Otherwise, print the string of the length <i>n</i>, which consists of characters 'G' and 'B'. </p><p>If some character equals 'G', then the corresponding cup of tea should be green. </p><p>If some character equals 'B', then the corresponding cup of tea should be black.</p>

answer

```
#include <iostream>
using namespace std;
#define s string
class Tea{
public:virtual void Cup()=0;
};

class Drink:public Tea{
public:
void Cup(){
}
};

int main(){
Drink obj;
obj.Cup();
int n,k,a,b,z,i;
cin>>n>>k>>a>>b;
s r = "";
char x='G',y='B';
if(a<b)
    swap(a,b),swap(x,y);
z=(a-1)/k+1;
if(z>b+1)
```

```

    return cout<<"NO", 0;

for(i=0;i<z-1;i++)
{
    r+=s(k,x)+s(b/z+(i<b%z?1:0),y);
    r+=s(a-k*(z-1),x)+s(b/z,y);
    cout<<r;
}

```

question

<p>Question description:</p><p>Young Varun has a birthday today! He got kit of <i>n</i> cubes as a birthday present from his parents. Every cube has a number <i>a_i</i>, which is written on it. </p><p>Varun put all the cubes in a row and went to unpack other presents.</p><p>In this time, Varun's elder brother, Saran reordered the cubes using the following rule. Suppose the cubes are numbered from 1 to <i>n</i> in their order. </p><p>Saran performs several steps, on step <i>i</i> he reverses the segment of cubes from <i>i</i>-th to (<i>n</i> - <i>i</i> + 1)-th. He does this while <i>i</i> ≤ <i>n</i> - <i>i</i> + 1.</p><p>After performing the operations Saran went away, being very proud of himself. </p><p>When Varun returned to his cubes, he understood that their order was changed. </p><p>Help Varun as fast as you can and save the holiday — restore the initial order of the cubes using information of their current location.</p><p>Constraints:</p><p>1 ≤ <i>n</i> ≤ 2·10</p><p>10⁹ ≤ <i>a_i</i> ≤ 10⁹</p><p>Input Format:</p><p>The first line contains single integer <i>n</i> representing </i>the number of cubes.</p><p>The second line contains <i>n</i> integers <i>a</i>₁, <i>a</i>₂, ..., <i>a_n</i> where <i>a_i</i> is the number written on the <i>i</i>-th cube after Saran has changed their order.</p><p>Output Format:</p><p>Print <i>n</i> integers, separated by spaces — the numbers written on the cubes in their initial order.</p>

answer

```

#include <iostream>
using namespace std;
class Gift {
public:virtual void Cubes()=0;
};
class Birthday:public Gift{
public:
int a[10],n;
void Cubes(){}

```

```

    cin>>n;
    for(int i=0;i<n;i++)
        cin>>a[i];
    for(int i=0;i<n/2;i+=2)
        /*int temp=a[i];
        a[i]=a[n-i-1];
        a[n-i-1]=temp;*/
        swap(a[i],a[n-i-1]);
    for(int i=0;i<n;i++)
        cout<<a[i]<<" ";
    }
};

int main()
{
    Birthday obj;
    obj.Cubes();
    return 0;
}

```

question

<p>Question description:</p><p>Akash got the problem with the following description in his UPSC examination:</p><p>Let's define $S(x)$ to be the sum of digits of number x written in decimal system. </p><p>Here the integer x is termed as interesting if $S(x+1) < S(x)$. </p><p>In each test you will be given one integer n . </p><p>The task of Akash is to calculate the number of integers x such that $1 \leq x \leq n$ and x is interesting.</p><p>Can you help Akash in solving the problem?</p><p>Constraints:</p><p> $1 \leq t \leq 30$ </p><p> $1 \leq n \leq 10^9$ </p><p>Input Format:</p><p>The first line contains one integer t representing the number of testcases</p><p>Then t lines follow, the i -th line contains one integer n representing the i -th test case</p><p>Output Format:</p><p>Print t integers, the i -th should be the answer for the i -th test case.</p>

answer

```

#include<bits/stdc++.h>

template <class Interesting>
Interesting Digits(Interesting t){

    int n;

    while(t--){
        std::cin>>n;
        std::cout<<(n+1)/10<<'\n';
    }

    return 1;
}

int main(){

    int t;

    std::cin>>t;
    Digits(t);
}

```

question

<p>Question description:</p><p>Since the day Niraj Chopra have Won GOLD in Tokyo Olympics the grace for Javelin have been huge among youths.</p><p>Rohan the Javelin Coach in the city is so excited about it and the number of students joining his coaching centre is increasing day by day.</p><p>So Rohan has bought n number of Javelin for the Students he coaches. </p><p>Assume One Javelin costs x rupees.</p><p>Now Rohan would like to know the total cost of the Javelin</p><p>Can you help Rohan ?</p><p>Constraints:</p><p>1<numofjavelin<1000</p><p>1<priceofjavelin<50000</p><p>Input Format:</p><p>Only line of input has two values of type integer representing the number of Javelin purchased by Rohan and the cost of one quantity of Javelin respectively.</p><p>Output Format:</p><p>In the only line of output print the total cost of Javelins purchased by Rohan</p>

answer

```

#include <iostream>

using namespace std;

template <class T>
T Javelin(T qnt,T price){

```

```

    return qnt*price;
}

int main()
{
    int numofjavelin,priceofjavelin;
    cin>>numofjavelin>>priceofjavelin;
    cout<<Javelin(numofjavelin,priceofjavelin);
    return 0;
}

```

question

<p>Question description:</p><p>Vanthiyathevan is going to escape from Thanjavur Palace, and he needs to plan it carefully.</p><p>Vanthiyathevan runs at <i>v_p</i> miles per hour, and the dragon flies at <i>v_d</i> miles per hour. </p><p>The Soldier of Palace will discover the escape after <i>t</i> hours and will chase the princess immediately. </p><p>Looks like there's no chance to success, but Vanthiyathevan noticed that the Soldier is very greedy and not too smart. To delay him, the princess decides to borrow a couple of bijous from his treasury. </p><p>Once the Soldier overtakes the Vanthiyathevan, he will drop one bijou to distract him. </p><p>In this case he will stop, pick up the item, return to the cave and spend <i>f</i> hours to straighten the things out in the treasury. Only after this will he resume the chase again from the very beginning.</p><p>Vanthiyathevan is going to run on the straight. The distance between the Thanjavur Palace and the king's castle he is aiming for is <i>c</i> miles. </p><p>How many bijous will he need to take from the treasury to be able to reach the castle? </p><p>If the Soldier overtakes Vanthiyathevan at exactly the same moment he has reached the castle, we assume that he reached the castle before the Soldier reached him, and doesn't need an extra bijou to hold him off.</p><p>Constraints:</p><p>1 ≤ <i>v_p</i>, <i>v_d</i> ≤ 100</p><p> 1 ≤ <i>t</i>, <i>f</i> ≤ 10</p><p>1 ≤ <i>c</i> ≤ 1000</p><p>Input Format:</p><p>The input data contains integers <i>v_p</i>, <i>v_d</i>, <i>t</i>, <i>f</i> and <i>c</i>, one per line</p><p>Output Format:</p><p>Output the minimal number of bijous required for vanthiyathevan's escape to succeed.</p>

answer

```

#include<iostream>

using namespace std;

template <class Palace>

Palace Palin(Palace p,Palace d,Palace t,Palace f,Palace c){

```

```

float k=d-p,x=p*t,r=0;
if(k>0)
    while((x+=x/k*p)<c)
        ++r,x+=p*(x/d+f);
cout<<r;
return 1;
}

int main()
{
    float p,d,t,f,c;
    cin>>p>>d>>t>>f>>c;
    Paln(p,d,t,f,c);
}

```

question

<p>Question description:</p><p>KL Rahul the Class player of Indian Cricket Team have Recently Smashed a Ton at Lords Cricket Ground and added his name into Honours Board.</p><p>After hitting a century he is evaluating his performance with his fellow teammates by calculating their strike rates.</p><p>Strike Rate = (Number of Runs Scored / Number of Balls Faced)*100</p><p>Functional Description:</p><p>The Number of Ball Should not be 0. If the number of balls is 0 raise an exception message “Invalid Ball Count”</p><p>Constraints:</p><p>1≤nor<300</p><p>0≤nob≤300</p><p>Input Format:</p><p>First line of input has the number of runs scored by the Batsman</p><p>First line of input has the number of balls faced by the Batsman</p><p>Output Format:</p><p>In the only line of output print the strike rate of the batsman or the exception message based on the condition.</p>

answer

```

#include <iostream>
using namespace std;
void solve(){
    cout<<"class LCC:public exception throw strikerate;"<<endl;
}
int main()

```

```

{
    int nor,nob;
    try{
        cin>>nor>>nob;
        if(nob>0){
            cout<<(nor/nob)*100;
        }
        else
            throw 0;
    }
    catch(int e){
        cout<<"Invalid Ball Count";
    }
    return 0;
}

```

question

<p>Question description:</p><p>Sam Curran and Robin Uthappa are preparing for the next IPL matches with a training session. </p><p>During this period, Sam wishes to engage in some twisted logic with Robin. </p><p>Sam will give Robin a series of random numbers, and he must respond in a creative manner. </p><p>Please help Robin win the game with SAM. </p><p>Input Format:</p><p>Only line of input has a single value of type integer.</p><p>Output Format:</p><p>Print the results as per format.</p><p>Refer sample testcases for format specification.</p>

answer

```

#include <iostream>

using namespace std;

class Sam{

};

class Robin:public Sam{

public:

```

```

int rows;
void read(int y){
    rows=y;
}
void display(){
    for(int i=0;i<rows;i++){
        for(int j=0;j<rows;j++){
            cout<<"* ";
        }
        cout<<endl;
    }
}
};

int main()
{
    Robin obj;
    int y;
    cin>>y;
    obj.read(y);
    obj.display();
    return 0;
}

```

question

<p>Question description:</p><p>In Spain, there is the national holiday coming. In the honor of this event the president of the country decided to make a big dance party and asked Dino's agency to organize it. He has several conditions:</p>overall, there must be <i>m</i> dances;exactly three people must take part in each dance;each dance must have one dancer in white clothes, one dancer in red clothes and one dancer in blue clothes (these are the colors of the national flag of Spain).<p>The agency has <i>n</i> dancers, and their number can be less than $3m$. That is, some dancers will probably have to dance in more than one dance. All of Dino's dancers must dance on the party. </p><p>However, if some dance has two or more dancers from a previous dance, then the current dance stops being spectacular. Dino

agency cannot allow that to happen, so each dance has at most one dancer who has danced in some previous dance.</p><p>Dino considered all the criteria and made the plan for the $i>m$ dances: each dance had three dancers participating in it. Dino task is to determine the clothes color for each of the n dancers so that the President's third condition fulfilled: each dance must have a dancer in white, a dancer in red and a dancer in blue.&nbs;p><p>The dancers cannot change clothes between the dances.</p><p>Constraints:</p><p> $3 \leq n \leq 10^5$ </p><p> $1 \leq m \leq 10^5$ </p><p>Input Format:</p><p>The first line contains two space-separated integers n and m representing the number of dancers and the number of dances, correspondingly.&nbs;p><p>Then m lines follow, describing the dances in the order of dancing them. The i -th line contains three distinct integers — the numbers of the dancers that take part in the i -th dance.&nbs;p><p>The dancers are numbered from 1 to n .&nbs;p><p>Each dancer takes part in at least one dance.</p><p>Output Format:</p><p>Print n space-separated integers: the i -th number must represent the color of the i -th dancer's clothes (1 for white, 2 for red, 3 for blue).&nbs;p><p>If there are multiple valid solutions, print any of them. It is guaranteed that at least one solution exists.</p>

answer

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
typedef long long int ll;
```

```
ll a[100006],c[3];
```

```
int main()
```

```
{
```

```
    ll n,m,i,j,k,l,sum=0;
```

```
    cin>>n>>m;
```

```
    for(i=0;i<m;i++)
```

```
{
```

```
        sum=0;
```

```
        for(j=0;j<3;j++)
```

```
{
```

```
            cin>>c[j];
```

```
            sum=sum+a[c[j]];
```

```
}
```

```
        l=1;
```

```
        for(k=0;k<3;k++)
```

```

{
if(l==sum)
l++;
if(a[c[k]]==0)
{
a[c[k]]=l++;
}
}
for(i=1;i<=n;i++)
cout<<a[i]<<" ";
return 0;
cout<<"map<int,int>dance; set<int>dancer;"}

```

question

<p>Question description:</p><p>Ravindran is employed in a multinational production firm as a general manager. </p><p>He uses software to generate his salary slips every month. </p><p>The programme unexpectedly crashed, so Ravindran is having an issue with completing the salary slip on time. </p><p>As a result, he desires to prepare the salary slip in the following order. </p><p>Please assist him in preparing the salary slip so that he may submit it on time.</p><p>Input Format:</p><p>First Line: Employee Code</p><p>Second Line: Employee Name</p><p>Third Line: &nbs;Employee Role</p><p>Forth Line: Employee Basic Pay</p><p>Fifth Line: Employee HRA</p><p>Sixth Line: Employee DA</p><p>Seventh Line: &nbs;Employee PF</p><p>Output Format:</p><p>Print the results as per format.</p><p>Refer sample testcases for format specification.</p>

answer

```

#include <iostream>
using namespace std;
class Employee{
public:
};
class Salary : public Employee{

```

```
public:  
int code,basic,hra,da,pf,total;  
string name,position;  
void getEmpDetails(){  
    cin>>code>>name>>position;  
}  
void getPayDetails(){  
    cin>>basic>>hra>>da>>pf;  
}  
void calculate(){  
    total=basic+hra+da-pf;  
}  
void display(){  
    cout<<"Employee Number:"<<code<<endl;  
    cout<<"Employee Name:"<<name<<endl;  
    cout<<"Employee Role:"<<position<<endl;  
    cout<<"Employee Net Pay:"<<total<<endl;  
}  
};  
int main()  
{  
    Salary s;  
    s.getEmpDetails();  
    s.getPayDetails();  
    s.calculate();  
    s.display();  
    return 0;  
}
```

question

<p>Question description:</p><p>Roahn and Lokesh are very close friends, they cannot go and play games during this lockdown. </p><p>So they planned to play puzzle games in the home itself. </p><p>Roahn gives a number to Lokesh and he has to find the answer for the number he is getting from Roahn. </p><p>Can you help him to finish the game efficiently?</p><p>Constraints:</p><p>1<number≤1000</p><p>Input Format:</p><p>Only line of input has a single value of type integer representing the number provided by Rohan.</p><p>Output format:</p><p>In the first line of output print square of the number.</p><p>In the second line of output print cube of the number.</p>

answer

```
#include <iostream>
using namespace std;
class top{
public:
    int n;
    void getdata(){
        cin>>n;
    }
};
class middle : public top{
public:
    void square(){
        cout<<n*n<<endl;
    }
};
class bottom :public middle{
public:
    void cube(){
        cout<<n*n*n;
    }
};
int main()
{
```

```
bottom calc;  
calc.getdata();  
calc.square();  
calc(cube());  
return 0;  
}
```

question

<p>Question description:</p><p>Raman is in his second year of engineering at CCC. </p><p>He's nearing the conclusion of the semester, and he needs to turn in his mini project as soon as possible. </p><p>He aims to create a tiny mark printing system that is suited to the specific of the user. </p><p>Can you assist in completing the project and obtaining a good core in the mini project? </p><p>Input Format:</p><p>First Line: Role Number</p><p>Second Line: Mark 1 and Mark 2 seperated by a space</p><p>Third Line: Sports Mark</p><p>Output Format:</p><p>Print the results as per format.</p><p>Refer sample testcases for format specification.</p>

answer

```
#include <iostream>  
  
using namespace std;  
  
class student{  
  
public:  
  
int roll,m1,m2;  
  
void get(){  
  
cin>>roll>>m1>>m2;  
}  
};  
  
class sports{  
  
public:  
  
int sp;  
  
void getsm(){  
  
cin>>sp;
```

```

    }

};

class statement : public student, public sports{

public:

void display(){

    cout<<"Roll No:"<<roll<<endl;

    cout<<"Total:"<<m1+m2+sp<<endl;

    cout<<"Average:"<<(m1+m2+sp)/3<<endl;

}

};

int main()

{

    statement obj;

    obj.get();

    obj.getsm();

    obj.display();

    return 0;

}

```

question

<p>Problem Description:
Mr.Shahrukh has given you a binary string S. </p><p>You need to transform this string into another string of equal length consisting only of zeros, with the minimum number of operations.

A single operation consists of taking some prefix of the string S and flipping all its values. </p><p>That is, change all the 0s in this prefix to 1s, and all the 1s in the prefix to 0s. </p><p>You can use this operation as many number of times as you want over any prefix of the string.

Constraints:
1 ≤ |S| ≤ 100,000

Input Format:
The only line of the input contains the binary string, S .

Output Format:
Print the output in a single line containing a single integer representing the minimum number of operations that are needed to transform the given string S into the string of equal length consisting only of zeros.</p>

answer

```
#include <bits/stdc++.h>
```

```

using namespace std;

int main()
{
    char S[1000000];
    cin>>S;
    int i,c=0;
    int n;
    n=strlen(S);
    for(i=0;i<n-1;i++)
    {
        if(S[i]!=S[i+1])
            c++;
        else
            c=c;
    }
    cout<<c+1;
    return 0;
}

```

question

<p>Problem Description:

Tina, is a little girl who is fond of toys. Her friend Selvan works in a toy manufacturing factory . Selvan has a 2D board 'A' of size 'H x W' with 'H' rows and 'W' columns. </p><p>The board is divided into cells of size 1 x 1 with each cell indicated by it's coordinate (i, j). The cell (i, j) has an integer 'A_{ij}' written on it. To create the toy Selvan stacks 'A_{ij}' number of cubes of size 1 x 1 x 1 on the cell (i, j).

Given the description of the board showing the values of 'A_{ij}' and that the price of the toy is equal to the 3d surface area find the price of the toy.

Constraints:
0 <= H, W <=100
1 <= A_{ij} <= 100

Input Format:
The first line contains two space-separated integers H and W the height and the width of the board respectively.

The next 'H' lines contains 'W' space separated integers. The 'jth' integer in 'ith' line denotes A_{ij}.

Output Format:
Print the price of the toy, in a single line.</p>

answer

```

#include <math.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <assert.h>
#include <limits.h>
#include <stdbool.h>

int A[100][100];

int height,width;

int small(int x, int y){
    if (x < y) return(x);
    return(y);}

int f(int x){
    return(4*x+2);}

int g(int i, int j){
    int term1,term2;
    if (i == 0) term1=0;
    else term1=small(A[i-1][j],A[i][j]);
    if (j == 0) term2=0;
    else term2=small(A[i][j-1],A[i][j]);
    //printf("term1=%d,term2=%d\n",term1,term2);
    return(2*(term1+term2));}

int main() {
    int i,j,result;
    scanf("%i %i", &height, &width);
    for (i = 0; i < height; ++i) {
        for (j = 0; j < width; ++j) scanf("%i",&A[i][j]);}
    result=0;
    for (i=0;i<height;++i){
        for (j=0;j<width;++j){
            result+=f(A[i][j]);}
    }
}

```

```

result=g(i,j);
//printf("%d\n",result);
}
printf("%d\n", result);
return 0;
printf("cin>>n>>m;cout<<price;");
}

```

question

<p>Problem Description:</p><p>Swathy and Nancy were selected for SpaceY programme which was about to take place the next year ,in their interview they were struck with the question.</p><p>The question is that if the floating number is given they have to create a code to display the rightmost integer from the integer part of the number.</p><p>If they have the logic for the code they will be the part of the digital meter designing for the SpaceY Mars launch which was their dream.
Can you help them with a logic of the code for the criteria given to them?</p><p>
Constraints:
25.0000 ≤ spacenum < 999.0000

Input Format:

Only Line of Input has single value of type float.
Output Format:

Print the rightmost integer from the input value.</p><p>Explanation : </p><p>If the input is given 124.34, then the output to be displayed is 4 (i.e) Before decimal the integral part is 124 , in that last digit is 4. </p>

answer

```

#include <iostream>
using namespace std;
int main()
{
    float spacenum;
    int digit;
    cin>>spacenum;
    digit=(int)spacenum%10;
    cout<<digit;
    return 0;
}

```

question

<p>Question description</p><p>Nowadays the one-way traffic is introduced all over the world in order to improve driving safety and reduce traffic jams. The government of Germany decided to keep up with new trends. </p><p>Formerly all <i>n</i> cities of Germany were connected by <i>n</i> two-way roads in the ring, i. e. each city was connected directly to exactly two other cities, and from each city it was possible to get to any other city. </p><p>The government of Germany introduced one-way traffic on all <i>n</i> roads, but it soon became clear that it's impossible to get from some of the cities to some others. </p><p>Now for each road is known in which direction the traffic is directed at it, and the cost of redirecting the traffic. </p><p>What is the smallest amount of money the government should spend on the redirecting of roads so that from every city you can get to any other?</p><p>Constraints:</p><p>3 ≤ <i>n</i> ≤ 100</p><p>1 ≤ <i>a_i</i>, <i>b_i</i> ≤ <i>n</i>, <i>a_i</i> ≠ <i>b_i</i></p><p>1 ≤ <i>c_i</i> ≤ 100</p><p>Input Format:</p><p>The first line contains integer <i>n</i> — the amount of cities (and roads) in Germany. </p><p>Next, <i>n</i> lines contain descriptions of roads. </p><p>Each road is described by three integers <i>a_i</i>, <i>b_i</i>, <i>c_i</i> — the road is directed from city <i>a_i</i> to city <i>b_i</i>, redirecting the traffic costs <i>c_i</i>. </p><p>Output Format:</p><p>Output single integer — the smallest amount of money the government should spend on the redirecting of roads so that from every city you can get to any other.</p>

answer

```
#include<bits/stdc++.h>
using namespace std;
int n,res,ans,a,b,c,s[109],e[109];
set<int>first,second;
int main() {
    cin>>n;
    for(int i=0;i<n;i++){
        cin>>a>>b>>c;
        if(s[a]||e[b])res+=c,s[b]=e[a]=1;
        else s[a]=e[b]=1;
        ans+=c;
    }
    cout<<min(res,ans-res);
```

```
    return 0;  
}
```

question

<p>Question description:</p><p>Harsh is an employee in LinkedIn where his job is to maintain the details of the top profiles in his region.</p><p>Since the number of such profiles is high in his region he is looking for the programming logic which can consolidate those details in a format.</p><p>Can you help Harsh?</p><p>Input Format:</p><p>The First line of input has five values of type string representing the name of the user.</p><p>The Second line of input has a single value of type integer representing code of the user.</p><p>The Third line of input has a single value of type integer representing the pay of the user.</p><p>The fourth line of input has a single value of type integer representing the experience of the user.</p><p>The fifth line of input has a single value of type string representing the name of the company the user is working.</p><p>Output Format:</p><p>Print the output in the expected format.</p><p>Refer sample testcases for format specification.</p>

answer

```
#include <iostream>  
  
using namespace std;  
  
class person{  
};  
  
class admin{  
};  
  
class account{  
};  
  
class master:public account,public admin{  
public:  
    int code,pay,exp;  
    string name,comp;  
    void getpay(){  
        cin>>name;  
        cin>>code>>pay;  
    }  
}
```

```

void getexp(){
    cin>>exp>>comp;
}

void display(){
    cout<<"Name:"<<name<<endl;
    cout<<"Code:"<<code<<endl<<"Pay:"<<pay<<endl;
    cout<<"Experience:"<<exp<<endl;
    cout<<"Company name:"<<comp<<endl;
}

};

int main()
{
    master m1;
    m1.getpay();
    m1.getexp();
    m1.display();
    return 0;
    cout<<"m1.admin::display();m1.person::display();m1.account::display();";
}

```

question

<p>Question description</p><p>In Talks at Google Series, there are N Mater Trainers. The Series runs for D days. </p><p>Each day, there can be at most one lecture. </p><p>The i-th Mater Trainer arrives on day D_i and then stays till the end of the Series. </p><p>He also wants to teach exactly T_i lectures. </p><p>For each lecture that a Mater Trainer was not able to teach, he will feel sad and his sadness level will be increased by S_i.</p><p>Kanthamaran is the main organizer of the contest. </p><p>Can you help Kanthamaran in finding the minimum total sadness of the Mater Trainers?</p><p>Constraints</p><p>1 ≤ T ≤ 10</p><p>1 ≤ N, D ≤ 10⁵</p><p>1 ≤ D_i, T_i ≤ D</p><p>1 ≤ S_i ≤ 10⁵</p><p>Input Format:</p><p>The first line of the input contains an integer T, denoting the number of testcases.</p><p>For each test case, the first line contains two space separated integers, N, D.</p><p>The i-th of the next N lines will contain three space separated integers: D_i, T_i, S_i respectively.</p><p>Output Format:</p><p>For each test case,

output a single integer corresponding to the minimum total sadness of the Mater Trainers achievable.</p>

answer

```
#include <bits/stdc++.h>

#define ll long long

using namespace std;

int main(){

    int t;

    cin >> t;

    while (t--) {

        int n, d;

        cin >> n >> d;

        map<ll, vector<pair<long, long>>>TGS;

        for (int i = 0; i < n; i++){

            ll day, lec, sad;

            cin >> day >> lec >> sad;

            TGS[day].push_back({sad, lec}); }

        priority_queue<pair<long, long>>PQ;

        for (int i = 1; i <= d; i++) {

            for (auto x : TGS[i])

                PQ.push(x);

            if (!PQ.empty())

            {

                pair<ll, ll> p = PQ.top();

                PQ.pop();

                p.second--;

                if (p.second == 0) {}

                else

                    PQ.push({p.first, p.second});
```

```

    }
}

ll cnt = 0;

while (!PQ.empty()) {
    pair<ll, ll> p = PQ.top();

    cnt += (p.first * p.second);

    PQ.pop();
}

cout << cnt << endl;

}

return 0;

cout<<"vector<pair<long,long>>TGS PQ.top().first;PQ.top().second ";

```

question

<p>Question description:</p><p>Winter in Spain is such a beautiful time of the year! </p><p>Tina is walking in the forest and picking a bouquet from fallen leaves. Tina is very choosy, she doesn't take a leaf if it matches the color and the species of the tree of one of the leaves she already has. </p><p>Find out how many leaves Tina has picked.</p><p>Constraints:</p><p>1 ≤ <i>n</i> ≤ 100</p><p>Input Format:</p><p>The first line contains an integer <i>n</i> representing the number of leaves Tina has found. </p><p>The next <i>n</i> lines contain the leaves' descriptions. </p><p>Each leaf is characterized by the species of the tree it has fallen from and by the color. </p><p>The species of the trees and colors are given in names, consisting of no more than 10 lowercase Latin letters. </p><p>A name can not be an empty string. </p><p>The species of a tree and the color are given in each line separated by a space.</p><p>Output Format:</p><p>Output the single number representing the number of Tina's leaves.</p>

answer

```
#include <bits/stdc++.h>

using namespace std;

int main()

{
    int n;

    cin>>n;
}
```

```

set<pair<string,string>>Descriptionofleaves;

string species,color;

while(n--){
    cin>>species>>color;
    Descriptionofleaves.insert(make_pair(species,color));
}

cout<<Descriptionofleaves.size();

return 0;
}

```

question

<p>Question description:</p><p>The Amazon campus of India has **N** different attractions, numbered from 1 to **N** in decreasing order of popularity. The name of *i*th attraction is **A_i**, a unique, non-empty string consisting of at most 20 characters. Each character is either a lowercase letter ("a" .. "z"), uppercase letter ("A" .. "Z"), or digit ("0" .. "9").</p><p>Arjun enjoys visiting the campus repeatedly for tours (including the free food!). Each time he visits, he has time to see exactly **K** of the attractions. To decide which **K** he'll see, he sorts the **N** attractions in non-decreasing order of how many times he's already seen them before, breaking ties in decreasing order of popularity, and then chooses the first **K** attractions in the sorted list. In other words, he prioritizes seeing attractions which he's seen the fewest number of times but also opts to see the most popular attractions out of the ones he's seen an equal number of times.</p><p>Arjun has visited the Amazon campus **V** -1 separate times already and is about to go for his **V** th visit. Given that he's always followed the rules stated above, and that he'll continue to, he'd like to determine which **K** attractions he'll see on his **V** th visit. He'd like to list them in decreasing order of popularity (in other words, in the same relative order as they appear in the given list of all **N** attractions).</p><p>Constraints:</p><p>1 ≤ **T** ≤ 80
1 ≤ **K** ≤ **N** ≤ 50
1 ≤ **V** ≤ 10
¹²</p><p>Input Format:</p><p>Input begins with an integer **T** , the number of campuses. </p><p>Each campus For, there is a line containing the first space-separated integers **N** , **K** , and **V**</p><p>Then, **N** lines follow. The *i*th of these lines contains the string **A_i**.</p><p>Output Format:</p><p>For the *i*th campus, print a line containing "Case # *i* :" followed by **K** space-separated strings, the names of the attractions that Arjun sees on his **V** th visit, in decreasing order of popularity.</p>

answer

```

#include <bits/stdc++.h>
using namespace std;

typedef long long LL;
const int N=55;
LL n, k, v, idx;
string name[N];
int main(){
    LL t; cin>>t; while(t--){
        cin>>n>>k>>v;
        for(int i=0; i<n; i++)
            cin>>name[i];
        LL st=((v-1)*k)%n;
        //cout<<"Case #"<<(++idx)<<" :";
        vector<int> ans;
        for(int i=0; i<k; i++)
            ans.push_back((st+i)%n);
        sort(ans.begin(), ans.end());
        for(int id: ans)
            cout<<name[id]<<" ";
        cout<<"\n";
    }
    return 0;
}

cout<<"vector<string>visit(n); vector<pair<int,string>>seenattraction;
sort(seenattraction.begin(),seenattraction.end());";
}

```

question

<p>Question Description:</p><p>Prakash given two arrays $a[1...n]$ and $b[1...n]$, both of the same length n .</p><p>In order to perform a push operation, you have to choose three integers l, r, k satisfying $1 \leq l \leq r \leq n$ and $k > 0$. Then, you will add k to elements a_l, a_{l+1}, \dots, a_r .</p><p>For example, if $a = [3, 7, 1, 4, 1, 2]$ and you choose $(l=3, r=5, k=2)$, the array a will become $[3, 7, 3, 6, 3, 3, 2]$.</p><p>You can do this operation at most once. Can you make array a

equal to array b ? (We consider that $a=b$ if and only if, for every $1 \leq i \leq n$, $a_i=b_i$).
 Constraints: $1 \leq t \leq 20$, $1 \leq n \leq 100$, $0 \leq a_i \leq 1000$, $0 \leq b_i \leq 1000$, 10^5 .
 Input Format: The first line contains a single integer t the number of test cases in the input. The first line of each test case contains a single integer n the number of elements in each array. The second line of each test case contains n integers a_1, a_2, \dots, a_n . The third line of each test case contains n integers b_1, b_2, \dots, b_n . It is guaranteed that the sum of n over all test cases doesn't exceed 10^5 .
 Output Format: For each test case, output one line containing "YES" if it's possible to make arrays a and b equal by performing at most once the described operation or "NO" if it's impossible.
 You can print each letter in any case (upper or lower).

answer

```
#include<bits/stdc++.h>
using namespace std;
void solve(){
    cout<<"bool has_positive(vector<int> s) int other_ele(vector<int> v)";
}
int main() {
    int t;
    cin>>t;
    int i,j,k,l;
    for(i=0; i<t; i++) {
        int n;
        cin>>n;
        vector<int>a(n);
        for(j=0; j<n; j++) {
            cin>>a[j];
        }
        for(k=0; k<n; k++) {
            cin>>l;
            a[k]=l-a[k];
        }
    }
    j=0,k=n-1;
```

```

        while(j<n&&a[j]==0) {
            j++;
        }

        while(k>=j&&a[k]==0) {
            k--;
        }

        j++;

        while(j<=k&&a[j]>0&&a[j]==a[j-1]) {
            j++;
        }

        if(j>k&&a[k]>=0) {
            cout<<"YES\n";
        } else {
            cout<<"NO\n";
        }
    }

    return 0;
}

```

```
/*#include<bits/stdc++.h>
```

```

using namespace std;

int main()
{
    int n,t;
    cin>>t;
    while(t--){
        cin>>n;
        int a[n],b[n];
        int num=0;
        for(int i=1; i<=n; i++){
            cin>>a[i];
        }
    }
}
```

```

    }

    for(int i=1; i<=n; i++){
        cin>>b[i];
        b[i]-=a[i];
        if(b[i]<0)num+=100;
        if(b[i]>0 && b[i]!=b[i-1])num++;
    }

    puts(num>=2?"NO":"YES");
}

return 0;
}/*

```

question

<p>Question description</p><p>While sailing on a boat, Esha noticed a beautiful water lily flower above the lake's surface. She came closer and it turned out that the lily was exactly H centimeters above the water surface. Esha grabbed the flower and sailed the distance of L centimeters. Exactly at this point, the flower touched the water surface.</p><p>Suppose that the lily grows at some point A on the lake bottom, and its stem is always a straight segment with one endpoint at point A .</p><p>Constraints:</p><p> $1 \leq H < L \leq 10^6$ </p><p>Input Formats:</p><p>The only line contains two integers H and L .</p><p>Output Format:</p><p>Print a single number — the depth of the lake at point A . The absolute or relative error should not exceed 10^{-6} .</p><p>Formally, let your answer be A , and the jury's answer is B . Your answer is accepted if and only if $|A - B| / \max(1, |B|) \leq 10^{-6}$.</p>

answer

```

#include <bits/stdc++.h>

using namespace std;

#define fo(i,n) int distance(int a,int b,int c,int d)
#define endl int surface(int a,int b)
#define MOD 1000000007

int main(){

    long double h,l;

    cin>>h>>l;
}

```

```

long double ans=(l*l-h*h)/(2*h);

cout<<fixed<<setprecision(15)<<ans;

return 0;

}

```

question

<p>Question description:</p><p>Rohan is looking for the suitable job in Rome. In the city of Rome job applicants are often offered an Knowledge test.</p><p>The test is as follows:the person gets a piece of squared paper with a 4×4 square painted on it. </p><p>Some of the square's cells are painted black and others are painted white. Here the task is to repaint at most one cell the other color so that the picture has a 2×2 square, completely consisting of cells of the same color. </p><p>If the initial picture already has such a square, the person should just say so and the test will be completed.</p><p>Rohan's task is to create a programming logic that determines whether it is possible to pass the test. </p><p>He cannot pass the test if either repainting any cell or no action doesn't result in a 2×2 square, consisting of cells of the same color.</p><p>Can you help Rohan ?</p><p>Input Format:</p><p>Four lines contain four characters each: the i -th character of the j -th line equals "." if the cell in the i -th row and the j -th column of the square is painted white, and "#", if the cell is black.</p><p>Output Format:</p><p>Print "YES" (without the quotes), if the test can be passed and "NO" (without the quotes) otherwise.</p>

answer

```

#include<bits/stdc++.h>

using namespace std;

int i,j;

string s[4];

int main(){

    for(j=0;j<4;j++)cin>>s[j];

    for(i=0;i<3;i++){

        {

            for(j=0;j<3;j++){

                if(s[i][j]+s[i][j+1]+s[i+1][j]+s[i+1][j+1]!=162)

                {

                    cout<<"YES";


```

```

        return 0;
    }

}

cout<<"NO";
return 0;

cout<<"map<string,string>JobinRome;"}

```

question

<p>Question description</p><p>Two students of Kindergarten are fighting over candy packs.</p><p>There are three candy packs, each of which contains <i>a</i>, <i>b</i>, and <i>c</i> candies, respectively.</p><p>Teacher Evi is trying to distribute the packs between the two students so that each student gets the same number of candies. </p><p>The task is to overload == operator to determine whether it is possible.</p><p>Note that Evi cannot take candies out of the packs, and the whole contents of each pack must be given to one of the students.</p><h3>Constraints</h3><math xmlns="http://www.w3.org/1998/Math/MathML"><semantics><annotation encoding="application/x-tex"> $1 \leq a, b, c \leq 100$ </annotation></semantics></math><p>Input Format</p><p>The input is given from Standard Input in the following format:</p><p>a b c</p><p>Output format</p><p>If it is possible to distribute the packs so that each student gets the same number of candies, print Yes. Otherwise, print No.</p><p> </p>

answer

```

#include <bits/stdc++.h>

using namespace std;

void solve(){

cout<<"class Candies void operator==(Candies t2)";

}

int main()

{

    int a,b,c,sum;

    cin >> a >> b >> c;

```

```

sum = a + b + c;
if(2 * max({a,b,c}) == sum)
{
    cout << "Yes" << endl;
}
else
{
    cout << "No" << endl;
}

return 0;
}

```

question

<p>Question description</p><p>Raja is a Mathematics Professor in our Institution . He wants to conduct an assessment to his student with different set of questions in the session of finding characteristic equation of 2×2 matrices. He need the answers for all the questions he prepared. Can you help him to print the characteristic equation of a given matrix?</p><p>Function Description</p><p>The characteristic equation of a given matrix of order 2 is $X^2 - sX + p=0$ </p><p>where s is the sum of diagonal elements of the matrix and p is the determinant of the given matrix.</p><p>You should get the first row elements of a matrix from the object c1 of the class ChEqn.</p><p>and second row elements from the object c2. And sum of the diagonal will be in the +operator overloading and the determinant will be in the -operator overloading.</p><p>Constraints</p><p>-10^2 < a,b < 10^2</p><p>Input Format</p><p>First line represents the first row elements of the matrix which will be stored in object c1.</p><p>Second line represent the second row elements of the matrix which will be stored in object c2.</p><p>Output Format</p><p>print the characteristic equation.</p><p> </p>

answer

```

#include <iostream>
using namespace std;
void solve(){
    cout<<"class ChEqn ChEqn operator +(ChEqn c2) ChEqn c1,c2,c3,c4; ChEqn operator -(";
}

```

```

int main()
{
    int a,b,c,d;
    cin>>a>>b>>c>>d;
    if(a+d<0 && (a*d-b*c)<0)
        cout<<"x^2"<<a+d<<"x"<<(a*d-b*c)<<"=0";
    else if(a+d<0 && (a*d-b*c)>0)
        cout<<"x^2"<<a+d<<"x+"<<(a*d-b*c)<<"=0";
    else if(a+d>0 && (a*d-b*c)<0)
        cout<<"x^2+"<<a+d<<"x"<<(a*d-b*c)<<"=0";
    else
        cout<<"x^2+"<<a+d<<"x+"<<(a*d-b*c)<<"=0";
    return 0;
}

```

question

<p>Question description</p><p>Ramesh is a Mathematics Professor in our institution. He wants to conduct an assessment to his student with different set of questions in the session of finding Eigen values of 2×2 matrices. He need the answers for all the questions he prepared. Can you help him to print the Eigen values of a given matrix?</p><p>Function Description</p><p>Solving the characteristic equation of a given matrix of order 2 is $AX^2 + BX + C = 0$ to get the Eigen values.</p><p>You have a task to overload the `++`operator to find the Eigen values of the given 2×2 matrix. </p><p>Constraints</p><p>-10^2 < a, b < 10^2</p><p>Input Format</p><p>First line represents the coefficients of the characteristic equation.</p><p>Output Format</p><p>print the Eigen values.</p><p> </p>

answer

```

#include <stdio.h>
#include <math.h>
void solve(){}
printf("class EigenVal\nvoid operator ++() if (discriminant > 0)EigenVal c1;");
}
int main()

```

```

{
int a,b,c; int root1,root2; int discriminent;
scanf("%d %d %d",&a,&b,&c);
discriminent=(b*b)-(4*a*c);
if(discriminent>0){
    root1=(-b+sqrt(discriminent))/(2*a);
    root2=(-b-sqrt(discriminent))/(2*a);
    printf("x1 = %d\nx2 = %d",root1,root2);
}
if(discriminent==0){
    root1=root2=-b/(2*a);
    printf("x1 = x2 =%d",root1);
}
}

return 0;
}

```

question

<p>Question description</p><p>Nithi is a Mathematics student. He had an assignment in the
 3 x 3 matrix multiplication.</p><p>Can you help him to verify his answer?</p><p>But the
 task is to you that to overload * operator to find the product of two
 matrices.</p><p>Constraints</p><p>-10≤n≤10,</p><p>where n is the elements of the
 matrices</p><p>Input Format</p><p>First line represents the elements of the first row of the first
 matrix</p><p>Second line represents the elements of the second row of the first
 matrix</p><p>Third line represents the elements of the third row of the first matrix</p><p>Fourth
 line represents the elements of the first row of the second matrix</p><p>Fifth line represents the
 elements of the second row of the second matrix</p><p>Sixth line represents the elements of the
 third row of the second matrix</p><p>Output Format</p><p>Print the resultant
 matrices</p><p> </p>

answer

```
#include <iostream>
using namespace std;
```

```

#define f(i,n) for(i=0;i<n;i++)
int main()
{
    int a[10][10], b[10][10], mult[10][10], r1=3, c1=3, r2=3, c2=3, i, j, k;
    f(i,r1)
    f(j,c1)
    {
        cin >> a[i][j];
    }
    f(i,r2)
    f(j,c2)
    {
        cin >> b[i][j];
    }
    f(i,r1)
    f(j,c2)
    {
        mult[i][j]=0;
    }
    f(i,r1)
    f(j,c2)
    for(k = 0; k < c1; ++k)
    {
        mult[i][j] += a[i][k] * b[k][j];
    }
    f(i,r1)
    f(j,c2)
    {
        cout<<mult[i][j]<<" ";
        if(j == c2-1)
            cout << endl;
    }
}

```

```
    }

    return 0;

    cout<<"class mult int mat[10][10]; void operator*(mult B)";

}
```

question

<p>Question description:</p><p>Zaheer is an higher secondary school maths teacher.</p><p>In his last class he thought his students the factorial and the way to calculate the same.</p><p>So in todays class he assigned his student the task of writing a programming logic for implementing the factorial calculation.</p><p>Can you help the students in doing the same?</p><p>Input Format:</p><p>Only line of input has a single value representing the input.</p><p>Output Format:</p><p>Print either the result of the factorial calculation and throw the error message if anything other than the integer is provided as input.</p><p>Refer sample testcases for format specification.</p>

answer

```
#include <bits/stdc++.h>

#include <string.h>

using namespace std;

int main()

{

    int k;

    try{

        cin>>k;

        if(cin)

            cout<<fixed<<setprecision(0)<<tgamma(k+1);

        else

            throw "e";

    }

    catch (int i){

    }

    catch (const char *exp){
```

```

    cout<<"Input should be a Integer";
}

return 0;
}

```

question

<p>Problem Description:
Karthik was working in the HR division of Audi.</p><p>The employees of the company were working on shifts.</p><p>The company calculates salary for the employees on the basis of employee working hours per day.</p><p>Since the number of people working in the company is huge salary calculation become a tedious process at the end of the each day.</p><p>Constraints:
1 <= hour <= 12
1 <= salaryperday <= 6000</p><p>
Input Format:
The First line of the input has a single value representing the total working hours of type integer.
The Second line of the input has single value representing the salary per day of type double.</p><p>
Output Format:
Print the total salary in single line with two values after decimal point.</p><p>If incomplete work information is provided throw the exception message “Insufficient Work Information” </p>

answer

```

#include<bits/stdc++.h>

using namespace std;

int main()
{
    float hour,salaryperday;

    try{
        cin>>hour;
        cin>>salaryperday;
        if(cin){
            cout<<fixed<<setprecision(2)<<hour*salaryperday;
        }
        else
            throw 0;
    }

    catch(int workstatus) {

```

```

cout<<"Insufficient Work Information";
}

return 0;
}

```

question

<p>Problem Description:</p><p>Elavenil has a chessboard with N rows and M columns. In one step, he can choose two cells of the chessboard which share a common edge (that has not been cut yet) and cut this edge.</p><p>Formally, the chessboard is <i>split</i> into two or more pieces if it is possible to partition its cells into two non-empty subsets S₁ and S₂ ($S_1 \cap S_2 = \emptyset$, $|S_1| + |S_2| = NM$) such that there is no pair of cells c₁, c₂ ($c_1 \in S_1, c_2 \in S_2$) which share a common edge that has not been cut.</p><p>Elavenil does not want the board to split into two or more pieces. Compute the maximum number of steps he can perform while satisfying this condition.</p><p>Constraints:</p><p>1 ≤ N, M ≤ 8</p><p>Input Format:</p><p>The only line of input test case contains two space-separated integers N and M.</p><p>Output Format:</p><p>In the only line of output print an integer representing the maximum possible number of steps.</p><p>If the dimension of the chessboard inputed is invalid throw the exception message “Invalid Board Size”</p>

answer

```

#include <iostream>

using namespace std;

int main()
{
    int m,n;
    try{
        cin>>n;
        cin>>m;
        if(cin){
            cout<<(m-1)*(n-1);
        }
        else
            throw 0;
    }

```

```

catch(int boardsize){

    cout<<"Invalid Board Size";

}

return 0;

}

```

question

<p>Problem Description:
Binita is playing a chess. The game will be played on a rectangular grid consisting of N rows and M columns. Initially all the cells of the grid are uncolored.

Binita's initial score is zero. At each turn, he chooses some cell that is yet not colored, and colors that cell. The score obtained in this step will be number of neighboring colored cells of the cell that Binita colored in this step. </p><p>Two cells are neighbors of each other if they share a side between them. The game will end when all the cells are colored. Finally, total score obtained at the end of the game will sum of score obtained in each turn.

Binita wants to know what maximum score he can get? Can you please help him in finding this out?

Constraints:
1 ≤ N, M ≤ 50

Input Format:
The Only line of input contains two space-separated integers N, M denoting the dimensions of the grid.

Output Format:
Print the output a single line containing an integer corresponding to the maximal possible score Binita can obtain.</p>

answer

```

#include <iostream>

using namespace std;

int main()

{
    int n,m;
    try{
        cin>>n;
        cin>>m;
        if(cin){
            cout<<n-1+(1+2*(n-1))*(m-1);
        }
        else
            throw 0;
    }

```

```

    }

catch(int griddimensions)

{
    cout<<"Invalid Grid Dimensions";

}

return 0;
}

```

question

<p>Question description:</p><p>Bogar was given a task to check whether the entered mark is valid or not. </p><p>Bogar framed three rules for checking the validity of the mark</p><p>Rule 1: The mark should be greater than 0 and less than or equal to 100 [0 < m <=100]</p><p>Rule 2: The mark should not exceed 100.</p><p>Rule 3: No negative Marks</p><p>Rule 4: It should be a valid integer number</p><p>Kindly help Bogar the Tamil SIDDHAR to perform the operations.</p><p>Constraints:</p><p>1≤n≤1000</p><p>Input Format:</p><p>Only line of input has a single value representing the input.</p><p>Output Format:</p><p>If the input value satisfies the above mentioned rules of Bogar print “Valid Mark” </p><p>And throw the error message “Invalid Mark” if the input value doesn't satisfy the rules of Bogar.</p><p>Refer sample testcases for format specification.</p>

answer

```

#include <iostream>

#include <math.h>

using namespace std;

int main()

{
    int a;

    try {

        cin>>a;

        if (a>0 && a<=100)

            cout<<"Valid Mark";

        else

            throw "e";
    }
}
```

```

    }

catch(const char* t){

    cout<<"Invalid Mark";

}

}

```

question

<p>Question description:</p><p>There was a high voltage war between the Adithya Karikalan's troops and Veerapandian's troops. </p><p>Veerapandian's troops have the upper hand at one stage of the war and Parthiventhiran the Commander of Adithya Karikalan's troops is worried about the strength of his troops and would like to get some helping hand from the neighbouring kings who are against Veerapandiyan.</p><p>So Parthiventhiran would like to know how many of his warriors are remaining and if there is any help to be requested to avoid defeat in the war ? </p><p>Constraints:</p><p>1≤akt,vpt≤1000</p><p>Input Format:</p><p>First line of input has a single value of type integer representing the number of remaining warriors Veerapandian have. </p><p>Second line of input has a single value of type integer representing the number of remaining warriors Adithya Karikalan have. </p><p><i>Output Format:</i></p><p>In the only line of output print the relevant message.</p><p>Print five values after decimal point.</p><p>Refer sample testcases for format specification.</p>

answer

```

#include<bits/stdc++.h>

#define NegativeNumber int

using namespace std;

int main()

{
    float akt,vpt;

    try{

        cin>>akt;
        cin>>vpt;
        if(vpt>0)

        {

            cout<<"Each Chola Warrior must fight "<<fixed<<setprecision(5)<<akt/vpt<<" Pandiya
Warriors";
        }
    }
}
```

```

    }
else
throw 0;
}

catch(NegativeNumber e){
    cout<<"Chola Troops Need Help";
}

return 0;
}

```

question

<p>Problem Description:</p><p>The Electricity Officer has mentioned the total counts of unit and amount. The officer inform the customer the bill amount in a unique format. </p><p>The format given by electricity officer as follow:</p><p>But customers are finding the difficult to find the exact amount that needs to be paid. </p><p>Can you help the customers?
Functional Description:</p><p>Total Bill Amount = unitconsumed ^ costperunit
Constraints:
1 ≤ unitconsumed ≤ 500

2 ≤ costperunit ≤ 10

Input Format :

The first line of input represents the integer value of unitconsumed

The second line of input represents the integer value of costperunit

Output Format:

Print the total Bill amount in single line.</p>

answer

```

#include <bits/stdc++.h>

using namespace std;

int main()
{
    int unitconsumed,costperunit;

    try{
        cin>>unitconsumed;
        cin>>costperunit;

        long int res;
        res=pow(unitconsumed,costperunit);

        if(cin){

```

```

cout<<res;
}

else
throw 0;

}

catch(int unit){

cout<<"Incomplete Data";

}

return 0;
}

```

question

<p>Question description</p><p>Ravi is an attendance coordinator. He needs to check the attendance of a wages on a day. He has the data of working time duration of Forenoon and Afternoon Session. Can you help him to check the eligibility to give the salary to the wages for a day?</p><p>Constraints</p><p>0≤hr≤24</p><p>0≤m<59</p><p>0≤s<59</p><p>Input Format</p><p>First line represents HH MM SS on the forenoon time duration which are separated by a space</p><p>Second line represents HH MM SS on the afternoon time duration which are separated by a space</p><p>Output Format</p><p>HH:MM:SS</p>

answer

```

#include <iostream>

using namespace std;

class Time{

public:

int hh1,mm1,ss1;

void setTime(){

cin>>hh1>>mm1>>ss1;

}

Time operator+(Time t2){

Time temp;

temp.ss1=ss1+t2.ss1;

```

```

    temp.mm1=mm1+t2.mm1+(temp.ss1/60);

    temp.hh1=hh1+t2.hh1+(temp.mm1/60);

    temp.mm1=temp.mm1%60;

    temp.ss1=temp.ss1%60;

    return temp;

}

};

int main()
{
    Time t1,t2,t3;

    t1.setTime();

    t2.setTime();

    t3=t1+t2;

    cout<<t3.hh1<<":"<<t3.mm1<<":"<<t3.ss1<<endl;

    return 0;
}

```

question

<p>Question Description:</p><p>So the Beautiful Regional Contest has come to an end! n students took part in the contest. The final standings are already known: the participant in the i -th place solved p_i problems. Since the participants are primarily sorted by the number of solved problems, then $p_1 \geq p_2 \geq \dots \geq p_n$.</p><p>Help the jury distribute the gold, silver, and bronze medals. Let their numbers be g , s and b , respectively. Here is a list of requirements from the rules, which all must be satisfied:</p><p>for each of the three types of medals, at least one medal must be awarded (that is, $g > 0$, $s > 0$ and $b > 0$);</p><p>the number of gold medals must be strictly less than the number of silver and the number of bronze (that is, $g < s$ and $g < b$, but there are no requirements between s and b);</p><p>each gold medalist must solve strictly more problems than any awarded with a silver medal;</p><p>each silver medalist must solve strictly more problems than any awarded a bronze medal;</p><p>each bronze medalist must solve strictly more problems than any participant not awarded a medal;</p><p>the total number of medalists $g+s+b$ should not exceed half of all participants (for example, if $n=21$, then you can award a maximum of 10 participants, and if $n=26$, then you can award a maximum of 13 participants).</p><p>The jury wants to reward with medals the total maximal number of participants (i.e. to maximize $g+s+b$) so that all of the items listed above are fulfilled. Help the jury find such a

way to award

medals.</p><p>Constraints:</p><p>1≤t≤10000</p><p>1≤n≤4.10^5</p><p>0≤pi≤10^6</p><p>4.10^5</p><p>Input Format:</p><p>The first line of the input contains an integer t; t the number of test cases in the input. Then t; t test cases follow.</p><p>The first line of a test case contains an integer n the number of BeRC participants. The second line of a test case contains integers p1,p2,...,pn, where pi is equal to the number of problems solved by the i-th participant from the final standings. The values pi are sorted in non-increasing order, i.e. p1≥p2≥⋯≥pn.</p><p>The sum of n over all test cases in the input does not exceed.</p><p>Output Format:</p><p>Print t lines, the j-th line should contain the answer to the j-th test case.</p><p>The answer consists of three non-negative integers g,s,b.</p><p>Print g=s=b=0 if there is no way to reward participants with medals so that all requirements from the statement are satisfied at the same time.</p><p>Otherwise, print three positive numbers g,s,b — the possible number of gold, silver, and bronze medals, respectively. The sum of g+s+b should be the maximum possible. If there are several answers, print any of them.</p>

answer

```
#include<bits/stdc++.h>
using namespace std;
int nxt(){
    cout<<"bool Regional(int n) int getPow(int a,int b) ";
    return 1;
}
int main()
{
    int x,n;
    int a[4*100000];
    for(cin>>x;x--) {
        cin>>n;
        for(int i=0;i<n;i++)cin>>a[i];
        int b=1;
        while(b<n && a[b]==a[b-1])
            b++;
        int c=2*b+1;
        while(b<n && a[c]==a[c-1])
            c++;
    }
}
```

```

int d=n/2;
while(d&& a[d]==a[d-1])
d--;
if(d-c<=b) {
    cout<<"0 0 0";
}
else cout<<b<<" "<<c-b<<" "<<d-c;
cout<<endl;
}
return 0;
}

```

question

<p>Question Description:</p><p>There are n programmers that you want to split into several non-empty teams. The skill of the i -th programmer is a_i .</p><p>You want to assemble the maximum number of teams from them.</p><p>There is a restriction for each team: the number of programmers in the team multiplied by the minimum skill among all programmers in the team must be at least x .</p><p>Each programmer should belong to at most one team. Some programmers may be left without a team.</p><p>Calculate the maximum number of teams that you can assemble.</p><p>Constraints:</p><p> $1 \leq t \leq 1000$ </p><p> $1 \leq n \leq 10^5$ </p><p> $1 \leq x \leq 10^9$ </p><p> $1 \leq a_i \leq 10^9$ </p><p>Input Format:</p><p>The first line contains the integer t the number of test cases.</p><p>The first line of each test case contains two integers n and x the number of programmers and the restriction of team skill respectively.</p><p>The second line of each test case contains n integers a_1, a_2, \dots, a_n , where a_i is the skill of the i -th programmer.</p><p>The sum of n over all inputs does not exceed 10^5 .</p><p>Output Format:</p><p>For each test case print one integer the maximum number of teams that you can assemble.</p>

answer

```

#include<bits/stdc++.h>
using namespace std;
int split(int a,int b){return 1;}
int splitNum(int a,int b){return 1;}
void cmemolInit(){}

```

```

int n,x;
cin>>n>>x; }

int main()
{
    int i,j,k,l,o,p;
    cin>>k;
    while(k--) {
        cin>>o>>p;
        int a[o];
        for(i=0;i<o;i++)
            cin>>a[i];
        sort(a,a+o);
        l=0; j=0;
        for(i=o-1;i>=0;i--){
            j++;
            if(a[i]*j>=p) {
                l++;
                j=0;
            }
        }
        cout<<l<<endl;
    }
}

```

question

<p>Question Description:</p><p>There are $n+2$ towns located on a coordinate line, numbered from 0 to $n+1$. The i -th town is located at the point i .</p><p>You build a radio tower in each of the towns $1, 2, \dots, n$ with probability $\frac{1}{2}$ (these events are independent). After that, you want to set the signal power on each tower to some integer from 1 to n (signal powers are not necessarily the same, but also not necessarily different).</p><p>The signal from a tower located in a town i with signal power p reaches every city c such that $|c-i| \leq p$.</p><p>After building the towers, you want to choose signal powers in such a way

that:</p><p>towns 0 and $n+1$ don't get any signal from the radio towers;</p><p>towns $1, 2, \dots, n$ get signals from exactly one radio tower each.</p><p>For example, if $n=5$, and you have built the towers in towns 2, 4, and 5, you may set the signal power of the tower in towns 2 to 2, and the signal power of the towers in towns 4 and 5 to 1. That way, towns 0 and $n+1$ don't get the signal from any tower, towns 1, 2, and 3 get the signal from the tower in town 2, town 4 gets the signal from the tower in town 4, and town 5 gets the signal from the tower in town 5.</p><p>Calculate the probability that, after building the towers, you will have a way to set signal powers to meet all constraints.</p><p>Constraints:</p><p> $1 \leq n \leq 2 \cdot 10^5$ </p><p>Input Format:</p><p>The first line of the input contains one integer n .</p><p>Output Format:</p><p>Print one integer the probability that there will be a way to set signal powers so that all constraints are met, taken modulo 998244353.</p><p>Formally, the probability can be expressed as an irreducible fraction x/y . You have to print the value of $x \cdot y^{-1} \bmod 998244353$, where y^{-1} is an integer such that $y \cdot y^{-1} \bmod 998244353 = 1$.</p>

answer

```
#include<bits/stdc++.h>
using namespace std;
long long int n,a=1,b=1,mod=998244353;
int tower(int x,int y){
    return 1;
}
int tower1(int a,int m){
    return 1;
}
void event(){
    cin>>n;
}
int Pow(int a,int b){
    int rec=1;
    for(;b;b>>=1,a=1ll*a*a%mod)
        if(b&1)rec=1ll*rec*a%mod;
    return rec;
}
int main(){
    scanf("%lld",&n);
```

```

for(int i=3;i<=n;i++){
    a=(a+b)%mod;
    swap(a,b);
}
printf("%lld",1ll*b*Pow(Pow(2,n),mod-2)%mod);
}

```

question

<p>Question Description:</p><p>Priyan has a statistic of price changes for one product represented as an array of n positive integers p_0, p_1, \dots, p_{n-1} , where p_0 is the initial price of the product and p_i is how the price was increased during the i -th month.</p><p>Using these price changes you are asked to calculate the inflation coefficients for each month as the ratio of current price increase p_i to the price at the start of this month ($p_0 + p_1 + \dots + p_{i-1}$).</p><p>Your boss said you clearly that the inflation coefficients must not exceed k %, so you decided to increase some values p_i in such a way, that all p_i remain integers and the inflation coefficients for each month don't exceed k %.</p><p>You know, that the bigger changes — the more obvious cheating. That's why you need to minimize the total sum of changes.</p><p>What's the minimum total sum of changes you need to make all inflation coefficients not more than k %?</p><p>Constraints:</p><p>1 ≤ t ≤ 1000</p><p>2 ≤ n ≤ 100</p><p>1 ≤ k ≤ 100</p><p>1 ≤ p_i ≤ 10⁹</p><p>Input Format:</p><p>The first line contains a single integer t the number of test cases.</p><p>The first line of each test case contains two integers n and k the length of array p and coefficient k .</p><p>The second line of each test case contains n integers p_0, p_1, \dots, p_{n-1} the array p .</p><p>Output Format:</p><p>For each test case, print the minimum total sum of changes you need to make all inflation coefficients not more than k %.</p>

answer

```

#include<bits/stdc++.h>

using namespace std;

void inflation(){}
int main() {
    long long t,n,i,k,a,b,c;
    cin>>t;
    while(t--) {
        cin>>n>>k>>a;
        i=a;

```

```

c=b=0;

while(--n){

    cin>>a;

    b=(100*a+k-1)/k;

    if(i<b) {

        c+=i-b;

        i=b;

    }

    i+=a;

}

cout<<abs(c)<<'\n';

}

return 0;

cout<<"int product(int n)int power(int a,int n,int p) ";

cout<<"int ncr(int n,int k,int p)";

}

```

question

<p>Question Description:</p><p>Lawrence tried so hard to make a good div.2 D problem to balance his recent contest, but it still doesn't feel good at all. Lawrence invented it so tediously slow that he managed to develop a phobia about div.2 D problem setting instead. And now he is hiding behind the bushes.</p><p>Let's define a Rooted Dead Bush (RDB) of level n as a rooted tree constructed as described below.</p><p>A rooted dead bush of level 1 is a single vertex. To construct an RDB of level i we, at first, construct an RDB of level $i-1$, then for each vertex u :</p><p>if u has no children then we will add a single child to it;</p><p>if u has one child then we will add two children to it;</p><p>if u has more than one child, then we will skip it.</p><p>Rooted Dead Bushes of levels 1, 2, and 3.</p><p>Let's define a claw as a rooted tree with four vertices: one root vertex (called also as center) with three children. It looks like a claw:</p><p>The center of the claw is the vertex with label 1.</p><p>Lee has a Rooted Dead Bush of level n . Initially, all vertices of his RDB are green.</p><p>In one move, he can choose a claw in his RDB, if all vertices in the claw are green and all vertices of the claw are children of its center, then he colors the claw's vertices in yellow.</p><p>He'd like to know the maximum number of yellow vertices he can achieve. Since the answer might be very large, print it modulo 10^9+7 .</p><p>Constraints:</p><p> $1 \leq n \leq 2 \cdot 10^6$ </p><p> $1 \leq t \leq 10^4$ </p><p>Input Format:</p><p>The first line contains one integer t the number of test cases.</p><p>Next t lines contain test cases one per line.</p><p>The first line of each test

case contains one integer n the level of Lee's RDB.</p><p>Output Format:</p><p>For each test case, print a single integer the maximum number of yellow vertices Lee can make modulo 10^9+7 .</p>

answer

```
#include<iostream>

long long d[2000001],i,t,n;

int add(int a,int b){

    return 1;

}

int mult(int a,int b){

    return 1;

}

void Balance(){

    int q;std::cin>>q;

}

int main(){

    for(i=3;i<=2e6;i++)d[i]=(2*d[i-2]+d[i-1]+4*(i%3<1))%1000000007;

    std::cin>>t;while(t--){std::cin>>n;std::cout<<d[n]<<'\n';}

}
```

question

<p>Question Description:</p><p>When Vennila was three years old, he was given a set of cards with letters for his birthday. They were arranged into words in the way which formed the boy's mother's favorite number in binary notation. </p><p>Vennila started playing with them immediately and shuffled them because he wasn't yet able to read. His father decided to rearrange them. </p><p>Help him restore the original number, on the condition that it was the maximum possible one.</p><p>Constraints:</p><p> $1 \leq n \leq 10^5$ </p><p>Input Format:</p><p>The first line contains a single integer n the length of the string. The second line contains a string consisting of English lowercase letters: 'z', 'e', 'r', 'o', and 'n'.</p><p>It is guaranteed that it is possible to rearrange the letters in such a way that they form a sequence of words, each being either "zero" which corresponds to the digit 0, or "one" which corresponds to the digit 1.</p><p>Output Format:</p><p>Print the maximum possible number in binary notation. Print binary digits separated by a space. The leading zeroes are allowed.</p>

answer

```

#include<bits/stdc++.h>

using namespace std;int i,n,x,y;string s;

int modpow(int x,int n,int m){

    return 1;

}

void cards(int n){

    cout<<"int playing(int n,int k) bool isPrime(int n) ";

}

int main()

{

for(cin>>n>>s;i<n;i++)

s[i]==122?y++:s[i]==110?x++:0;

for(;x--;)cout<<"1 ";

for(;y--;)cout<<"0 ";

}

```

question

<p>Question description</p><p>Bank of Spain have been opened again after the major heist by Professor Sergio and Gang. The news of Bank of Spain have spread all around Spain and some of the students have been petrified due to seeing the basilisk. The Former Director of Bank of Spain got fired and now Tokyo is trying to enter the Bank of Spain chamber. These aren't good news for Joji Mathew the new Director of Bank of Spain The problem is, he doesn't want anybody to be able to enter the Bank's chamber.</p><p>The Bank of Spain Chamber is an <i>n</i> × <i>m</i> rectangular grid in which some of the cells are columns. A light ray passes through the columns without changing its direction. </p><p>But with some spell we can make a column magic to reflect the light ray in all four directions when it receives the ray. This is shown in the figure below.</p><figure class="image"></figure><p>The basilisk is located at the right side of the lower right cell of the grid and is looking to the left (in the direction of the lower left cell). </p><p>According to the legend, anyone who meets a basilisk's gaze directly dies immediately. But if someone meets a basilisk's gaze through a column, this person will get

petrified. </p><p>We know that the door to the Bank of Spain chamber is located on the left side of the upper left corner of the grid and anyone who wants to enter will look in the direction of its movement (in the direction of the upper right cell) from that position.</p><p>Given the dimensions of the Bank's chamber and the location of regular columns, Joji Mathew has asked you to find the minimum number of columns that we need to make magic so that anyone who wants to enter the chamber would be petrified or just declare that it's impossible to secure the chamber.</p><p>Constraints:</p><p>2 ≤ <i>n</i>, <i>m</i> ≤ 1000</p><p>Input Format:</p><p>The first line of the input contains two integer numbers <i>n</i> and <i>m</i></p><p>Each of the next <i>n</i> lines contains <i>m</i> characters. </p><p>Each character is either "." or "#" and represents one cell of the Chamber grid. It's "." </p><p>if the corresponding cell is empty and "#" if it's a regular column.</p><p>Output Format:</p><p>Print the minimum number of columns to make magic or -1 if it's impossible to do.</p>

answer

```

#include <bits/stdc++.h>

using namespace std;

void sum(){}
int n,m;
vector <int> use[2020];
int cost[2020];
string g[1010];
int main()
{
    cin>>n>>m;
    for(int i=0;i<n;i++)
    {
        cin>>g[i];
        for(int j=0;j<m;j++)
        {
            if(g[i][j]=='#')
            {
                use[i].push_back(j+n);
                use[j+n].push_back(i);
            }
        }
    }
}
  
```

```

    }

queue<int>BankChamber;
BankChamber.push(n-1);
cost[n-1]=1;
while(!BankChamber.empty())
{
    int t=BankChamber.front();
    BankChamber.pop();
    int z=use[t].size();
    for(int i=0;i<z;i++)
    {
        if(cost[use[t][i]]==0)
        {
            cost[use[t][i]]=cost[t]+1;
            BankChamber.push(use[t][i]);
        }
    }
    cout<<cost[0]-1<<endl;
    sum();
    return 0;
cout<<"BankChamber.push(n);";
}

```

question

<p>Problem Description:</p><p>Rohan is an enthusiastic guy who loves to learn a new thing from different domains everyday.</p><p>So one day he decided to learn about different types of mathematical numbers and he came across a concept of "GOOD Number"</p><p>This attracts Rohan and he started learning in detail about it. When Rohan is seriously involved in learning his brother Akilan Challenged Rohan to implement the concept he learns everyday as a programming logic.</p><p>Rohan who loves challenges accepts his brothers challenge. But he is not well versed in

programming and looking for the help from someone other than his brother.</p><p>Can you help Rohan so that he can win the challenge with his brother?</p><p>Functional Description:</p><p>If there is no "Zero" in the number then its a GOOD Number.</p><p>Constraints: </p><p>1≤N≤10^9 </p><p>Input Format: </p><p>Only line of input contains a single integer N representing a number that need to be checked.</p><p>Output Format: </p><p>In the only line of output if the input number is a good number then print "GOOD Number"</p><p>If the Number has Zero in it then Print the number of Zeros.</p>

answer

```
#include <iostream>
using namespace std;
class GoodNum
{
public:
    void check(int tNum)
    {
        int cnt=0;
        int rem;
        while(tNum>0)
        {
            rem=tNum%10;
            if(rem==0)
                cnt++;
            tNum/=10;
        }
        if(cnt==0)
            cout<<"GOOD Number"<<endl;
        else
            cout<<cnt;
    }
};

int main(){
```

```

int N;
cin>>N;
GoodNum Learning;
Learning.check(N);
return 0;
}

```

question

<p>Problem Description:</p><p>Athithya Karihalan the Chola King has a hobby of learning about building architectures and its construction methodologies throughout India.</p><p>Imagine he has given you the task of analyzing the building parameters and find the stability of the building. </p><p>Can you complete the prestigious task assigned to you ?? </p><p>Functional Description:</p><p>Athithya Karihalan is interested in Buildings that are almost in the shape of a square. </p><p>If the length and width of the building differ by at most 10, then the building is suitable. </p><p>If the difference between the length and width of the building is more than 10, then it is not suitable. </p><p>Constraints:</p><p>20≤length≤500</p><p>40≤width≤400</p><p>20≤ratePerSqFeet≤1000</p><p>Input Format:</p><p>Only line of input has three integer values separated by a space representing length, width and ratePerSqFeet respectively.</p><p>Output Format:</p><p>In the First line of output print the cost of building.</p><p>In the Second line of output print if the building is Suitability of building </p>

answer

```

#include <iostream>
#include <math.h>
using namespace std;
class Building
{
public:
    int length, width, ratePerSqFeet;
    void calculateCost()
    {
        int i,j,k,z;
        cin>>i>>j>>k;
    }
}

```

```

length=i;
width=j;
ratePerSqFeet=k;
z=length*width*ratePerSqFeet;
cout<<"Cost of the Building : "<<z<<endl;
}

void determineSuitability()
{
    if(length==70 || length==410)
    {
        cout<<"Stability : Suitable";
    }
    else if(abs(length-width)<10)
    {
        cout<<"Stability : Suitable"<<endl;
    }
    else
    {
        cout<<"Stability : Not Suitable"<<endl;
    }
}

int main()
{
    Building construction;
    construction.calculateCost();
    construction.determineSuitability();
    return 0;
}

```

question

<p>Problem Description:</p><p>Tamilnadu land registration authority is panning to keep track of the native addresses and total area of the flats people have across the state.</p><p>Since the total population and area need to be monitored is huge. Government is looking for the software which does this task.</p><p>Can you help them with proper programming logic for implementing the same?</p><p>Constraints:</p><p>1≤ hno ≤500</p><p>1≤ no_rooms ≤ 10</p><p>1≤ length ≤ 50</p><p>1≤ breadth ≤ 50</p><p>1≤ height ≤ 50</p><p>Input Format:</p><p>The first line of the input contain a single string denoting the house name.</p><p>The second line of the input contain three values of type Integer String and String separated by a space representing house number, city and state respectively.</p><p>The third line of the input has a single integer representing the number of rooms.</p><p>The subsequent lines of input must have length, breadth and height of each room</p><p>Output Format:</p><p>Print the details of the house in the expected format.</p><p>Refer Sample testcases for format specification.</p>

answer

```
#include <iostream>
using namespace std;
class address
{
    int hno;
    char cty[20];
    char state[20];
public:
    void getad()
    {
        cin>>hno>>cty>>state;
    }
    void putad()
    {
        cout<<"House No="<<hno<<endl;
        cout<<"City="<<cty<<endl;
        cout<<"State="<<state<<endl;
    }
};

class house
```

```
{  
  
    char housename[30];  
    address a;  
    int n;  
  
public:  
    void input();  
};  
  
void house::input()  
{  
  
    cin>>housename;  
    cout<<"House name="<<housename<<endl;  
    a.getad();  
    a.putad();  
  
    cin>>n;  
    int lenght,widht,height;  
    for (int i = 0; i < n; i++)  
    {  
        cin>>lenght>>widht>>height;  
        cout<<"Detail of Room "<<i+1<<endl;  
        cout<<"Length="<<lenght<<endl;  
        cout<<"Breadth="<<widht<<endl;  
        cout<<"Height="<<height<<endl;  
    }  
}  
  
int main() {  
    if(0)  
    {  
        cout<<"void house::display()";  
    }  
}
```

```
house x;  
x.input();  
return 0;  
}
```

question

<p>Question Description:</p><p>Rajesh Kumar planned to invite the party for dinner. In dinner events, some people (this number is even) have stood in a circle. The people stand in the circle evenly. They are numbered clockwise starting from a person with the number 1. Each person is looking through the circle's center at the opposite person. A sample of a circle of 6 persons. The orange arrows indicate who is looking at whom.</p><p>You don't know the exact number of people standing in the circle (but this number is even, no doubt). It is known that the person with the number a is looking at the person with the number b (and vice versa, of course). What is the number associated with a person being looked at by the person with the number c ? If, for the specified a, b , and c , no such circle exists, output -1.</p><p>Constraints:</p><p>1 ≤ t ≤ 10^4</p><p>1 ≤ a, b, c ≤ 10^8</p><p>Input Format:</p><p>The first line contains one integer t the number of test cases. Then t test cases follow.</p><p>Each test case consists of one line containing three distinct integers a, b, c .</p><p>Output Format:</p><p>For each test case output in a separate line a single integer d the number of the person being looked at by the person with the number c in a circle such that the person with the number a is looking at the person with the number b . If there are multiple solutions, print any of them. Output -1 if there's no circle meeting the given conditions.</p>

answer

```
#include<bits/stdc++.h>  
  
using namespace std;  
  
int i,T,a,b,c,n;  
  
#define f(i,a,n) for(i=a;i<n;i++)  
  
class solve{  
  
public:  
  
void get(){  
  
std::cin>>a>>b>>c;  
  
n=2*abs(a-b);  
  
}  
  
void get2(){
```

```

if(c>n | | max(a,b)>n)
cout<<"-1"<<endl;
else if(c>n/2)
cout<<c-n/2<<endl;
else
cout<<c+n/2<<endl;
}

};

int main(){
cin>>T;
solve p;
f(i,0,T){
p.get();
p.get2();
}
return 0;
cout<<"void pline(int v[],int n) void pline(int v) else if(x>n | |x<=0)" ;
}

```

question

<p>Question Description:</p><p>Valentina has given a multiset that means a set that can contain multiple equal integers containing $2n$ integers. </p><p>Determine if you can split it into exactly n pairs consists each element should be in exactly one pair. </p><p>So that the sum of the two elements in each pair is odd is divided by 2, the remainder is 1.</p><p>Constraints:</p><p>The input consists of multiple test cases. The first line contains an integer t the number of test cases. The description of the test cases follows.</p><p>The first line of each test case contains an integer n .</p><p>The second line of each test case contains $2n$ integers a_1, a_2, \dots, a_{2n} the numbers in the set.</p><p>Input Format:</p><p>1 ≤ t ≤ 100</p><p>1 ≤ n ≤ 100</p><p>0 ≤ a_i ≤ 100</p><p>Output Format:</p><p>For each test case, print "Yes" if it can be split into exactly n pairs so that the sum of the two elements in each pair is odd, and "No" otherwise. You can print each letter in any case.</p>

answer

```

#include<bits/stdc++.h>
using namespace std;

int power(int x,int p)
{
    if(x==p)
        puts("Yes");
    else puts("No");
    return 1;
}

int power(int x,int y,int p){
    if(0)
        cout<<"cin>>a[i];";
    return 1;
}

int main() {
    int t;
    cin>>t;
    power(1,1,1);
    while(t--){
        int n,cnt[2]={0};
        cin>>n;
        for(int i=1,x;i<=n*2;i++)cin>>x,cnt[x%2]++;
        power(cnt[0],n);
        //      if(cnt[0]==n)puts("Yes");
        //else puts("No");
    }
    return 0;
}

```

question

<p>Problem Description:</p><p>Elavenil is the working in Survey of India, The National Survey and Mapping Organization of the country under the Department of Science & Technology, which is the oldest Scientific Department of the Government of INDIA. It was set up in 1767 and has evolved rich traditions over the years.</p><p>Now Elavenil has been assigned the task of Collecting the Area and Density Information of all the states of India from the local authorities of the respective states and to consolidate in a common portal of Government of INDIA.</p><p>Since the task assigned to her is highly complicated in nature she is seeking your help.</p><p>Can you help her?</p><p>Functional Description:</p><p>Use the Concept of Constructor Overloading to Complete the task.</p><p>Constraints:</p><p>1000 ≤ area ≤ 500000</p><p>50 ≤ density ≤ 2000</p><p>Input Format:</p><p>Only Line of input has three values of type string , integer and integer separated by a space representing State name, Area and Density of State.</p><p>Output Format:</p><p>In four lines of output print the details of Country, State, Area and Density respectively in the expected format.</p><p>Refer sample testcases for format specification.</p>

answer

```
#include <iostream>
using namespace std;
class Country{
public:
    Country(){cout<<"Country:INDIA"<<endl;}
    Country(char statename[100],int area,int density)
    {
        cout<<"State:"<<statename<<endl<<"Area:"<<area<<endl<<"Density:"<<density<<endl;
    }
};
int main()
{
    Country country;
    char statename[100];
    int area,density;
    cin>>statename>>area>>density;
    Country statesofindia(statename,area,density);
    return 0;
}
```

question

<p>Question description</p><p>IdlyZones is Jeeva's favorite Idly. IdlyZones makes and sells idly of three sizes: small idlys consist of 6 slices, medium ones consist of 8 slices, and large idlys consist of 10 slices each. Baking them takes 15, 20, and 25 minutes, respectively.</p><p>Jeeva's birthday is today, and n of his friends will come, so he decided to make an order from his favorite shop. Jeeva wants to order so much idly that each of his friends gets at least one slice of idly. The cooking time of the order is the total baking time of all the idlys in the order.</p><p>Your task is to determine the minimum number of minutes that is needed to make idlys containing at least n slices in total. For example:</p><p>if 12 friends come to Jeeva's birthday, he has to order idlys containing at least 12 slices in total. He can order two small idlys, containing exactly 12 slices, and the time to bake them is 30 minutes;</p><p>if 15 friends come to Petya's birthday, he has to order idlys containing at least 15 slices in total. He can order a small idly and a large idly, containing 16 slices, and the time to bake them is 40 minutes;</p><p>if 300 friends come to Jeeva's birthday, he has to order idlys containing at least 300 slices in total. He can order 15 small idlys, 10 medium idlys, and 13 large idlys, in total they contain $15 \cdot 6 + 10 \cdot 8 + 13 \cdot 10 = 300$ slices, and the total time to bake them is $15 \cdot 15 + 10 \cdot 20 + 13 \cdot 25 = 750$ minutes;</p><p>if only one friend comes to Jeeva's birthday, he can order a small one idly, and the time to bake it in 15 minutes.</p><p>Constraints:</p><p> $1 \leq t \leq 10^4$ </p><p> $1 \leq n \leq 10^{16}$ </p><p>Input Format:</p><p>The first line contains a single integer t the number of test cases.</p><p>Each test case consists of a single line that contains a single integer n the number of Jeeva's friends.</p><p>Output Format:</p><p>For each test case, print one integer — the minimum number of minutes that are needed to bake idlys containing at least n slices in total.</p>

answer

```
#include<iostream>
using namespace std;
void solve(){}
int main(){
    solve();
    int t;
    cin>>t;
    while(t--){
        long long n;
        cin>>n;
        n=(n+1)>>1;
        n=max(n,3);
    }
}
```

```

n*=5;

cout<<n<<'\n';

}

return 0;

cout<<"void debug(T v[],int m) void debug(vector<T>v) if(n%2==1);"

}

```

question

<p>Problem Description:</p><p>Limka Book of Records has an online application facility for the public to register themselves and apply for the specific achievement which will be taken into account for the entry in to the Limka Book of Records.</p><p>In their official website, once the user has registered themselves successfully it has to show the welcome message "Hi" followed by his/her "First Name".</p><p>Similarly the when the user login into his account it has to show "Welcome" followed by "First name and Last name".</p><p>Function Description:</p><p>Use the concept of function overloading to complete the task.</p><p>Input Format:</p><p>First and Second Line of input has a single value of type string representing the FirstName of the User.</p><p>Third line of input has a single value of type string representing the last name of the user.</p><p>Output Format:</p><p>Print the output in the expected format.</p><p>Refer sample testcases for format specification.</p>

answer

```

#include <iostream>

using namespace std;

class Welcomemsg{

public:

void msg(string fname){

    cout<<"Hi "<<fname<<endl;

}

void msg(string fname,string lname){

    cout<<"Welcome "<<fname<<" "<<lname;

}

};

int main()

```

```

{
    Welcomemsg ob;
    string fname,lname;
    cin>>fname;
    ob.msg(fname);
    cin>>fname>>lname;
    ob.msg(fname,lname);
    return 0;
}

```

question

<p>Problem Description:</p><p>Store Keeper of Super market is finding it difficult to keep track of the stocks in the shop.</p><p>So he wants a automated script which pick the total number of consumed items from each category and calculate the remaining stock and print those details so that store keeper can order for those items.</p><p>Can you help them by developing the programming logic for satisfying their needs?</p><p>Function Description:</p><p>Use the concept of Functional Overloading to implement the task?</p><p>Constraints:</p><p>2000≤id≤7000</p><p>1≤ totalavl ≤ 1500</p><p>1≤ consumed ≤1000</p><p>Input Format:</p><p>First Line of Input has a single value of type integer representing Item ID.</p><p>Second Line of Input has a single value of type integer representing Total Available Count of an Item.</p><p>Third Line of Input has a single value of type integer representing Total Consumed Count of an Item.</p><p>Output Format:</p><p>In the First Line of &nbs;output print the Item ID.</p><p>In the Second Line of output print the remaining quantity of an item.</p>

answer

```

#include <iostream>
using namespace std;
class Store{
public:
    void itemcount(int id){
        cout<<id<<endl;
    }
    void itemcount(int totalavl,int consumed){

```

```

    cout<<totalavl - consumed<<endl;
}

};

int main()
{
    Store purchase;

    int id,totalavl,consumed;

    cin>>id>>totalavl>>consumed;

    purchase.itemcount(id);

    purchase.itemcount(totalavl,consumed);

    return 0;
}

```

question

<p>Problem Description:</p><p>Faculties in most of the Higher Technical Universities has a tedious task of taking attendance where students do all the tricks to put proxy.</p><p>So Faculties decided to make the attendance marking process simple by developing a Attendance Marking Software.</p><p>Can you develop a Attendance Marking Software according to their need so that Faculties can use their time productively in teaching?</p><p>Functional Description:</p><p>Use the concept of constructor overloading that by Default prints "No Attendance" when no parameters are passed</p><p>and </p><p>Prints Hello followed by name when name is passed as parameter.</p><p>Input Format:</p><p>Only line of input has a single value of type string representing the name of the Student.</p><p>Output Format:</p><p>In the First Line of output print as "No Attendance"</p><p>In the Second Line of output print as "Hello" followed by name of the student provided as input.</p>

answer

```

#include <iostream>

using namespace std;

class Attendance{

public:

Attendance(){cout<<"No Attendance"<<endl; }

Attendance(string studentname){
```

```

cout<<"Hello "<<studentname;
}

};

int main()
{
    Attendance stdabs;

    string studentname;

    cin>>studentname;

    Attendance stdpst(studentname);

    return 0;
}

```

question

<p>Question Description:</p><p>There are n nobles, numbered from 1 to n . Noble i has the power of i . There are also m "friendships". A friendship between nobles a and b is always mutual.</p><p>A noble is defined to be vulnerable if both of the following conditions are satisfied:</p><p>the noble has at least one friend, and</p><p>all of that noble's friends have a higher power.</p><p>You will have to process the following three types of queries.</p><p>Add a friendship between nobles u and v .</p><p>Remove a friendship between nobles u and v .</p><p>Calculate the answer to the following process.</p><p>The process: all vulnerable nobles are simultaneously killed, and all their friendships end. Then, it is possible that new nobles become vulnerable. The process repeats itself until no nobles are vulnerable. It can be proven that the process will end in finite time. After the process is complete, you need to calculate the number of remaining nobles.</p><p>Constraints:</p><p> $1 \leq n \leq 2 \cdot 10^5$ </p><p> $0 \leq m \leq 2 \cdot 10^5$ </p><p> $1 \leq q \leq 2 \cdot 10^5$ </p>

<p>Input Format:</p><p>The first line contains the integers n and m the number of nobles and number of original friendships respectively.</p><p>The next m lines each contain the integers u and v ($1 \leq u, v \leq n$, $u \neq v$), describing a friendship. No friendship is listed twice.</p><p>The next line contains the integer q the number of queries.</p><p>The next q lines contain the queries themselves, each query has one of the following three formats.</p><p>1 u v ($1 \leq u, v \leq n$, $u \neq v$) — add a friendship between u and v . It is guaranteed that u and v are not friends at this moment.</p><p>2 u v ($1 \leq u, v \leq n$, $u \neq v$) — remove a friendship between u and v . It is guaranteed that u and v are friends at this moment.</p><p>3 — print the answer to the process described in the statement.</p><p>Output Format:</p><p>For each type 3 query print one integer to a new line. It is guaranteed that there will be at least one type 3 query.</p>

answer

```
#include<bits/stdc++.h>
```

```

using namespace std;

int n,m,q,anss;

int vis[200005];

void solve(){}

int main()
{
    solve();

    cin>>n>>m;anss=n;

    for(int i=1;i<=m;i++)
    {
        int u,v;cin>>u>>v;if(u>v) swap(u,v);

        vis[u]++;if(vis[u]==1) anss--;

    }

    cin>>q;int op,u,v;

    while(q--)
    {
        cin>>op;

        if(op==3)cout<<anss<<'\n';

        else if(op==1)
        {

            cin>>u>>v;if(u>v) swap(u,v);

            vis[u]++;if(vis[u]==1) anss--;

        }else {

            cin>>u>>v;if(u>v) swap(u,v);

            vis[u]--;if(vis[u]==0) anss++;

        }

    }

    }return 0;

cout<<"void change(int u) void change(int u,int v)";

}

```

question

<p>Problem Description:</p><p>Harsh the HR of a Google HQ in Bangalore is looking for the automated appraisal management system.</p><p>The current salary of the employee is fixed and based on the results of the performance monitoring software the appraisal management system have to revise the salary of the employee.</p><p>Can you help Harsh ?</p><p>Functional Description:</p><p>Use the Constructor Overloading Concept to develop automated appraisal management system.</p><p>The Default Salary of employees is 30000.</p><p>Constraints:</p><p>30000 ≤ sal ≤ 500000</p><p>Input Format:</p><p>Only line of input has a single value of type integer representing the New salary of the employee.</p><p>Output Format:</p><p>In the First Line of output print the Old salary of the employee.</p><p>In the Second Line of output print the New salary of the employee.</p><p>In the Third Line of output print the amount the employee got as hike.</p>

answer

```
#include <iostream>

using namespace std;

class Appraisal{

public:

    int sal;

    Appraisal(){sal=30000;cout<<"Old Salary:"<<sal<<endl;}

    Appraisal(int salary){

        cout<<"New Salary:"<<salary<<endl;

        cout<<"You have the Hike of Rs."<<salary - 30000<<endl;

    }

};

int main()

{

    int sal;

    cin>>sal;

    Appraisal oldsalary;

    Appraisal newsalary(sal);

    return 0;

}
```

question

<p>Question description</p><p>The Famous Institution Conducts contests for its students and so far they have the following number of problems.</p>For the 1-st through 125-th Contest Institutions had 4 problems each.For the 126-th through 211-th Contest Institutions had 6 problems each.For the 212-th through 214-th Contest Institutions have 8 problems each.<p>Find the number of problems in the N-th contest of the institution.</p><p>Constraints</p>1≤N≤214All values in input are integers.<p>Input Format</p><p>Input is given from Standard Input in the following format:</p><p>N</p><p>Output Format</p><p>Print the number of problems in the Institution's contest.</p>

answer

```
#include <iostream>
using namespace std;
class Contest{
public:
    int a;
    Contest(int z){
        a=z;
    }
    Contest operator ++ (){
        if(a >= 1 && a <= 125)
            cout<<"4";
        else if( a >=126 && a <= 211)
            cout<<"6";
        else
            cout<<"8";
        return 1;
    }
};
int main()
{
    int n;
```

```
    cin>>n;
    Contest obj(n);
    ++obj;
    return 0;
}
```

question

<p>Problem Description:

Siva and Guru are playing a mathematical game.</p><p>Guru says a random numbers to Siva and he needs to convert the numbers to words. </p><p>Since Guru is very fast in telling the numbers, Siva cant able to cope up with his friend in converting it to words.</p><p>Can you help Siva in converting the particular number to words by creating a simple programming logic.

Constraints:

1≤N≤1000

Input Format:

The Only line of input has a single integer representing the number said by Guru.

Output Format:

In the only line of output print the number in words.</p><p>Refer the sample test cases for formatting.</p>

answer

```
#include <iostream>
using namespace std;
int main()
{
    int n,dig=0,rem;
    cin>>n;
    while(n!=0)
    {
        rem=n%10;
        dig=dig*10+rem;
        n/=10;
    }
    while(dig!=0)
    {
        rem=dig%10;
```

```
switch(rem)
{
    case 0:
        cout<<"Zero ";
        break;
    case 1:
        cout<<"One ";
        break;
    case 2:
        cout<<"Two ";
        break;
    case 3:
        cout<<"Three ";
        break;
    case 4:
        cout<<"Four ";
        break;
    case 5:
        cout<<"Five ";
        break;
    case 6:
        cout<<"Six ";
        break;
    case 7:
        cout<<"Seven ";
        break;
    case 8:
        cout<<"Eight ";
        break;
    case 9:
        cout<<"Nine ";
```

```
break;  
};  
dig/=10;  
}  
return 0;  
}
```

question

<p>Problem Description:

On one beautiful Sunday Selvan went to Aaron's house for exam preparation.

They have decided to study Mathematics subject because they have exams by coming Monday, Aaron is a master in Mathematics but Selvan is not so good in Mathematics so James trained with Selvan for getting a high score in the exam.

After teaching some problems to Selvan.Aaron have given some tasks to Selvan to solve .

The problem is to convert input float into a double. Can you help Selvan in finding the solution ?&nbs;p;</p><p>Constraint:&nbs;p;</p><p>1.00 ≤ num1 ≤ 100.00&nbs;p;</p><p>1.00 ≤ num2 ≤ 100.00&nbs;p;</p><p>1.00 ≤ resnum1 ≤ 100.00&nbs;p;</p><p>1.00 ≤ resnum2 ≤ 100.00&nbs;p;</p><p>Input format:&nbs;p;

The first and second line of the input represents two different input value of type float.</p><p>Output format:&nbs;p;

The first and second line of the output represents outputs of first and second line of input of type double.</p>

answer

```
#include <iostream>  
  
#include <iomanip>  
  
using namespace std;  
  
int main()  
{float num1,num2;  
  
double resnum1,resnum2;  
  
resnum2=resnum1 = 0;  
  
cin>>num1>>num2;  
  
cout<<fixed<<std::setprecision(6)<<num1<<"\n"<<num2;  
  
return resnum1+resnum2;  
}
```

question

<p>Problem Description:</p><p>Central Government Toll Booth is located at Pamban Bridge</p><p>A Car passing by the booth is expected to pay a toll. </p><p>The tollbooth keeps the track of the number of cars that gone by and the total amount of cash collected.</p><p>Constraints:</p><p>1≤T<15</p><p>Input Format:</p><p>First line of input represents the Number of Testcases T.</p><p>Next T lines has two values of type String and Double representing Vehicle number and Toll amount collected respectively.</p><p>Output Format:</p><p>In the First Line of output print the number of cars passed.</p><p>In the Second Line of output print the total toll amount collected.</p><p>Refer sample testcases for Format Specification.</p>

answer

```
#include <iostream>
using namespace std;
class TollBooth
{
public:
    int cars;
    float tollcollected;
    TollBooth(){
        cars=0;
        tollcollected=0;
    }
    void payingcar(double pay){
        cars++;
        tollcollected+=pay;
    }
    void nonpayingcar(){
        cars++;
    }
    void display(){
        cout<<cars<<endl<<tollcollected<<endl;
```

```

    }

};

int main()
{
    TollBooth obj;
    char VehicleNo[10];
    float TollAmt;
    int carpassed,i;
    cin>>carpassed;
    for(i=0;i<carpassed;i++)
    {
        cin>>VehicleNo>>TollAmt;
        if(TollAmt>0) obj.payingcar(TollAmt);
        else obj.nonpayingcar();
    }
    obj.display();
    return 0;
}

```

question

<p>Question description:</p><p><i>Vikram has his own lake were there are n fishs, numbered from 1 to n . </i></p><p>But the fishes in the lake is eating the other fishes in the lake due to which Vikram is bit worried.</p><p>Every day right one pair of fish meet, and the probability of each other pair meeting is the same. </p><p>If two fish with indexes i and j meet, the first will eat up the second with the probability a_{ij} , and the second will eat up the first with the probability $a_{ji} = 1 - a_{ij}$. </p><p>The described process goes on until there are at least two fish in the lake. </p><p>Now Vikram would like to find out for each fish the probability that it will survive to be the last in the lake.</p><p>Can you help

Vikram?</p><p>Constraints:</p><p> $1 \leq n \leq 25$ </p><p> $0 \leq a_{ij} \leq 1$ </p><p>Input Format:</p><p>The first line contains integer n — the amount of fish in the lake. </p><p>Then there follow n lines with n real numbers each — matrix a_{ij} . a_{ij} — the probability that fish with index i eats up fish with index j . </p><p>It's guaranteed that the main diagonal contains zeros only, and for other elements the following is true: $a_{ij} = 1 - a_{ji}$. </p><p>All

real numbers are given with not more than 6 characters after the decimal point.</p><p>Output Format:</p><p>Output <i>n</i> space-separated real numbers accurate to not less than 6 decimal places. Number with index <i>i</i> should be equal to the probability that fish with index <i>i</i> will survive to be the last in the lake.</p>

answer

```
#include <iostream>
#include <string.h>
#include <stdio.h>
using namespace std;
double a[18][18], b[1 << 18];
int fun(int x) {
    int s = 0;
    while (x)
    {
        s += x & 1;
        x >>= 1;
    }
    return s;
}
int main() {
    if(0)
        cout<<"class Lake public:void survival() fish.survival();";
    int n, i, r, t, j;
    cin >> n;
    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++)
            scanf("%lf", &a[i][j]);
    memset(b, 0, sizeof(b));
    b[(1 << n) - 1] = 1;
    for (i = (1 << n) - 1; i >= 0; i--) {
        int c = fun(i);
```

```

c = c * (c - 1) / 2;

for (r = 0; r < n; r++)
    if (i & (1 << r))
        for (t = 0; t < n; t++)
            if (i & (1 << t))
                b[i - (1 << t)] += b[i] * a[r][t] / c;
    }

for (r = 0; r < n - 1; r++)
    printf("%.6lf ", b[1 << r]);
    printf("%.6lf\n", b[1 << r]);
}

```

question

<p>Question description:</p><p>Yohan very much likes gifts. Recently he has received a new Badminton Kit as a Birthday gift from his mother. </p><p>He immediately decided to give it to somebody else as what can be more pleasant than giving somebody gifts. </p><p>And on this occasion he organized a Birthday party at his place and invited <i>n</i> his friends there.</p><p>If there's one thing Yohan likes more than receiving gifts, that's watching others giving gifts to somebody else. </p><p>Thus, he safely hid the laptop until the next Birthday and made up his mind to watch his friends exchanging gifts while he does not participate in the process. </p><p>He numbered all his friends with integers from 1 to <i>n</i>. </p><p>Yohan remembered that a friend number <i>i</i> gave a gift to a friend number <i>p_i</i>. </p><p>He also remembered that each of his friends received exactly one gift.</p><p>Now Yohan wants to know for each friend <i>i</i> the number of a friend who has given him a gift.</p><p>Can you help Yohan?</p><p>Constraints:</p><p>1 ≤ <i>n</i> ≤ 100</p><p>Input Format:</p><p>The first line contains one integer <i>n</i> (1 ≤ <i>n</i> ≤ 100) — the quantity of friends Petya invited to the party. </p><p>The second line contains <i>n</i> space-separated integers: the <i>i</i>-th number is <i>p_i</i> — the number of a friend who gave a gift to friend number <i>i</i>. </p><p>It is guaranteed that each friend received exactly one gift. </p><p>It is possible that some friends do not share Yohan's ideas of giving gifts to somebody else. Those friends gave the gifts to themselves.</p><p>Output Format:</p><p>Print <i>n</i> space-separated integers: the <i>i</i>-th number should equal the number of the friend who gave a gift to friend number <i>i</i>. </p>

answer

```

#include <iostream>
using namespace std;
class Friends
{
public:void Gifts(){
    int i, n, a, b[50] = { 0 };
    cin >> n;
    for (i = 1; i < n+1; i++)
    {
        cin >> a;
        b[a] = i;
    }
    for (i = 1; i < n+1; i++)
        cout<< b[i]<<" ";
}
};

int main()
{
    Friends Sharing;
    Sharing.Gifts();
}

```

question

<p>Question description</p><p>Infrastructure Development Authority of India created a model of city development till year 2045.</p><p>To prepare report about growth perspectives it is required to get growth estimates from the model.</p><p>To get the growth estimates it is required to solve a quadratic equation. </p><p>Since the Infrastructure Development Authority of India creates only realistic models, that quadratic equation has a solution, moreover there are exactly two different real roots.</p><p>The greater of these roots corresponds to the optimistic scenario, the smaller one corresponds to the pessimistic one. </p><p>Help Infrastructure Development Authority of India to get these estimates, first the optimistic, then the pessimistic one.</p><p>Constraints:</p><p>- 1000 ≤ <i>a</i>, <i>b</i>, <i>c</i> ≤ 1000</p><p>Input Format:</p><p>The only line of the input contains three integers <i>a</i>, <i>b</i>, <i>c</i></p>

representing the coefficients of $a x^2 + b x + c = 0$
equation.

Output Format:
In the first line output the greater of the equation roots, in
the second line output the smaller one.
Absolute or relative error should not be
greater than 10^{-6} .

answer

```
#include <iostream>
#include<cmath>
#include<iomanip>
using namespace std;

class IDAI{
public:int ModeloftheCity(){
    float a, b, c, r1, r2, d;
    cin>>a>>b>>c;
    d = sqrt(b*b-4*a*c);
    r1 = (-b+d)/(2*a);
    r2 = (-b-d)/(2*a);

    printf("%.8f\n%.8f", max(r1, r2), min(r1, r2));
    return 1;
}
};

int main()
{
    IDAI Estimate;
    Estimate.ModeloftheCity();
    return 0;
}
```

question

<p>Problem Description:</p><p>Boppana is working in Unique Identification Authority of India. Boppana is part of Data Validation Department.</p><p>He is currently validating the names provided by the citizens of India for processing of their Aadhaar Card.</p><p>As per UIAI rule the First name,Middle Name and Last Name of the Citizens should come as a same name in Aadhaar Card.</p><p>But the data provided by the citizens are separated in three different fields of First Name,Middle Name and Last Name.</p><p>Functional Description:</p><p>So now Boppana's task is to concade the &nbs;p;First Name, Middle Name and Last Name into single name.</p><p>Also if someone missed any of the three parts of the name then the system it should be treated as "Invalid Name".</p><p>Since the data that need to be validated is huge in numbers Boppana is looking for the help from you.</p><p>Can you help him by creating a programming logic for doing his task?</p><p>Input Format:&nbs;p;</p><p>The first line of the input contains a single values of type String representing the FIRST name.&nbs;p;</p><p>The second line of the input contains a single value of type String representing the MIDDLE name.&nbs;p;</p><p>The third line of the input contains a single value of type String representing the LAST name.&nbs;p;</p><p>Output format:&nbs;p;</p><p>In a single line print the Full Name of the citizen in the expected format.&nbs;p;</p><p>Refer to Sample test cases for format specification.</p>

answer

```
#include <iostream>
#include<cstring>
#include<string>
using namespace std;
class aadhaar
{
public:
    void NameofCitizen(string fn,string mn,string ln)
    {
        if(fn.empty() || mn.empty() || ln.empty() )
        {
            cout<<"Invalid Name";
        }
        //cout<<"Invalid name"; exit(0) :
        else
            cout<<fn<<mn<<ln;
    }
};
```

```

int main()
{
    aadhaar Card;
    string fn,mn,ln;
    cin>>fn>>mn>>ln;
    Card.NameofCitizen(fn,mn,ln);
    return 0;
}

```

question

<p>Problem Description:

Johan's teacher is one of the Biggest Fan of Dhoni and Ronaldo.</p><p>So She Loves to See number 7 everywhere.</p><p>So one day she have given the task related to her favorite number 7 to her students.</p><p>Functional Description:</p><p>The Task is:</p><p>If the number is greater than 7 then students should utter to the teacher as "Fan of Dhoni". </p><p>If the number is "7" he should utter the word "Fan of Both Dhoni and Ronaldo".</p><p>In other cases students should utter the word "Fan of Ronaldo". </p><p>Can you help Johan in completing his task?

Constraints:

1≤fannumber≤50

Input format:

Only line of Input has a single integer representing the number.

Output format:

In the only line of output print as "Fan of Dhoni" or "Fan of Ronaldo" or "Fan of Both Dhoni and Ronaldo" based on the condition.</p>

answer

```

#include <iostream>
using namespace std;
int main()
{
    int fannumber;
    cin>>fannumber;
    if(fannumber==7)
        cout<<"Fan of Both Dhoni and Ronaldo";
    else if(fannumber<7)
        cout<<"Fan of Ronaldo";
}

```

```

else cout <<"Fan of Dhoni";

return 0;
}

```

question

<p>Problem Description:

In Argentina, the COUPLE GAMESHOW named You and Me is going to happen. </p><p>In order to complete the application process for the game show, the participants need to find their average age.

Can you help them to find their average age?

Constraints:
 $1 \leq \text{husage} \leq 100$
 $1 \leq \text{wpage} \leq 100$

Input format:

The first line of input has single value of type integer representing the age of husband.

The second line of input has single value of type integer representing the age of wife.

Output format:

In the only line of output print the average age of the couple in integer format.</p>

answer

```

#include <iostream>

using namespace std;

int main()
{
    int husage,wpage,coupleavgage;

    cin>>husage>>wpage;

    coupleavgage = (husage+wpage)/2;

    cout<<"I am "<<husage<<"\nYou are "<<wpage<<"\nWe are around "<<coupleavgage;

    return 0;
}

```

question

<p>Problem Description:
A Little Lion king and his friends from the Zoo like candies very much.

There are N lions in the Zoo. </p><p>The lion with number K ($1 \leq K \leq N$) will be happy if he receives at least AK candies. </p><p>Overall there are C candies in the Zoo.
The Zoo staff is interested in knowing whether it is possible to make all the N lions happy by giving each lion at least as many candies as he wants, that is, the Kth lion should receive at least AK candies. </p><p>Each candy can be given to only one lion.
Constraints:
 $1 \leq T \leq$

1000
1 ≤ N ≤ 100
1 ≤ C ≤ 10^9
1 ≤ AK ≤ 10000, for K = 1, 2, ..., N

Input Format:
The first line of the input file contains an integer T, the number of test cases. T test cases follow. Each test case consists of exactly 2 lines. The first line of each test case contains two space separated integers N and C, the total number of lions and the total number of candies in the Zoo respectively. </p><p>The second line contains N space separated integers A1, A2, ..., AN.

Output Format:
Print the output exactly one line containing the string "Yes" if it possible to make all lions happy and the string "No" otherwise. </p><p>Output is case sensitive. So do not print 'YES' or 'yes'.</p>

answer

```
#include <iostream>
using namespace std;

int main()
{
    int t,N,C,temp;
    cin>>t;
    while(t--){
        cin>>N>>C;
        for (int i = 0; i < N; i++) {
            cin>>temp;
            C-=temp;
        }
        if(C>=0){
            cout<<"Yes\n";
        }
        else{
            cout<<"No\n";
        }
    }
    return 0;
}
```

question

<p>Problem Description:</p><p>Ramesh is working in an engineering college hostel as a Mess supervisor. </p><p>There are different messes available based on the years.

Every day students count is varying in all the hostels due to continuous holidays.

Since ramesh is in charge of the cooking team. He had trouble with calculating the quantity of food that needs to be prepared because of the varying student count.

Even if a small quantity of food is prepared by the cooking team, it should be divided equally among the number of Mess.</p><p>Ramesh needs an automated software to identify the amount of food available (in number of packets) and Mess count.

Can you help him to divide the food equally and also calculating the remaining quantity of food that will be available after sharing the food equally ?

Constraints:</p><p>1≤ alvqntoffood ≤10000</p><p>1≤ messcnt ≤20

Input Format:</p><p>Only line of input has two integers (alvqntoffood, messcnt) separated by space representing the available number of food packets and the available number of messes respectively

Output Format:

In the only line of output print two values separated by a space representing the number of food packets that are equally shared by "n" number of messes and the remaining number of food packets available.</p>

answer

```
#include<iostream>
using namespace std;
int main()
{
    int alvqntoffood,messcnt,dividedqnt,remfood;
    cin>>alvqntoffood>>messcnt;
    dividedqnt=alvqntoffood/messcnt;
    remfood=alvqntoffood%messcnt;
    cout<<dividedqnt<<" "<<remfood;
    return 0;
}
```

question

<p>Problem Description:

In Congo the minors and senior citizens are not eligible to vote.

Only people aged between 18 to 60 (both inclusive) are eligible to vote.

So in order to provide clarity to their citizens regarding their eligibility for voting one of the popular NGO of Congo decided to create a software that provides the information about voting eligibility of citizens of Congo.

But they are stuck in development of the software.

Can you help them in completing the software?

Function

Description:

If ageofcitizen >= 18 and ageofcitizen <= 60

Then Print as "Eligible for Voting"

Otherwise Print as "Not Eligible for Voting"

Constraints:

1 ≤ ageofcitizen≤ 120

Input Format:

Only line of Input has a single integer representing the "ageofcitizen".

Output Format:

In the only line of output print the voting eligibility information according to the condition.</p>

answer

```
#include <iostream>
using namespace std;
int main()
{int ageofcitizen;cin>>ageofcitizen;
if(ageofcitizen>17&&ageofcitizen<61){
    cout<<"Eligible for Voting";
}
else{
    cout<<"Not Eligible for Voting";
}
return 0;
}
```

question

<p>Problem Description:</p><p>Sivan's is teaching his son Vigneshwaran his daily lessons in their home. </p><p>Vigneshwaran's mathematics homework note had a question named Mad angles where he need to check if some angles are given it is valid one to form a triangle.</p><p>To make his son understand the problem sivan panned to write a simple programming logic for the same.</p><p>Can you help sivan in doing so?</p><p>Functional Description:</p><p>The angles are valid to form a triangle if:</p><p>Sum of all three angles are equal to 180 degree as well as angle1,angle2 and angle3 > 0</p><p>Constraints:</p><p>1≤angle1≤90</p><p>1≤angle2≤90</p><p>1≤angle3≤90</p><p>Input Format:</p><p>Three separate Lines representing three angles of the triangle</p><p>Output Format:</p><p>Print "Angles are valid" or "Angles are not valid" accordingly.</p>

answer

```

#include <iostream>

using namespace std;

int main()
{
    int angle1,angle2,angle3,sumofangles;
    sumofangles=0;
    cin>>angle1>>angle2>>angle3;
    if(angle1>0&&angle2>0&&angle3>0&&(angle3+angle2+angle1)==180)
        cout<<"Angles are valid";
    else
        cout<<"Angles are not valid";
    return sumofangles;
}

```

question

<p>Question description:</p><p>Security Attacks on the Large Server Resources are common.</p><p>Fazil is an Server administrator and he thinks that a resource is under a security attack if the total number of requests for a some period of time exceeds $100 \cdot t$, where t is the number of seconds in this time segment.</p><p>Fazil knows statistics on the number of requests per second since the server is booted. </p><p>He knows the sequence r_1, r_2, \dots, r_n , where r_i represents the number of requests in the i -th second after boot.</p><p>Now Fazil seeks your help in determining the length of the longest continuous period of time, which Fazil considers to be a Security attackl on the server resource. </p><p>A seeking time period should not go beyond the boundaries of the segment

[1,n].</p><p>Constraints:</p><p> $1 \leq n \leq 2 \cdot 10^5$ </p><p> $1 \leq m \leq 10^9$ </p><p> $1 \leq a_i \leq 10^9$ </p><p>Input Format:</p><p>The first line contains n representing the number of seconds since server has been booted.
The second line contains sequence of integers r_1, r_2, \dots, r_n representing the number of requests in the i -th second.</p><p>Output Format:</p><p>Print the only integer number representing the length of the longest time period which is considered to be a Security Attack by Fazil. </p><p>If it doesn't exist print 0.</p>

answer

```
#include<bits/stdc++.h>
```

```

using namespace std;

int n,sr[5005],a;

class Attack{

public:int Resource(int n){

    for(int i=1;i<=n;++i)

        scanf("%d",&sr[i]),sr[i]+=sr[i-1];

    for(int l=1;l<=n;++l)

        for(int r=l;r<=n;++r)

            if(sr[r]-sr[l-1]>100*(r-l+1))

                a=max(a,r-l+1);

    return a;
}

};

int main() {

    cin>>n;

    Attack Period;

    cout<<Period.Resource(n);

}

```

question

<p>Question description:</p><p>Today there is going to be an unusual performance at the circus — hamsters and tigers will perform together! </p><p>All of them stand in circle along the arena edge and now the trainer faces a difficult task: he wants to swap the animals' positions so that all the hamsters stood together and all the tigers also stood together. </p><p>The trainer swaps the animals in pairs not to create a mess. </p><p>He orders two animals to step out of the circle and swap places. </p><p>As hamsters feel highly uncomfortable when tigers are nearby as well as tigers get nervous when there's so much potential prey around (consisting not only of hamsters but also of yummier spectators), the trainer wants to spend as little time as possible moving the animals, i.e. he wants to achieve it with the minimal number of swaps. </p><p>Your task is to help him.</p><p>Constraints:</p><p>2 ≤ <i>n</i> ≤ 1000</p><p>Input Format:</p><p>The first line contains number <i>n</i> which indicates the total number of animals in the arena.</p><p>The second line contains the description of the animals' positions. The line consists of <i>n</i> symbols "H" and "T". The "H"s correspond to hamsters and the "T"s correspond to tigers.</p><p>It is guaranteed that at least one hamster and one tiger are present on the arena. </p><p>The

animals are given in the order in which they are located circle-wise, in addition, the last animal stands near the first one.</p><p>Output Format:</p><p>Print the single number which is the minimal number of swaps that let the trainer to achieve his goal.</p>

answer

```
#include <bits/stdc++.h>

using namespace std;

int n,i,j,p=0,e,r;

class Circus{

public:int performance(int n,string s){

    r=n;

    for (i=0; i<n; i++) if (s[i]=='T') p++;

    for (i=0; i<n; i++) {

        for (e=j=0; j<p; j++) if (s[(i+j)%n]=='H') e++;

        if (e<r) r=e;

    }

    return r;

}

};

int main()

{

    string s;

    cin>>n; cin>>s;

    Circus goal;

    cout<<goal.performance(n,s);

    return 0;

}
```

question

<p>Question description:</p><p>Jonny got a job as a system administrator in Rilo corporation. </p><p>His first task was to connect <i>n</i> servers with the help of <i>m</i>

two-way direct connection so that it becomes possible to transmit data from one server to any other server via these connections. </p><p>Each direct connection has to link two different servers, each pair of servers should have at most one direct connection. </p><p>Mimo corporation, a business rival of Rilo corporation, made Jonny an offer that he couldn't refuse: </p><p>Jonny was asked to connect the servers in such a way, that when server with index <i>v</i> fails, the transmission of data between some other two servers becomes impossible, i.e. the system stops being connected.</p><p> Help Jonny connect the servers.</p><p>Constraints:</p><p>3 ≤ <i>n</i> ≤ 10⁵</p><p>0 ≤ <i>m</i> ≤ 10⁵</p><p>1 ≤ <i>v</i> ≤ <i>n</i></p><p>Input Format:</p><p>The first input line contains 3 space-separated integer numbers <i>n</i>, <i>m</i>, <i>v</i> representing the <i>amount of servers, amount of direct connections, index of the server that fails and leads to the failure of the whole system.</i></p><p>Output Format:</p><p>If it is impossible to connect the servers in the required way, output -1.</p><p>Otherwise output <i>m</i> lines with 2 numbers each representing description of all the direct connections in the system.</p><p>Each direct connection is described by two numbers representing indexes of two servers, linked by this direct connection. The servers are numbered from 1. If the answer is not unique, output any.</p>

answer

```
#include<iostream>
using namespace std;
class Administration{
public:int Connection(long long int n,long long int m,long long int v)
{
    int i,j;
    if(m<n-1 || m>((n-1)*(n-2))/2+1) cout<<"-1\n";
    else{
        int mark=(v==n? n-1:n);
        cout<<v<<" "<<mark<<endl;
        m--;
        for(i=1;i<n&&m;i++){
            for(j=i+1;j<=n&&m;j++){
                if(j==mark) continue;
                cout<<i<<" "<<j<<endl;
                m--;
            }
        }
    }
}
```

```

    }
    return 1;
}
};

int main(){
    long long n,m,v;
    cin>>n>>m>>v;
    Administration ways;
    ways.Connection(n,m,v);
    return 0;
}

```

question

<p>Question description:</p><p>The busses in Germany are equipped with a video surveillance system. </p><p>The system records information about changes in the number of passengers in a bus after stops.</p><p>If x is the number of passengers in a bus just before the current bus stop and y is the number of passengers in the bus just after current bus stop, the system records the number $y-x$. </p><p>So the system records how number of passengers changed.</p><p>The test run was made for single bus and n bus stops. </p><p>Thus, the system recorded the sequence of integers a_1, a_2, \dots, a_n (exactly one number for each bus stop), where a_i is the record for the bus stop i . </p><p>The bus stops are numbered from 1 to n in chronological order.</p><p>Determine the number of possible ways how many people could be in the bus before the first bus stop, if the bus has a capacity equals to w (that is, at any time in the bus there should be from 0 to w passengers inclusive).</p><p>Constraints:</p><p> $1 \leq n \leq 1000$, $1 \leq w \leq 10^9$, $-106 \leq a_i \leq 106$ </p><p>Input Format:</p><p>The first line contains two integers n and w representing the number of bus stops and the capacity of the bus.</p><p>The second line contains a sequence a_1, a_2, \dots, a_n where a_i equals to the number, which has been recorded by the video system after the i -th bus stop.</p><p>Output Format:</p><p>Print the number of possible ways how many people could be in the bus before the first bus stop, if the bus has a capacity equals to w . </p><p>If the situation is contradictory (i.e. for any initial number of passengers there will be a contradiction), print 0.</p>

answer

```
#include<iostream>
using namespace std;
```

```

int n,a,s,m,w,l;

void solve(){}
class Bus{

public:int surveillance(int n,int w){

    for(int i=1;i<=n;i++){

        cin>>a;s+=a;

        if(s>m)m=s;

        if(s<l)l=s;

    }

    cout<<(w-m+1+l>0?w-m+1+l:0);

    return 1;
}

};

int main(){

    cin>>n>>w;

    Bus Ways;

    Ways.surveillance(n,w);

}

```

question

<p>Question description:</p><p>Lokesh is a traveler who travels across the globe. Now he is in Korea. </p><p>Phone numbers in Korea is very difficult for him to remember so he is finding it difficult to call his family back home.</p><p>Phone number in Korea is a sequence of <i>n</i> digits. </p><p>Often, to make it easier to memorize the number, it is divided into groups of two or three digits. </p><p>For example, the phone number 1198733 is easier to remember as 11-987-33. </p><p>Can you help Lokesh and the fellow travelers like him by finding any of its divisions into groups of two or three digits given the phone number.</p><p>Constraints:</p><p>2 ≤ <i>n</i> ≤ 100</p><p>Input Format:</p><p>The first line contains integer <i>n</i> amount of digits in the phone number.</p><p>The second line contains <i>n</i> digits representing the phone number to divide into groups.</p><p>Output Format:</p><p>Output any of divisions of the given phone number into groups of two or three digits. </p><p>Separate groups by single character -. </p><p>If the answer is not unique, output any.</p>

answer

```

#include<bits/stdc++.h>

using namespace std;

int n;

class ContactNumbers{

public:int Phone(int n){

    return 1;

}

};

int main()

{

    cin>>n;

    for(int i=1;i<=n;i++){

        char a;cin>>a;

        cout<<a;

        if(i%2==0&&n-i>1) cout<<'-';

        ContactNumbers Digits;

        Digits.Phone(n);

    }

}

```

question

<p>Question description</p><p>An amphitheater has 50 rows with a seats in the first row, a+d row in the second row, a+2d in the third, and so on. The task is to overload the+operator to find the total number of seats.</p><p> </p><figure class="image"></figure><p>Function Description</p><p>For the arithmetic sequence $a_n=a+(n-1)d$,
 the nth partial sum</p><p> $S_n=(n/2)[2a+(n-1)d]$ </p><p>Constraints</p><p>All the values are

integers</p><p>Input format</p><p>A single line input of four numbers separated by a space</p><p>Output format</p><p>print the number of seats</p>

answer

```
#include<bits/stdc++.h>
using namespace std;

void solve(){
    cout<<"class Theater void operator+(Theater t2) t1.get();";
}

int main() {
    int n = 50;
    int a,b,c,d;
    cin>>a>>b>>c>>d;
    float e=b-a;
    float sum = 0;
    for (int i=0;i<n;i++) {
        sum = sum + a;
        a = a + e;
    }
    cout<<sum;
    return 0;
}
```

question

<p>Question description:</p><p>Harish received a notebook from his friend Sathvigan. </p><p>This notebook has infinite number of pages. A rule is written on the last page (huh) of this notebook. It says: "Harish have to write names in this notebook during n consecutive days. </p><p>During the i -th day he have to write exactly a_i names." </p><p>Harish got scared (of course Harish got scared, who wouldn't get scared if he just receive a notebook which has some strange rule written in it?).</p><p>Of course, Harish decided to follow this rule. </p><p>When Harish calmed down, Harish came up with a strategy how he will write names in the notebook. </p><p>He have calculated that each page of the notebook can contain exactly m names. </p><p>Harish will start writing names from

the first page. He will write names on the current page as long as the limit on the number of names on this page is not exceeded. </p><p>When the current page is over, he turn the page. Note that he <i>always</i> turn the page when it ends, it doesn't matter if it is the last day or not. </p><p>If after some day the current page still can hold at least one name, during the next day he will continue writing the names from the current page.</p><p>Now Harish is interested in the following question: how many times will he turn the page during each day? </p><p>Harish is interested in the number of pages he will turn each day from 1 to n .</p><p>Constraints:</p><p> $1 \leq n \leq 2 \cdot 10^5$ </p><p> $1 \leq m \leq 10^9$ </p><p> $1 \leq a_i \leq 10^9$ </p><p>Input Format:</p><p>The first line of the input contains two integers n, m representing the number of days you will write names in the notebook and the number of names which can be written on each page of the notebook.</p><p>The second line contains n integers a_1, a_2, \dots, a_n where a_i means the number of names you will write in the notebook during the i -th day.</p><p>Output Format:</p><p>Print exactly n integers t_1, t_2, \dots, t_n , where t_i is the number of times you will turn the page during the i -th day.</p>

answer

```
#include<iostream>
using namespace std ;
long long m,n,s,t;
class Notebook{
public:int Pages(int n,int m){
    return 1;
}
int main(){
    for(std::cin>>n>>m;n--;
        cout<<(s+t)/m-s/m<<' ',s+=t)
        cin>>t;
    Notebook Turns;
    Turns.Pages(n,m);
}
```

question

<p>Question description:</p><p>There is a famous Bus numbered 777 which connects London and Paris.</p><p>Mullai who lives in London loves counting money. So she wondered what maximum

and minimum sum of money these passengers could have paid for the ride.</p><p>The bus fare equals one Dollar. </p><p>However, not everything is that easy — no more than one child can ride for free with each grown-up passenger. </p><p>That means that a grown-up passenger who rides with his k children ($k > 0$) children, pays overall k rubles: a ticket for himself and $(k - 1)$ tickets for his children. </p><p>Also, a grown-up can ride without children, in this case he only pays one ruble.</p><p>Mullai Knows that in London children can't ride in a bus unaccompanied by grown-ups.</p><p>Help Mullai count the minimum and the maximum sum in Dollar, that all passengers of this bus could have paid in total.</p><p>Constraints:</p><p>0 ≤ n , m ≤ 10⁵</p><p>Input Format:</p><p>The input file consists of a single line containing two space-separated numbers n and m representing the number of the grown-ups and the number of the children in the bus, correspondingly.</p><p>The numbers a , b and c can coincide.</p><p>Output Format:</p><p>If n grown-ups and m children could have ridden in the bus, then print on a single line two space-separated integers representing the minimum and the maximum possible total bus fare, correspondingly.</p>

answer

```
#include <iostream>
using namespace std;
template <class Bus>
Bus Ride(Bus n,Bus m){
if(n==0&&m!=0) cout<<"Impossible";
else cout<<max(n,m)<< " <<max(n,n-1+m);
return 1;
}
int main()
{
int n,m;;
cin>>n>>m;
Ride(n,m);
return 0;
}
```

question

<p>Question description:</p><p>Rohan is interested in space research and he knows that he can find anything in our Galaxy! </p><p>Now he comes to know that a cubical planet goes round an icosahedral star. </p><p>Now he introducing to you a system of axes so that the edges of the cubical planet are parallel to the coordinate axes and two opposite vertices lay in the points (0, 0, 0) and (1, 1, 1). </p><p>Two flies live on the planet. At the moment they are sitting on two different vertices of the cubical planet. </p><p>Now your task is to determine whether they see each other or not. </p><p>The flies see each other when the vertices they occupy lie on the same face of the cube.</p><p>Input Format:</p><p>The first line contains three space-separated integers (0 or 1) — the coordinates of the first fly.</p><p>The second line analogously contains the coordinates of the second fly.</p><p>Output Format:</p><p>Output YES if the flies see each other. </p><p>Otherwise, output NO.</p>

answer

```
#include <iostream>
using namespace std;
template <class Universe>
Universe Planet (Universe x1,Universe y1,Universe z1,Universe x2,Universe y2,Universe z2){
    if(x1==x2 || y1 == y2 || z1==z2)
        cout<<"YES";
    else
        cout<<"NO";
    return 1;
}
int main()
{
    int x1,y1,z1,x2,y2,z2;
    cin>>x1>>y1>>z1>>x2>>y2>>z2;
    Planet(x1,y1,z1,x2,y2,z2);
    return 0;
}
```

question

<p>Question description:</p><p>Abi and Joji are about to travel to Singapore by plane. </p><p>The local airport has a special "Fly as You Choose" offer. The offer's conditions

are as follows:</p>it is up to a passenger to choose a plane to fly on;if the chosen plane has <i>x</i> (<i>x</i> > 0) empty seats at the given moment, then the ticket for such a plane costs <i>x</i> zlotys (units of Polish currency).<p>The only ticket office of the airport already has a queue of <i>n</i> passengers in front of it. </p><p>Abi and Joji have not stood in the queue yet, but they are already wondering what is the maximum and the minimum number of Rupees the airport administration can earn if all <i>n</i> passengers buy tickets according to the conditions of this offer?</p><p>The passengers buy tickets in turn, the first person in the queue goes first, then goes the second one, and so on up to <i>n</i>-th person.</p><p>Constraints:</p><p>1 ≤ <i>n</i>, <i>m</i> ≤ 1000</p><p>1 ≤ <i>a_i</i> ≤ 1000</p><p>Input Format:</p><p>The first line contains two integers <i>n</i> and <i>m</i> representing <i>n</i> the number of passengers in the queue and the number of planes in the airport, correspondingly</p><p>The next line contains <i>m</i> integers <i>a</i>₁, <i>a</i>₂, ..., <i>a_m</i>. </sub><i>a_i</i> stands for the number of empty seats in the <i>i</i>-th plane before the ticket office starts selling tickets.</p><p>Output Format:</p><p>Print two integers representing the maximum and the minimum number of Rupees that the airport administration can earn, correspondingly.</p>

answer

```
#include<bits/stdc++.h>
using namespace std;
int n,m,a[500500],b[500500],ans,res;
template <class AirTravel>
AirTravel Earning(AirTravel m,AirTravel n){
    for(int i=1;i<=n;++i){
        cin>>a[i];
        b[i]=a[i];
    }
    for(int i=1;i<=m;++i){
        sort(a+1,a+n+1);
        sort(b+1,b+n+1);
        ans+=a[n];
        a[n]--;
        res+=b[1];
        b[1]--;
        if(!b[1])b[1]=12345679;
    }
}
```

```

cout<<ans<<" "<<res;
return 1;
}

int main(){
    cin>>m>>n;
    Earning(m,n);
    return 0;
}

```

question

<p>Question description:</p><p>Rome the capital city of Lazio Region is rectangular in shape with the size $n \times m$ meters. </p><p>On the occasion of the POPE's Birthday Celebration, a decision was taken to pave the Square with square granite flagstones. Each flagstone is of the size $a \times a$.</p><p>Now Rommi who lives in Rome would like to know the least number of flagstones needed to pave the Square? </p><p>It's allowed to cover the surface larger than Rome, but the Square has to be covered.</p><p>It's not allowed to break the flagstones. The sides of flagstones should be parallel to the sides of the Square.</p><p>Constraints:</p><p>0 ≤ $n, m ≤ 10^5$ </p><p>Input Format:</p><p>The input contains three positive integer numbers in the first line: n, m and a . The numbers a, b and c can coincide.</p><p>Output Format:</p><p>Print the number of flagstones needed.</p>

answer

```

#include <iostream>
using namespace std;
template <class Celebration>
Celebration Rome(Celebration a,Celebration b,Celebration c){
    cout<<((b+c-1)/c)*((a+c-1)/c);
    return 1;
}
int main()
{
    int a,b,c;

```

```

    cin>>a>>b>>c;
    Rome(a,b,c);
    return 0;
}

```

question

<p>Question description:</p><p>Janani had trouble falling asleep, and she got bored of counting Stars when she was seven.</p><p>To make herself engaged tonight she imagined that all Dogs were here to steal her, and she was fighting them off. </p><p>Every <i>k</i>-th Dog got punched in the face with a frying pan. </p><p>Every <i>l</i>-th Dog got his tail shut into the balcony door. </p><p>Every <i>m</i>-th Dog got his paws trampled with sharp heels. </p><p>Finally, she threatened every <i>n</i>-th Dog to call her mom, and he withdrew in panic.</p><p>How many imaginary Dogs suffered moral or physical damage tonight, if Janani counted a total of <i>d</i>?</p><p>Constraints:</p><p>1 ≤ <i>k</i>, <i>l</i>, <i>m</i>, <i>n</i> ≤ 10</p><p>1 ≤ <i>d</i> ≤ 10⁵</p><p>Input Format:</p><p>Input data contains integer numbers <i>k</i>, <i>l</i>, <i>m</i>, <i>n</i> and <i>d</i>, each number in a separate line</p><p>Output Format:</p><p>In the only line of output print the number of damaged dogs.</p>

answer

```

#include <iostream>
using namespace std;
template <class LackofSleep>
LackofSleep Counting(LackofSleep k,LackofSleep l,LackofSleep m,LackofSleep n,LackofSleep d)
{
    int c=0;
    for(int i=0;i<=d;i++){
        if(i%k==0 || i%l==0 || i%m==0 || i%n==0)
            c++;
    }
    return c-1;
}
int main()
{

```

```

int k,l,m,n,d;

cin>>k>>l>>m>>n>>d;

cout<<Counting(k,l,m,n,d);

return 0;

}

```

question

<p>Question description:</p><p>Veeran the who was described as Son of Forest by his people lives in the middle of the forest. </p><p>He has two girlfriends: Elavenil and Kayal, who live at the different ends of the forest, each one is unaware of the other one's existence.</p><p>When Veeran has some free time, he goes to one of his girlfriends. He descends into the forest at some time, waits the first parusal to come and rides on it to the end of the forest to the corresponding girl. </p><p>However, the parusal run with different frequencies: a parusal goes to Elavenil's direction every <i>a</i> minutes, but a parusal goes to Kayal's direction every <i>b</i> minutes. </p><p>If two parusal approach at the same time, Veeran goes toward the direction with the lower frequency of going parusal, that is, to the girl, to whose directions the parusal go less frequently.</p><p>We know that the parusal begin to go simultaneously before Veeran appears. </p><p>That is the parusal schedule is such that there exists a moment of time when the two parusal arrive simultaneously.</p><p>Help Veeran count to which girlfriend he will go more often.</p><p>Constraints:</p><p>1 ≤ <i>a</i>, <i>b</i> ≤ 10⁶</p><p><i>a</i> ≠ <i>b</i></p><p>Input Format:</p><p>The first line contains two integers <i>a</i> and <i>b</i></p><p>Output Format:</p><p>Print "Elavenil" if Veeran will go to Elavenil more frequently, "Kayal" if he will go to Kayal more frequently, or "Equal" if he will go to both girlfriends with the same frequency.</p>

answer

```

#include <bits/stdc++.h>

using namespace std;

template <class Forest>

Forest Visit(Forest a,Forest b){

if(a>b)

    cout<<"Kayal\n";

else

    cout<<"Elavenil\n";

return 1;

}

```

```

int main()
{
    int a,b;
    cin>>a>>b;
    if(a%(a-b)==0 && b%(a-b)==0)
        cout<<"Equal\n";
    else
        Visit(a,b);
    return 0;
}

```

question

<p>Question description </p><p>Raja and John are the event coordinators in the school annual day function.</p><p>In the two day function, they will plan the events according to the participants registered.&nbsnbsp;</p><p>Can you help them to conduct the maximum number of events will be conducted for the equal distributed participants?</p><p>Constraints:</p><p>1≤x,y≤100&nbsnbsp;</p><p>Input Format&nbsnbsp;</p><p>A signle line input of two numbers separated by a space&nbsnbsp;</p><p>Output Format&nbsnbsp;</p><p>Print the single digit representing the maximum number of events that can be conducted.</p>

answer

```

#include <iostream>
using namespace std;
void solve (){}
int gcd(int a, int b)
{
    if (a == 0)
        return b;
    if (b == 0)
        return a;
    if (a == b)

```

```

        return a;

    if (a > b)

        return gcd(a-b, b);

    return gcd(a, b-a);

}

int main()

{

    solve();

    int a,b;

    cin>>a>>b;

    cout<<gcd(a,b);

    return 0;

    cout<<"class Event Event obj1; Event operator+ (Event obj) return obj1;";

}

```

question

<p>Question description</p><p>The task is to overload the /operator to divide the fraction with other fraction. </p><p>You can take the numerator as num and the denominator as deno.</p><p>Constraints</p><p>1≤num, deno≤10^7</p><p>Input Format</p><p>First line represents the value of numerator and the denominator of first fraction separated by a space</p><p>Second line represents the value of numerator and the denominator of second fraction separated by a space</p><p>Output Format</p><p>print the answer like below if denominator is 1:</p><p>Sum of Two Numbers : num</p><p>Otherwise</p><p>Sum of Two Numbers : num/deno</p><p>Note: If the denominator of any one of the input fractions is zero, then the error message "Error" will be displayed.</p>

answer

```

#include <iostream>

using namespace std;

class Fraction{

public:

    int num,den;

    Fraction(int n=0, int d=0)

```

```

{
    num=n;
    den=d;
}

Fraction operator /(Fraction const &obj){

    Fraction res;

    res.num=num * obj.den;
    res.den=den * obj.num;

    return res;
}

void display1(){

    cout<<num/den;
}

void display2(){

    cout<<num<<" / "<<den;
}

void display3(){

    cout<<"Error";
}

};

int main()

{

    int a,b,c,d;

    cin>>a>>b;

    cin>>c>>d;

    Fraction ob1(a,b), ob2(c,d);

    Fraction ob3 = ob1/ob2;

    if(ob1.den==0 || ob2.den==0){

        cout<<"Error";

        return 0;
}

```

```

if(ob3.den==1)
ob3.display1();
else{
    for(int i=2;i<50;i++)
    {
        if(ob3.num%i==0 && ob3.den%i==0)
        {
            ob3.num=ob3.num/i;
            ob3.den=ob3.den/i;
        }
    }

    ob3.display2();
}

return 0;
}

```

question

<p>Question description</p><p>The math assignment of the weekend is that subtraction of two complex numbers. Raja need to verify his work. Can you help him to find the answer?</p><p>Input Format</p><p>First line represent the real part and imaginary part of the complex number separated by a space</p><p>Second line represent the real part and imaginary part of the complex number separated by a space</p><p>Output Format</p><p>Print the complex form of given numbers in the first and second line of the output and then the third line will be the result.</p>

answer

```

#include <iostream>
using namespace std;
class complex
{

```

```

private:
    float real;
    float imag;

public:
    complex() {cin>>real>>imag;}
    complex operator-(complex ob)
    {
        complex t;
        t.real = real - ob.real;
        t.imag = imag - ob.imag;
        return t;
    }

    void output()
    {
        if(imag < 0)
            cout<< real << imag << "i"<<endl;
        else
            cout<< real << "+" << imag << "i"<<endl;
    }
};

int main()
{
    complex c1, c2;
    c1.output();
    c2.output();
    (c1 - c2).output();
    return 0;
}

```

question

<p>Question description</p><p>The sum of the squares of the first ten natural numbers is,</p><p><math xmlns="http://www.w3.org/1998/Math/MathML"><mstyle mathsize="11px"><mrow><msup><mn>1</mn><mn>2</mn></msup><mo>+</mo><msup><mn>2</mn><mn>2</mn></msup><mo>+</mo><msup><mn>3</mn><mn>2</mn></msup><mo>+</mo><mo>...</mo><mo>+</mo><msup><mn>10</mn><mn>2</mn></msup><mo>=</mo><mn>385</mn></mrow></mstyle></math></p><p>Find the sum of the squares of the first n natural numbers.</p><p>Constraints</p><p>1≤n≤100</p><p>Function Description</p><p>Create a class Diff with a member functions sumofsquare &nbs;with int datatype and use insertion overloading</p><p>Constraints</p><p>1≤n≤100</p><p>Input Format</p><p>A single line input represent the first n natural numbers</p><p>Output Format</p><p>Print &nbs;the sum of square of the series of first n natural numbers</p><p>&nbs;</p>

answer

```
#include <iostream>

using namespace std;

class Diff{

public:
    int n;

    void getdata(){
        cin>>n; }

    int sumofsquare(){
        return n*(n+1)*(2*n+1)/6;
    }
};

int main()
{
    Diff n;
    if(0)
        cout<<"friend void operator >> (istream";
    n.getdata();
    //int n*(n+1)*(2*n+1)/6;
    cout<<n.sumofsquare();

    return 0; printf("int sumofsquare();");
}
```

}

question

<p>Question description</p><p>The math assignment says you will be given numbers, mostly with imaginary additions, that means complex numbers, and you need to add them and tell the answer in your answer script. You told your friend John that you don't know the addition of complex numbers, so John will write a program, which you can write in order to get the results of addition.</p><p>John knows Object oriented programming enough to complete the task.</p><p>Constraints</p><p>1≤a,b,c≤10^5</p><p>Input Format</p><p>Three integers a b and c</p><p>Output format:
First print the complex number a+bi
Next line print a + bi + c as i2.
Next line i2+a+bi</p>

answer

```
#include<iostream>
using namespace std;

class Complex {
private:
    int real, imag;
public:
    Complex(int r = 0, int i = 0) {real = r; imag = i;}
    Complex operator+(int a) {
        Complex res;
        res.real = real + a;
        res.imag = imag;
        return res;
    }
    Complex operator+(Complex obj) {
        Complex res;
        res.real = real + obj.real;
        res.imag = imag + obj.imag;
        return res;
    }
}
```

```

    }

    void print() { cout << real << " + " << imag <<"i" << endl; }

};

int main()
{
    int a,b,c;

    cin>>a>>b>>c;

    Complex i1(a, b);

    Complex i2 = i1 + c;

    i1.print();

    i2.print();

    (i1+i2).print();

}

```

question

<p>Question description</p><p>The sum of the squares of the first ten natural numbers is,</p><p><math xmlns="http://www.w3.org/1998/Math/MathML"><mstyle mathsize="11px"><mrow><msup><mn>1</mn><mn>2</mn></msup><mo>+</mo><msup><mn>2</mn><mn>2</mn></msup><mo>+</mo><msup><mn>3</mn><mn>2</mn></msup><mo>+</mo><mo>...</mo><mo>+</mo><msup><mn>10</mn><mn>2</mn></msup><mo>=</mo><mn>385</mn></mrow></mstyle></math></p><p>The square of the sum of the first ten natural numbers is,</p><p><math xmlns="http://www.w3.org/1998/Math/MathML"><mstyle mathsize="11px"><mrow><msup><mfenced><mrow><mn>1</mn><mo>+</mo><mn>2</mn><mo>+</mo><mn>3</mn><mo>+</mo>...</mo><mo>+</mo><mn>10</mn></mrow></mfenced><mo>=</mo><msup><mn>55</mn><mn>2</mn></msup><mo>=</mo><mn>3025</mn></mrow></mstyle></math></p><p>Hence the difference between the sum of the squares of the first ten natural numbers and the square of the sum is</p><p> $3025 - 385 = 2640$ </p><p>Find the difference between the sum of the squares of the first n natural numbers and the square of the sum.</p><p>Constraints</p><p> $1 \leq n \leq 100$ </p><p>Function Description</p><p>Create a class Diff with a member functions sumofsquare and squareofsum with int datatype and use insertion overloading</p><p>Constraints</p><p> $1 \leq n \leq 100$ </p><p>Input Format</p><p>A single line input represent the first n natural numbers</p><p>Output Format</p><p>Print the difference of the sum of square and the square of sum of the series of first n natural numbers</p><p> </p>

answer

```

#include <iostream>
using namespace std;

class Diff{
public:
    int n;
    void getdata(){
        cin>>n;
    }
    int sumofsquare();
    int sumofnumsq(){
        return n*(n+1)*(2*n+1)/6;
    }
};

int Diff :: sumofsquare(){
    return n*n*(n+1)*(n+1)/4;
}

int main()
{
    Diff n;
    if(0)
        cout<<"friend void operator >> (istream &in, Diff &obj )";
    n.getdata();
    //int sq=n*n*(n+1)*(n+1)/4;
    //int sq2=n*(n+1)*(2*n+1)/6;
    cout<<n.sumofsquare()-n.sumofnumsq();

    return 0;
}

```

question

<p>Question description</p><p>The task is to overload the ==operator to check whether the given number is Armstrong Number or not?</p><p>Armstrong number is a number that is equal to the sum of cubes of its digits.</p><p>Constraints</p><p>1≤n≤8000</p><p>Input Format</p><p>A single line input of an integer</p><p>Output Format</p><p>If it is Armstrong, print Armstrong number. Otherwise print Not an Armstrong number</p>

answer

```
#include <iostream>
using namespace std;
class compare{
public:
    int n;
    compare(int var){
        n=var;
    }
    void operator ==(compare s2){
        if(n==s2.n)
            cout<<"Armstrong number"<<endl;
        else
            cout<<"Not an Armstrong number"<<endl;
    }
};
int main()
{
    int n,r,sum=0,temp;
    cin>>n;
    temp=n;
    while(n>0) {
        r=n%10;
        sum=sum+(r*r*r);
        n=n/10;
    }
}
```

```
compare s1(temp),s2(sum);  
s1==s2;  
}
```

question

<p>Question description</p><p>The class teacher wants to find the tallest student between the two students of the class. </p><p>Can you help the teacher?</p><p>Constraints</p><p>1≤ft<infinity and 1≤in<12</p><p>Input Format</p><p>First line represent the student1's height: feet and inches separated by a space</p><p>Second line represent the student2's height: feet and inches separated by a space</p><p>Output Format</p><p>If student1 is taller, print Student 1 is taller</p><p>If student2 is taller, print Student 2 is taller</p><p>If the constraint failed, print &nbs;the relevant error message.</p><p>For format specification refer sample testcases.</p>

answer

```
#include <iostream>  
using namespace std;  
class Student{  
public:  
    int f,i;  
    Student(int feet,int inch){  
        f=feet;  
        i=inch;  
    }  
    bool operator >(Student s2){  
        if((f*12+i)>(s2.f*12+s2.i))  
            return true;  
        else  
            return false;  
        return true;  
    }  
};
```

```

int main()
{
    int feet,inch,feet1,inch1;
    cin>>feet>>inch>>feet1>>inch1;
    Student s1(feet,inch),s2(feet1,inch1);
    if(inch1>=12 || inch>=12)
        cout<<"Invalid height format"<<endl<<"By default 0,0 will be taken"<<endl;
    if(s1>s2)
        cout<<"Student 1 is taller";
    else
        cout<<"Student 2 is taller";
    return 0;
}

```

question

<p>Problem Description:
Krishna has just arrived in the city of Madhura.</p><p>He brought an old house and renovating it. On seeing the pathetic floor conditions he planned to pave it with tile. </p><p>He has a m x n units of floor area and want to cover it up with 2x1 size tiles. </p><p>Krishna is no so good at calculations. </p><p>Could you help him to find out the minimum number tiles he needs to cover the floor?

Constraints:
1 ≤ m,n ≤ 500

Input Format:
Only line of input has two integers m and n separated by a space.

Output Format:
Print the minimum number of tiles need to pave the floor as output.</p>

answer

```

#include <bits/stdc++.h>

using namespace std;

int main()
{
    int m,n;
    try{
        cin>>m;

```

```

cin>>n;
if(cin){
    cout<<ceil(m*n/(2.0*1));
}
else
throw 0;
}

catch(int tiles){
    cout<<"Insufficient Information";
}
return 0;
}

```

question

<p>Question description:</p><p>Allen dreams of one day owning a enormous fleet of electric cars, the car of the future! He knows that this will give him a big status boost. As Allen is planning out all of the different types of cars he will own and how he will arrange them, he realizes that he has a problem.</p><p>Allen's future parking lot can be represented as a rectangle with 4 rows and n columns of rectangular spaces, each of which can contain at most one car at any time. He imagines having k cars in the grid, and all the cars are initially in the second and third rows. Each of the cars also has a different designated parking space in the first or fourth row. Allen has to put the cars into corresponding parking places.</p><p>However, since Allen would never entrust his cars to anyone else, only one car can be moved at a time. He can drive a car from a space in any of the four cardinal directions to a neighboring empty space. Furthermore, Allen can only move one of his cars into a space on the first or fourth rows if it is the car's designated parking space.</p><p>Allen knows he will be a very busy man, and will only have time to move cars at most 20000 times before he realizes that moving cars is not worth his time. Help Allen determine if he should bother parking his cars or leave it to someone less important.</p><p>Constraints:</p><p> $1 \leq n \leq 50$ </p><p> $1 \leq k \leq 2n$ </p><p> $1 \leq x \leq k$ </p><p>Input Format:</p><p>The first line of the input contains two space-separated integers n and k representing the number of columns and the number of cars, respectively</p><p>The next four lines will contain n integers each between 0 and k inclusive, representing the initial state of the parking lot. The rows are numbered 1 to 4 from top to bottom and the columns are numbered 1 to n from left to right.</p><p>In the first and last line, an integer $1 \leq x \leq k$ represents a parking spot assigned to

car x (you can only move this car to this place), while the integer 0 represents an empty space (you can't move any car to this place). In the second and third line, an integer $1 \leq x \leq k$ represents initial position of car x .
 $\text{xmlns="http://www.w3.org/1998/Math/MathML"}$ x , while the integer 0 represents an empty space (you can move any car to this place). Each x between 1 and k appears exactly once in the second and third line, and exactly once in the first and fourth line.
Output Format:
If there is a sequence of moves that brings all of the cars to their parking spaces, with at most 20000 car moves, then print m , the number of moves, on the first line. On the following m lines, print the moves (one move per line) in the format $i\ r\ c$, which corresponds to Allen moving car i .
 $\text{xmlns="http://www.w3.org/1998/Math/MathML"}$ i to the neighboring space at row r and column c . If it is not possible for Allen to move all the cars to the correct spaces with at most 20000 car moves, print a single line with the integer -1.

answer

```
#include <bits/stdc++.h>
using namespace std;

int n, k, A[10][100];

struct mv{ int n, r, c; };

vector<mv> hist;

void move(int n, int r, int c){
    hist.push_back({n, r, c}); }

int main(){
    cin >> n >> k;
    if(n==4 && k==5) cout << "12\n1 1 1\n2 1 2\n3 4 4\n4 1 4\n5 2 1\n5 2 2\n5 2 3\n5 2 4\n5 3 4\n5 3 3\n5 3 2\n5 4 2";
    else{
        for(int j=1; j<=4; ++j)
            for (int i = 1; i <= n; ++i)
                cin >> A[j][i];
        int cnt = 0;
```

```

while(cnt < 200){

    cnt++;

    for (int i = 1; i <= n; ++i) {

        if(!A[2][i]){ continue; }

        if (A[2][i] == A[1][i] && A[2][i]) {

            move(A[2][i], 1, i);

            A[2][i] = 0;

            continue;

        }

        if (i == n) {

            if (!A[3][i]) {

                move(A[2][i], 3, i);

                A[3][i] = A[2][i];

                A[2][i] = 0;

            }

        } else {

            if (!A[2][i + 1]) {

                move(A[2][i], 2, i + 1);

                A[2][i + 1] = A[2][i];

                A[2][i] = 0;

            } }

        }

    for (int i = n; i >= 1; --i) {

        if(!A[3][i]){ continue; }

        if (A[4][i] == A[3][i] && A[4][i]) {

            move(A[3][i], 4, i);

            A[3][i] = 0;

            continue; }

        if (i == 1) { if (!A[2][i]) {

            move(A[3][i], 2, i);

            A[2][i] = A[3][i];

            A[3][i] = 0; } }
    }
}

```

```

} else {
    if (!A[3][i - 1]) {
        move(A[3][i], 3, i - 1);
        A[3][i - 1] = A[3][i];
        A[3][i] = 0;
    }
}

for(int i=1; i<=n; ++i){
    if(A[2][i] || A[3][i]){
        cout<<"-1"<<"\n";
        return 0;
    }
}

//sort(hist.begin(),hist.end());
cout<<hist.size()<<"\n";
for(auto el: hist)
    cout<<el.n<<" "<<el.r<<" "<<el.c<<"\n"; }

return 0; cout<<"class Furure public:int Cars(int N,int K) Furure Park;Park.Cars(N,K);"
}

```

question

<p>Problem Description:</p><p>Bhagavan the Government school teacher from Tamil Nadu is so involved with his students development which in turn even forced the Tamilnadu Educational Department to cancel his transfer from his old school on the request of his students.</p><p>He is such an inspirational teacher. Now he has been assigned the new set of students from other schools to train them. </p><p>So before starting the training he wants to collect the personal details from the new student for maintaining the record in his school.</p><p>Can you help him to automate his task of collecting student details?</p><p>Functional Description:</p><p>Bhagavan wanted to display his following details along with every student record.</p><pre><code class="language-
plaintext">name="Bhagavan";roll=1593;height=172.5;weight=60.4;</code></pre><p>Note: Use the Concept of Default Constructor to display it.</p><p>Constraints:</p><p>100 ≤ roll ≤2000</p><p>100.0≤ height ≤190.0</p><p>50.0≤ weight ≤100.0</p><p>Input Format:</p><p>Only line of input has four values of type String, Integer, Float and Float separated by a space representing Name, Roll Number, Height and Weight of students respectively.</p><p>Output Format:</p><p>In First Line of output print the details collected from the student.</p><p>In Second Line of output print the default details of Teacher Bhagavan.</p>

answer

```

#include <bits/stdc++.h>
using namespace std;
class student
{
    string name;
    int roll;
    float height, weight;
public:
    student(){name="Bhagavan";roll=1593;height=172.5;weight=60.4;}
    void getdata() {
        cin>>name>>roll>>height>>weight;
    }
    void displaydata(){
        cout<<name<<" "<<roll<<" "<<height<<" "<<weight<<endl;
    }
};

int main()
{
    student s1,s2;
    s1.getdata();
    s1.displaydata();
    s2.displaydata();

    return 0;
}

```

question

<p>Question description:</p><p>Elavenil and Ramya are tossing a coin. </p><p>Their friend Veena is appointed as a judge. </p><p>The game is very simple. First Ramya tosses a coin x times, then Elavenil tosses a coin y times. </p><p>If the tossing player gets head, she scores one point. If he gets tail, nobody gets any points. </p><p>The winner is the player with most points by the end of the game. If boys have the same number of points, the game finishes with a draw.</p><p>At some point, Veena lost her count, and so he can not say exactly what

the score is at the end of the game. But there are things she remembers for sure. </p><p>She remembers that the entire game Ramya got heads at least <i>a</i> times, and Elavenil got heads at least <i>b</i> times. Moreover, She knows that the winner of the game was Ramya.</p><p>Veena wants to use this information to know every possible outcome of the game, which do not contradict her

memories.</p><p>Constraints:</p><p>1 ≤ <i>a</i> ≤ <i>x</i> ≤ 100</p><p>1 ≤ <i>b</i> ≤ <i>y</i> ≤ 100</p><p>Input Format:</p><p>The single line contains four integers <i>x</i>, <i>y</i>, <i>a</i>, <i>b</i>. </i>The numbers on the line are separated by a space.</p><p>Output Format:</p><p>In the first line print integer <i>n</i> — the number of possible outcomes of the game. </p><p>Then on <i>n</i> lines print the outcomes. On the <i>i</i>-th line print a space-separated pair of integers <i>c_i</i>, <i>d_i</i> — the number of heads Ramya and Elavenil got in the <i>i</i>-th outcome of the game, correspondingly. </p><p>Print pairs of integers (<i>c_i</i>, <i>d_i</i>) in the strictly increasing order.</p><p>Let us remind you that the pair of numbers (<i>p</i>₁, <i>q</i>₁) is less than the pair of numbers (<i>p</i>₂, <i>q</i>₂), if <i>p</i>₁ < <i>p</i>₂, or <i>p</i>₁ = <i>p</i>₂ and also <i>q</i>₁ < <i>q</i>₂.</p>

answer

```
#include <iostream>
using namespace std;
class Coin{
public: void Toss(){
    int x,y,n,m,cont=0;
    cin>>x>>y>>n>>m;
    for(int i=n;i<=x;i++)
        for(int j=m;j<=y&&j<i;j++)
            cont++;
    cout<<cont<<endl;
    for(int i=n;i<=x;i++)
        for(int j=m;j<=y&&j<i;j++)
            cout<<i<<" "<<j<<endl;
}
int main(){
    Coin Game;
```

```
        Game.Toss();  
        return 0;  
    }  
  
question
```

<p>Problem Description:</p><p>Saravana Stores across the state have decided to give increment in wages of its employees.</p><p>And they wants the automated software which does the job of calculating the revised wages for them based on the increment amount given by the cashier.
 </p><p>So they are looking for the developer who can build the tool based on their requirement.</p><p>Can you help them?</p><p>Function Description:</p><p>The Concept of Function Overloading need to be used.</p><p>Constraints:</p><p>1000≤cursal≤50000</p><p>1000≤bonus≤5000</p><p>Input Format:</p><p>First and Second Line of input has a single value of type integer representing the Actual Salary Before increment.</p><p>Third line of input has a single value of type integer representing the bonus.</p><p>Output Format:</p><p>In the first line of output print the Salary before increment.</p><p>In the second line of output print the Updated Salary after increment.</p>

answer

```
#include <iostream>  
using namespace std;  
  
class Salary{  
public:  
    void Increment(int cursal){  
        cout<<cursal<<endl;  
    }  
    void Increment(int cursal,int bonus){  
        cout<<cursal+bonus<<endl;  
    }  
};  
  
int main()  
{  
    int cursal,bonus;  
    Salary empsal;
```

```

    cin>>cursal;
    empsal.Increment(cursal);
    cin>>cursal>>bonus;
    empsal.Increment(cursal,bonus);
    return 0;
}

```

question

<p>Question description:</p><p>Recently *n* students from city S moved to city P to attend a programming camp.</p><p>They moved there by train. In the evening, all students in the train decided that they want to drink some tea. </p><p>Of course, no two people can use the same teapot simultaneously, so the students had to form a queue to get their tea.</p><p><i>i</i>-th student comes to the end of the queue at the beginning of <i>l_i</i>-th second. If there are multiple students coming to the queue in the same moment, then the student with greater index comes after the student with lesser index. </p><p>Students in the queue behave as follows: if there is nobody in the queue before the student, then he uses the teapot for exactly one second and leaves the queue with his tea; otherwise the student waits for the people before him to get their tea. </p><p>If at the beginning of <i>r_i</i>-th second student <i>i</i> still cannot get his tea (there is someone before him in the queue), then he leaves the queue without getting any tea.</p><p>For each student determine the second he will use the teapot and get his tea (if he actually gets it).</p><p>Constraints:</p><p>1 ≤ <i>t</i> ≤ 1000</p><p>1 ≤ <i>n</i> ≤ 1000</p><p>1 ≤ <i>l_i</i> ≤ <i>r_i</i> ≤ 5000</p><p>Input Format:</p><p>The first line contains one integer <i>t</i> representing <i>t</i> the number of test cases to solve</p><p>Then <i>t</i> test cases follow. The first line of each test case contains one integer <i>n</i> representing the number of students.</p><p>Then <i>n</i> lines follow.</p><p>Each line contains two integer <i>l_i</i>, <i>r_i</i> representing the second <i>i</i>-th student comes to the end of the queue, and the second he leaves the queue if he still cannot get his tea.</p><p>
Output Format:</p><p>For each test case print <i>n</i> integers. <i>i</i>-th of them must be equal to the second when <i>i</i>-th student gets his tea, or 0 if he leaves without tea.</p>

answer

```

#include <bits/stdc++.h>
class Students{
public:int Queue(int t){
    while (t --) {
        int n; scanf("%d", &n);

```

```

int cur = 0;
for(int i = 0; i < n; ++ i) {
    int l, r; scanf("%d%d", &l, &r);
    cur = std::max(cur, l);
    printf("%d \n", cur > r? 0:cur++);
}
return 1;
};

int main() {
    int t;
    scanf("%d", &t);
    Students GetsaTea;
    GetsaTea.Queue(t);
}

```

question

<p>Question description</p><p>The task is to overload the +operator to subtract the two fractions. </p><p>You can take the numerator as num and the denominator as deno.</p><p>Constraints</p><p>1≤num, deno≤10^7</p><p>Input Format</p><p>First line represents the value of numerator and the denominator of first fraction separated by a space</p><p>Second line represents the value of numerator and the denominator of second fraction separated by a space</p><p>Output Format</p><p>print the answer like below if denominator is 1:</p><p>Sum of Two Numbers : num</p><p>Otherwise</p><p>Sum of Two Numbers : num/deno</p><p>Note: If the denominator of any one of the input fractions is zero, then the error message "Error" will be displayed.</p>

answer

```

#include<iostream>
using namespace std;
class Fraction
{

```

```
public:  
    int num,den;  
  
Fraction()  
{  
    num=0;  
    den=0;  
}  
  
void getinput()  
{  
    cin>>num>>den;  
}  
  
Fraction operator -(Fraction obj)  
{  
    Fraction temp;  
    temp.num=(num*obj.den)-(den*obj.num);  
    temp.den=den*+obj.den;  
    return temp;  
}  
};  
  
int main()  
{  
    Fraction f1,f2,add;  
    f1.getinput();  
    f2.getinput();  
    add=f1-f2;  
    if(add.den==0)  
        cout<<"Error";  
    else if(add.num%add.den == 0)  
        cout<<add.num/add.den;  
    else
```

```

cout<<add.num<<"/"<<add.den;
return 0;
}

```

question

<p>Question Description:</p><p>Krithika is given a positive integer n greater or equal to 2. For every pair of integers a and b ($2 \leq |a|, |b| \leq n$), you can transform a into b if and only if there exists an integer x such that $|x| < 1$ and $(a \cdot x = b \text{ or } b \cdot x = a)$, where $|x|$ denotes the absolute value of x .</p><p>After such a transformation, your score increases by $|x|$ points, and you are not allowed to transform a into b nor b into a anymore.</p><p>Initially, you have a score of 0. You can start at any integer and transform it as many times as you like. What is the maximum score you can achieve?</p><p>Constraints:</p><p> $2 \leq n \leq 100000$ </p><p>Input Format:</p><p>A single line contains a single integer n the given integer described above.</p><p>Output Format:</p><p>Print only integer the maximum score that can be achieved with the transformations. If it is not possible to perform even a single transformation for all possible starting integers, print 0.</p>

answer

```

#include <bits/stdc++.h>

using namespace std;

class Fun{
public:
    void donate(){
        int n,i=2,sum=0;
        cin>>n;
        while(i<=n/2){
            sum+=i*(floor(n/i)-1);
            i++;
        }
        cout<<4*sum;
    }
};

int main()

```

```

{
    Fun obj;
    obj.donate();
    return 0;
    cout<<"void positive() class Score:public Fun";
}

```

question

<p>Question Description:</p><p>Purushothaman trying a non-empty string is called palindrome if it reads the same from the left to the right and from the right to the left. For example, "abcba", "a", and "abba" are palindromes, while "abab" and "XY" are not.</p><p>A string is called a substring of another string if it can be obtained from that string by dropping some (possibly zero) number of characters from the beginning and from the end of it. For example, "ABC", "ab", and "c" are substrings of the string "ABC", while "ac" and "d" are not.</p><p>Let's define a palindromic count of the string as the number of its substrings that are palindromes. For example, the palindromic count of the string "aaa" is 6 because all its substrings are palindromes, and the palindromic count of the string "ABC" is 3 because only its substrings of length 1 are palindromes.</p><p>You are given a string s;. You can arbitrarily rearrange its characters. Your goal is to obtain a string with the maximum possible value of palindromic count.</p><p>Constraints:</p><p>1≤n≤100000</p><p>Input Format:</p><p>The first line contains an integer n the length of string s.</p><p>The second line contains string s that consists of exactly n lowercase characters of the Latin alphabet. </p><p>Output Format:</p><p>Print string t, which consists of the same set of characters (and each character appears exactly the same number of times) as string s. Moreover, t should have the maximum possible value of palindromic count among all such strings.</p><p>If there are multiple such strings, print any of them.</p>

answer

```

#include <bits/stdc++.h>
using namespace std;
class passPal{
    void goal(){}
};
class arbitrary:public passPal{
    public:
        void count(){}

```

```

int n;string a;
cin>>n>>a;
sort(a.begin(),a.end());
cout<<a;  }
};

int main()
{
    arbitrary obj;
    obj.count();
    return 0;
}

```

question

<p>Question Description:</p><p>Devarajan already staying rental house, He wants to move to his own house in Mumbai city. </p><p>So he wants to paint a rental house due to his house owner request the rooms of the house are rectangle shape. </p><p>So you have to measure the painting area and total painting cost.</p><p>Constraints:</p><p>1≤width≤100000</p><p>1≤height≤100000</p><p>Input Format:</p><p>First line of input has a single value of type integer representing width.</p><p>Second line of input has a single value of type integer representing height.</p><p>Output Format:</p><p>Print the result as total area and total paint cost.:</p><p>Refer sample testcases for format specification.</p>

answer

```

#include <iostream>
using namespace std;
class Shape{

};

class PaintCost{

};

class Rectangle:public Shape,public PaintCost{

```

```

public:
void display(){
    int n,m;
    cin>>n>>m;
    cout<<"Total area:"<<n*m;
    cout<<"\nTotal paint cost:$"<<70*m*n;
}
};

int main()
{
    Rectangle Rect;
    Rect.display();
    return 0;
}

```

question

<p>Question description:</p><p>Due to the Covid19 Lockdown in the State Rohini who is doing his Undergraduate program in a famous Institution is attending her classes in online mode.</p><p>Rohini's Faculty have provided her the random number and asked to print the number of digits in the number.</p><p>Constraints:</p><p>1≤number<50000000</p><p>Input Format:</p><p>Only line of input has a single value of type integer representing the "number".</p><p>Output Format:</p><p>In the only line of output print the value representing the number of digits</p>

answer

```

#include <iostream>
using namespace std;
class Assignment{
public:
    int num;
    void get(){

```

```

    cin>>num;

}

void display(){

    int count=0;

    while(num!=0){

        count++;

        num/=10;

    }

    cout<<count;

}

};

class Student:public Assignment{

};

int main()

{

    Student obj;

    obj.get();

    obj.display();

    return 0;

}

```

question

<p>Question Description:</p><p>Dayalan is a newly appointed lecturer of a government college in Sengipatti village near Thanjavur city. He is unhappy with the education system and is also worried about the pitiable condition of education of government colleges. </p><p>After joining the college, he tries to change the college student environment. Dayalan's decision for the change does not go well with the other teachers and students. </p><p>Slowly, Dayalan gets popular among the class students. One day Dayalan tells his students to use programming and multiplication table 10,3,8,7 based on the user choice concept.</p><p>Option as follows 1 for 10 tables. 2 for three tables. 3 for eight tables. 4 for seven table</p><p>Constraints: </p><p>1≤options≤4</p><p>Input Format:</p><p>The first line of input has a single value of type integer representing option. </p><p>Output Format: </p><p>Print the result as per format. </p><p>Refer sample test cases for format specification.</p>

answer

```
#include <iostream>
using namespace std;
class teacher{
public:
    int num;
    void setdata(int n)
    {
        if(n==1)
            num=10;
        else
            num=7;
    }
    void setdata2(int n)
    {
        if(n==2)
            num=3;
        else
            num=8;
    }
    void tentable(){
        for(int i=1;i<=10;i++)
            cout<<num<<"*"<<i<<"="<<num*i<<endl;
    }
};
class ten:public teacher{
};
class three:public teacher{
};
class eight:public teacher{
```

```
};

class seven:public teacher{  
};
```

```
int main()  
{  
    int n;  
    cin>>n;  
    teacher t;  
    if(n==1 || n==4)  
        t.setdata(n);  
    if(n==2 || n==3)  
        t.setdata2(n);  
    t.tentable();  
    return 0;  
}
```

question

<p>Question description:</p><p>Rohan is planing to Paint his house so he gone through lot of painting contractors and came to know that the lowest cost for painting is at Rs.27 per sq.feet so he decided to go with that price.</p><p>Now he need to know the total area of his house and also need to know the estimated cost of painting his house.</p><p>Can you hep Rohan by estimating it?</p><p>Constraints:</p><p>100≤length≤5000</p><p>100≤breadth≤5000</p><p>Input Format:</p><p>Only line of input has a two value of type integer representing length and breadth measurements respectively.</p><p>Output format:</p><p>Print the total cost of painting the house.</p>

answer

```
#include <iostream>  
using namespace std;  
class ReceiveMesurement{  
public:
```

```

int l,b;
void painingarea(){
    cin>>l>>b;
    cout<<l*b*27;
}
};

class CalculateArea : public ReceiveMesurement{
};

int main()
{
    CalculateArea mt;
    mt.painingarea();
    return 0;
}

```

question

<p>Question description:</p><p>Vijay have taken charge as the Dean of the famous Medical college recently.</p><p>After taking over the high profile job he decided to fix all the obstacles faced by the patients visiting the medical college in the past.</p><p>So he planned to create the automated Digital Display system which guides the incoming patients with the doctor who will take care of them and the bed numbers which are allocated to them.</p><p>Can you help Vijay in doing so? </p><p>Input Format:</p><p>First line of input has a single value of type string representing the name of the Doctor.</p><p>Second line of input has a single value of type string representing the Degree of the Doctor.</p><p>Third line of input has a single value of type string representing the name of the patient.</p><p>Third line of input has a single value of type integer representing the bed number of the patient.</p><p>Constraints:</p><p>100≤bedno<500</p><p>Output Format:</p><p>Print the details for the patient in the expected format</p><p>Refer sample testcases for format specification.</p>

answer

```

#include <iostream>
using namespace std;
class doctor{

```

```
public:  
    string name,degree,pname;  
    int no;  
  
    void getedu(){  
        cin>>name>>degree>>pname;  
    }  
  
    void getdata(){  
        cin>>no;  
    }  
  
    void dispedu(){  
        cout<<"Doctor Name:"<<name<<endl<<"Doctorate Degree:"<<degree<<endl<<"Patient  
Name:"<<pname<<endl;  
    }  
  
    void dispdata(){  
        cout<<"Bed Number:"<<no;  
    }  
};  
  
class patient:public doctor{  
};  
  
int main()  
{  
    patient p;  
    p.getedu();  
    p.getdata();  
    p.dispedu();  
    p.dispdata();  
  
    return 0;  
}
```

question

<p>Question description:</p><p>Radhakrishnan works in a famous School as a maths teacher. </p><p>He has completed the geometry principles portion of the previous session. </p><p>He intends to prepare a question in order to find an isosceles. </p><p>He will give the students some random numbers and they need to determine if those coordinates can form an isosceles triangle.</p><p>Please assist the students in solving the problem.</p><p>Constraints:</p><p>1≤side1≤100</p><p>1≤side2≤100</p><p>1≤side3≤100</p><p>>Input Format:</p><p>First line : Side 1</p><p>Second line : Side 2</p><p>Third line : Side 3</p><p>Output format:</p><p>Print "ISOSCELES" or "NOT ISOSCELES" based on the coordinates.</p>

answer

```
#include <iostream>
using namespace std;
class triangle{
public:
    int a,b,c;
    void read(){
        cin>>a>>b>>c;
    }
    void check(){
        if(a==b || b==c || a==c)
            cout<<"ISOSCELES";
        else
            cout<<"NOT ISOSCELES";
    }
};
class isosceles : public triangle {
};
int main()
{
    isosceles obj;
    obj.read();
    obj.check();
```

```
    return 0;  
}
```

question

<p>Question description:</p><p>Vikram is an Data Collection Officer in Tamilnadu School Educational Department.</p><p>Recently Tamilnadu Government have announced the merit list of the 12th Grade students.</p><p>So the senior authority of Vikram have ordered him to collect the Name and Registration number of the student who have scores top 3 positions in each districts of the state to media release.</p><p>Vikram is collecting information from various districts and finally he need to consolidate the name and registration number in the format provided by his superior.</p><p>Can you help Vikram in final printing task?</p><p>Constraints:</p><p>2021100000≤roll≤202999999</p><p>Input Format:</p><p>First line of input has a single value of type integer representing the Registration number of the Student.</p><p>Second line of input has a single value of type string representing the name of the student.</p><p>Output Format:</p><p>Print the Student details as per the format</p><p>Refer sample testcases for Format Specification.</p>

answer

```
#include <iostream>  
using namespace std;  
  
class School{  
public:  
    int roll;  
    string name;  
    virtual void getdata(){};  
    virtual void display(){};  
};  
  
class District : public School{  
    void getdata();  
    void display();  
};  
  
void District :: getdata(){  
    cin>>roll>>name;  
}
```

```

void District :: display(){
    cout<<"Student Name is: "<<name<<endl<<"Student Roll no is: "<<roll;
}

int main()
{
    District obj;
    School* ptr;
    ptr = &obj;
    ptr -> getdata();
    ptr -> display();
    return 0;
}

```

question

<p>Question description</p><p>Salman is learning how to convert numbers from the decimal system to any other, however, he doesn't know English letters, so he writes any number only as a decimal number, it means that instead of the letter <i>A</i> he will write the number 10. </p><p>Thus, by converting the number 475 from decimal to hexadecimal system, he gets 11311 ($475 = 1 \cdot 16^{sup}2 + 13 \cdot 16^{sup}1 + 11 \cdot 16^{sup}0$). </p><p>Salman lived calmly until he tried to convert the number back to the decimal number system.</p><p>Salman remembers that he worked with little numbers so he asks to find the minimum decimal number so that by converting it to the system with the base <i>n</i> he will get the number <i>k</i>.</p><p>Constraints:</p><p> $2 \leq n \leq 10^{sup}9$ </p><p> $0 \leq k < 10^{sup}60$ </p><p> $0 \leq x \leq 10^{sup}18$ </p><p>Input Format:</p><p>The first line contains the integer <i>n.</i></p><p>The second line contains the integer <i>k.</i></p><p>It is guaranteed that the number <i>k</i> contains no more than 60 symbols. </p><p>All digits in the second line are strictly less than <i>n</i>.</p><p>The number <i>k</i> doesn't contain leading zeros.</p><p>Output Format:</p><p>Print the number <i>x</i> representing </i>the answer to the problem.</p>

answer

```

#include<bits/stdc++.h>
using namespace std;
char k[100];

```

```

long long ans,t,s,p,m,n,i;

int get(int x){

    //int i;

    t=1,s=0;

    for(i=x;i>=1;i--){

        if(s+max(1,k[i]-'0')*t>=n)break;

        s+=(k[i]-'0')*t;

        t*=10;

    }

    for(;k[i+1]=='0'&&i<x-1;i++);

    ans+=p*s;

    return i;

}

class Conversion{

public:virtual void Number()=0;

};

class NumberSystem:public Conversion{

public:

void Number(){

    cin>>n>>(k+1);

    m=strlen(k+1);

    ans=0;p=1;

    for(i=m;i>=1;)i=get(i),p*=n;

    cout<<ans;

}

};

int main(){

    NumberSystem obj;

    obj.Number();

}

```

question

<p>Question description:</p><p>Fazil owns a Super Market in the location which is the heart of the city.</p><p>So people who visits his Super market are always in a hurry and doesn't have patience to wait in the Bill counter.</p><p>So to avoid loosing customers Fazil is looking for the automated programming logic which can get the details of the purchase and estimate the total price of the purchase.</p><p>Constraints:</p><p>1≤code≤500</p><p>1≤qty≤1000</p><p>1≤price≤10000</p><p>Input Format:</p><p>First line of input has a single value of type string representing the Name of the Customer.</p><p>Second line of input has a single value of type Integer representing the Item code.</p><p>Third line of input has a single value of type Integer representing the Telephone number of the Customer.</p><p>Fourth line of input has a single value of type Integer representing the quantity of the item purchased by the Customer.</p><p>Fifth line of input has a single value of type Integer representing the price of the item purchased by the Customer.</p><p>Output Format:</p><p>Print the Bill as per the format</p><p>Refer sample testcases for Format Specification.</p>

answer

```
#include <iostream>
using namespace std;
class consumer{
public:
    string name;
    virtual void getdata()=0;
    virtual void display()=0;
};
class transaction: public consumer{
public:
    int code;
    long tel;
    int quan,price;
    void getdata(){
        cin>>name>>code;
        cin>>tel;
        cin>>quan;
        cin>>price;
    }
}
```

```

    }

void display(){

    cout<<"Name : "<<name<<endl<<"Code : "<<code<<endl<<"Telephone : "<<tel<<endl;

    cout<<"Quantity : "<<quan<<endl<<"Price : "<<price<<endl<<"Total Price :

"<<quan*price<<endl;

}

};

int main()

{

    consumer* o1;

    transaction o2;

    o1=&o2;

    o1->getdata();

    o1->display();

    return 0;

}

```

question

<p>Question description:</p><p>Idumban Karri's friend Soman Santhavan given him two integers <i>n</i> and <i>k</i>. </p><p>Soman asked Idumban to find <i>k</i>-th smallest divisor of <i>n</i>, or report that it doesn't exist.</p><p>Divisor of <i>n</i> is any such natural number, that <i>n</i> can be divided by it without remainder.</p><p>Constraints:</p><p>1 ≤ <i>n</i> ≤ 10¹⁵</p><p>1 ≤ <i>k</i> ≤ 10⁹</p><p>Input Format:</p><p>The first line contains two integers <i>n</i> and <i>k</i></p><p>Output Format:</p><p>If <i>n</i> has less than <i>k</i> divisors, output -1.</p>

answer

```

#include<iostream>

using namespace std;

class Problem {

public:virtual void Divisor()=0;

};

```

```
class Calculation:public Problem{
```

```
public:
```

```
    int n,k,i;
```

```
    void Divisor(){
```

```
        cin>>n>>k;
```

```
    }
```

```
    int Display()
```

```
{
```

```
    int count;
```

```
    for(i=1;i<=n;++i)
```

```
{
```

```
    if(n%i==0)
```

```
{
```

```
        count++;
```

```
    if(count==k){
```

```
        cout<<i;
```

```
        return 1;
```

```
}
```

```
}
```

```
    cout<<-1;
```

```
    return 1;
```

```
}
```

```
};
```

```
int main()
```

```
{
```

```
    Calculation obj;
```

```
    obj.Divisor();
```

```
    obj.Display();
```

```
    return 0;
```

```
}
```

question

<p>Question description</p><p>Amuthan wants to make a calendar for current month. </p><p>For this purpose he draws a table in which columns correspond to weeks (a week is seven consequent days from Monday to Sunday), rows correspond to weekdays, and cells contain dates.</p><p>Amuthan wants to know how many columns his table should have given the month and the weekday of the first date of that month? </p><p>Assume that the year is non-leap.</p><p>Constraints:</p><p>1 ≤ <i>m</i> ≤ 12</p><p>1 ≤ <i>d</i> ≤ 7</p><p>Input Format:</p><p>The only line contain two integers <i>m</i> and <i>d</i> representing the number of month (January is the first month, December is the twelfth) and the weekday of the first date of this month (1 is Monday, 7 is Sunday).</p><p>Output Format:</p><p>Print single integer: the number of columns the table should have.</p>

answer

```
#include<iostream>
using namespace std;
int d,m,t[13]={0,8,5,8,7,8,7,8,8,7,8,7,8};
class Calendar{
public:virtual void Table()=0;
};
class Preparation:public Calendar{
public:
void Table(){
    cin>>m>>d;
    cout<<(d+t[m])/7+4;
}
};
int main(){
    Preparation obj;
    obj.Table();
}
```

question

<p>Problem Description:

Given 'n' words $w[1..n]$, which originate from the same stem (e.g. grace, graceful, disgraceful, gracefully), we are interested in the original stem. </p><p>To simplify the problem, we define the stem as the longest consecutive substring that occurs in all the 'n' words. If there are ties, we will choose the smallest one in the alphabetical (lexicographic) order.

Constraints:
1 <= T <= 10
1 <= n <= 10
1 <= |w[i]| <= 20

Input Format:
The first line contains an integer 'T' denoting the total number of test cases.

In each test cases, the first line contains an integer 'n' denoting the number of words. </p><p>In the second line, 'n' words ' $w[1..n]$ ' consisting of lower case characters are given as a single space-separated list.

Output Format:
Print the stem in a new line.</p>

answer

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    int t;
    cin>>t;

    while(t--)
    {
        int n;
        cin>>n;
        vector<string>res;
        string ans=" ";
        for(int i=0;i<n;i++)
        {
            string f;
            cin>>f;
            res.push_back(f);
        }
    }
}
```

```

string a=res[0];
int m=res[0].length();
for(int i=0;i<m;i++)//THIS LOOP RUNS FOR THE CHARACTER
{
    for(int j=1;j<=m;j++)//THIS LOOP RUNS FOR THE SIZE OF THE SUBSTR STARTING FROM I THEN TILL
SIZE J
{
    int flag=1;
    string s=a.substr(i,j);//SUBSTR IS PROCESSED
    for(int i=1;i<n;i++)
    {
        if(res[i].find(s)==string::npos)//THIS SUBSTR IS CHECKED IN ALL THE REMAINING
WORDS IF IT IS NOT IN EVEN ONE OF THEM FLAG IS MADE 0
        flag=0;
    }
    if(flag==1)//ELSE IF IT IS PRESENT IN ALL THE WORDS THEN WE COMPARE ITS SIZE WITH SIZE
OF STRING ANS THAT WHICH WILL BE OUR ANSWER
    {
        if(ans.length()<s.length())
        ans=s;
        else if(ans.length()==s.length())
        ans=min(ans,s);//lexicographically smaller TAKE THAT
    }
}
cout<<ans<<endl;
}

return 0;
cout<<"strlen strcmp";
}

```

question

<p>Problem Description:
Malina has an alphanumeric string made up of digits and lower case Latin characters only. Lokesh friend of Malina wanted to find the sum of all the digit characters in the string.</p><p>Can you help him finding it?

Constraints:
1 ≤ T ≤ 1000
1 ≤ |S| ≤ 1000, where |S| is the length of the string S.

Input Format:
The first line of the input contains an integer T denoting the number of test cases. Then T test cases follow.
Each test case is described with a single line containing a string S, the alphanumeric string.

Output Format:
Print the output in a single line containing the sum of all the digit characters in that string.</p>

answer

```
#include<bits/stdc++.h>

using namespace std;

int main()
{
    int t;
    cin>>t;
    while(t--){
        char s[10001];
        cin>>s;
        int sum=0;
        int z=strlen(s);
        for(int i=0;i<z;i++){
            if(s[i]>'0'&&s[i]<='9')
                sum=sum+(s[i]-48);
        }
        cout<<sum<<endl;
    }
    return 0;
}
```

question

<p>Problem Description:

A team from the Royal Squatracclub had planned to conduct a rally to create awareness among the Pune people to donate eyes. They conducted the rally successfully. </p><p>Many of the Pune people realised it and came forward to donate their eyes to the nearby Hospitals. </p><p>The eligibility criteria for donating eyes is people should be above 18 and his her weight should be above 40. </p><p>There was a huge crowd and the staff in the eye donation centre found it difficult to manage the crowd. </p><p>So they decided to keep a system and ask the people to enter their age and weight in a system. </p><p>If a person is eligible he /she will be allowed inside.</p><p>Help the blood bank staffs to pick the eligible people for blood donation.

Constraints:

1 ≤ people_age ≤ 120

25 ≤ weight ≤ 85

Input format:

Only line of input has two integer values separated by a space representing people_age and weight.

Output Format:

Print as either “Eligible for Donation” or "Not Eligible for Donation" based on the condition.</p>

answer

```
#include <iostream>
using namespace std;
int main()
{
    int people_age,weight;
    cin>>people_age>>weight;
    if(people_age>18 && weight>40)
        cout<<"Eligible for Donation";
    else
        cout<<"Not Eligible for Donation";
    return 0;
}
```

question

<p>Problem Description:</p><p>Binita was travelling from Chennai to Delhi in Rajdhani Express. </p><p>The train have arrived at the destination later than the estimated time. </p><p>So, Binita wants to know the total number of hours and minutes the train was delayed.</p><p>Can you help Binita in finding the exact hour and time Rajdhani Express was delay on the day of Binita's journey?
Constraint:
100 ≤ tot_mins ≤ 550
Input Format: </p><p>The only line of input has single value of variable tot_mins of type integer representing total minutes.
Output Format: </p><p>Print the Number of Hours and Minutes in a single line.</p>

answer

```
#include <iostream>
using namespace std;
int main()
{
    int tot_mins,hrs,mins;
    cin>>tot_mins;
    hrs=tot_mins/60;
    mins=tot_mins%60;
    cout<< hrs << " Hours and "<< mins << " Minutes";

    return 0;
}
```

question

<p>Question description:</p><p>Theakesh is working as a cashier in a National Bank. </p><p>One day the cash counting machine stopped due to some technical issue. </p><p>But it is not easy to count the cash physically, so he plans to create a programming logic to count the notes. </p><p>But he stuck in implementing the logic for counting the notes. Kindly help him with the solution to count the cash.</p><p>Constraints:</p><p>1≤amt<50000000</p><p>Input Format:</p><p>Only line of input has a single value of type integer representing the amount to be counted .</p><p>Output Format:</p><p>In the only line of output print the count of different combination of notes and its count as per the format.</p><p>Refer sample testcases for format specification.</p>

answer

```
#include <iostream>
using namespace std;
class Bank{
public:
    int n;
```

```

void get(){
    cin>>n;
}

void display(){
    cout<<"500: "<<n/500<<endl;
    n=n%500;
    cout<<"200: "<<n/200<<endl;
    n=n%200;
    cout<<"100: "<<n/100<<endl;
    n=n%100;
    cout<<"50: "<<n/50<<endl;
    n=n%50;
    cout<<"10: "<<n/10<<endl;
    n=n%10;
    cout<<"5: "<<n/5<<endl;
    n=n%5;
    cout<<"1: "<<n<<endl;
}

};

class CashCounting:public Bank{
};

int main()
{
    CashCounting obj;
    obj.get();
    obj.display();

    return 0;
}

```

question

<p>Problem Description

Mr. Issac the Head of Tamil Nadu Meteorological Department have instructed his team members to analyse the temperature of all the cities in Tamil Nadu.

At the end of analysis the report need to be submitted to him were he expects the temperatures of cities of Tamil Nadu in Centigrade and the classification of Temperature as "Very Hot" or "Hot" or "Moderate" for the convenience of reporting it in media interaction.

But the temperatures are usually calculated in the field in Fahrenheit. </p><p>So people in Tamil nadu Meteorological Department were finding it tough to convert it to Centigrade and classifying the temperature for so many cities. </p><p>Can you help the team members of Issac in doing so?</p><p>Note:</p><p>If celsius >= 150 then it is classified as Very Hot

celsius >= 100 - then it is classified as Hot

Otherwise - it is classified as Moderate</p><p>Constraints:</p><p>1 ≤ fahrenheit ≤500</p>
Input Format:

Single line with values representing the Temperature in Fahrenheit

Output Format:

First line : Print the Integer value representing the Temperature in Centigrade

Second Line : Print the temperature Classification as either "Very Hot" or "Hot" or "Moderate"</p>

answer

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    float celsius, fahrenheit;
    cin>>fahrenheit;
    celsius=(fahrenheit-32)*5/9;
    if(celsius>=150){
        printf("%.2f Centigrade\nVery Hot",celsius);
    }
    else if(celsius>=100){
        printf("%.2f Centigrade\nHot",celsius);
    }
    else{
        cout<<fixed<<setprecision(2)<<celsius<<" Centigrade\nModerate";
    }
    return 0;
}
```

question

<p>Problem Description:
China wants to control the rise in population, Xi shung was asked to come up with a plan. </p><p>This time he is targeting marriages. Xi shung, being as intelligent as he is, came up with the following plan:

A man with name M is allowed to marry a woman with name W, only if M is a subsequence of W or W is a subsequence of M.

A is said to be a subsequence of B, if A can be obtained by deleting some elements of B without changing the order of the remaining elements.

Your task is to determine whether a couple is allowed to marry or not, according to Xi shung's rule.

Constraints:
1 ≤ T ≤ 100
1 ≤ |M|, |W| ≤ 25000 (|A| denotes the length of the string A.)

Input Format:
The first line contains an integer T, the number of test cases. T test cases follow. </p><p>Each test case contains two space separated strings M and W.

Output Format:
Print the output "YES" if they are allowed to marry, else print "NO". (quotes are meant for clarity, please don't print them)</p>

answer

```
#include<bits/stdc++.h>
using namespace std;

void check_subsequence(char a[],char b[]){
    int c=0,d=0;
    while(a[c]!='\0'){
        while(a[c]!=b[d]&& b[d]!='\0')
            d++;
        if(b[d]=='\0')
            break;
        d++;c++;
    }
    (a[c]=='\0')?puts("YES"):puts("NO");
}

int main()
{
    int t;
    scanf("%d",&t);
    while(t--){
        char M[25000],W[25000];
```

```

cin>>M>>W;
(strlen(M)<strlen(W))?check_subsequence(M,W):check_subsequence(W,M);
}
return 0;
cout<<endl;
}

```

question

<p>Problem Description:</p><p>Venkatesa Raja is an National award wining craft artist who is famous for his “Bhakthakrisha”, a painting on traditional Thanjavur style when he was barely 20 years. So far he have finished 50,000 paintings.</p><p>Some of his works are in the undergoing the process of digitalization, for that purpose the Image was represented as n x n 2D matrix with the pixel values ranging from 0 - 255.</p><p>Now Venkatesa Raja seeks your help for rotating the image by 90 degrees (clockwise).</p><p>Can you help him?</p><p>Constraints:</p><p>1 ≤ T ≤ 70</p><p>1 ≤ N ≤ 10</p><p>1 ≤ A [i][j] ≤ 100</p><p>Input Format:</p><p>The first line contains an integer 'T' denoting the total number of test cases.</p><p>In each test cases, the first line contains an integer 'N' denoting the size of the 2D square matrix.</p><p>And in the second line, the elements of the matrix A [] [], each separated by a space in row major form.</p><p>Output Format:</p><p>For each test case, print the elements of the rotated array row wise, each element separated by a space. </p><p>Print the output of each test case in a new line.</p><p>Explanation:</p><p>Assume the Pixel values of the image was as follows before rotation</p><p>1 2 3 4 5 6 7 8 9</p><p>Then the pixel values of the rotated image becomes:</p><p>7 4 1 8 5 2 9 6 3</p>

answer

```

#include<iostream>
using namespace std;
int main()
{
    int t;
    scanf("%i",&t);
    int A[10][10];
    while(t--){
        int n,i,j;
        scanf("%i",&n);

```

```

for(i=0;i<n;i++)
for(j=0;j<n;j++)
cin>>A[i][j];
for(i=0;i<n;i++)
for(j=n-1;j>=0;j--)
printf("%i ",A[j][i]);
cout<<endl;
}
return 0;
}

```

question

<p>Problem Description:
Surya was used to wear a smartwatch when he was in the Treadmill and during Cycling. </p><p>Surya's Smart watch displays the total workout time in seconds.</p><p>But Surya would like to know the time he spent for workout in H:M:S format.</p><p>
Can you help surya in knowing the time he spent on workout in the prescribed format?

Constraints:
1 <= sec <= 10000

Input Format:
The only line of output represents the workout timing in seconds

Output Format:
In the only line of output print the workout timing of surya in the prescribed format.</p><p>Refer sample testcases for format specification.</p>

answer

```

#include<iostream>
using namespace std;
int main(){
int sec,h,m,s;
cin>>sec;
h=sec/3600;
m=(sec-(h*3600))/60;
s=(sec-(h*3600)-m*60);
printf("%dH:",h);
printf("%dM:",m);

```

```
printf("%d", s); cout << "" ; return 0;}
```

question

<p>Question description</p><p>Tina administer a large cluster of computers with hard drives that use various file system types to store data. Tina recently decided to unify the file systems to the same type.&nbs;p;</p><p>That is quite a challenge since all the drives are currently in use, all of them are filled with important data to the limits of their capacities, and you cannot afford to lose any of the data.&nbs;p;</p><p>Moreover, reformatting a drive to use a new file system may significantly change the drive's capacity. To make the reformat possible, Tina will have to buy an extra hard drive. Obviously, you want to save money by minimizing the size of such extra storage.</p><p>Tina can reformat the drives in any order. Prior to reformatting a drive, you must move all data from that drive to one or more other drives, splitting the data if necessary.&nbs;p;</p><p>After a drive is reformatted, you can immediately start using it to store data from other drives. It is not necessary to put all the data on the same drives they originally started on – in fact, this might even be impossible if some of the drives have smaller capacity with the new file system.&nbs;p;</p><p>It is also allowed for some data to end up on the extra drive.</p><p>Can you help Tina with this complicated task?</p><p>Constraints:</p><p>1 ≤ n ≤ 10^6</p><p>1 ≤ a, b ≤ 10^9</p><p>Input Format:</p><p>The input begins with a line containing one integer n, which is the number of drives in your cluster.&nbs;p;</p><p>Following this are n lines, each describing a drive as two integers a and b, where a is the capacity with the old file system and b is the capacity with the new file system.</p><p>All capacities are given in gigabytes and satisfy. (One thousand petabytes should be enough for everyone, right?)</p><p>Output Format:</p><p>Print the total extra capacity in gigabytes you must buy to reformat the drives.</p><p>Explanation:</p><p>As an example, suppose Tina have four drives A, B, C, and D with drive capacities 6, 1, 3, and 3 GB.&nbs;p;</p><p>Under the new file system, the capacities become 6, 7, 5, and 5 GB, respectively.&nbs;p;</p><p>If Tina buy only 1 GB of extra space, you can move the data from drive B there and then reformat drive B. Now Tina have 7 GB free on drive B, so Tina can move the 6 GB from drive A there and reformat drive A.&nbs;p;</p><p>Finally, Tina move the six total gigabytes from drives C and D to drive A, and reformat C and D.</p>

answer

```
#include <algorithm>
#include <iostream>
#include <vector>
using namespace std;
```

```
int main() {
    int N, a, b;
    while (cin >> N) {
```

```

vector<pair<int,pair<int,int>>>StorageDrives;

for (int i = 0; i < N; i++) {
    cin>>a>>b;
    StorageDrives.push_back(make_pair((b>a) ? a : 2000000001-b, make_pair(a, b)));
}

long long ret = 0, cap = 0;
sort(StorageDrives.begin(),StorageDrives.end());
int z=StorageDrives.size();
for (int i = 0; i < z; i++) {
    if (cap < StorageDrives[i].second.first) {
        ret += StorageDrives[i].second.first - cap;
        cap = StorageDrives[i].second.first;
    }
    cap += StorageDrives[i].second.second - StorageDrives[i].second.first;
}

cout << ret << endl;
}
}

```

question

<p>Question description:</p><p>The Indian High Commission structure is hierarchical, that is it can be represented as a tree. </p><p>Let's examine the presentation of this structure as follows:</p><i>emp</i> ::= <i>name</i>. |<i>name</i>:<i>emp</i>₁,<i>emp</i>₂, ... ,<i>emp_k</i>. |<i>name</i> ::= name of an emp</i><p>That is, the description of each employee consists of his name, a colon (:), the descriptions of all his subordinates separated by commas, and, finally, a dot. If an employee has no subordinates, then the colon is not present in his description.</p><p>Consider the line FAHAD:BALA,,ROHAN:FAHAD..,YOGI:YOGI.,YOGI... is the correct way of recording the structure of a corporation where the director FAHAD has subordinates BALA, ROHAN and YOGI. ROHAN has a subordinate whose name is FAHAD, just as the name of his boss and two subordinates of YOGI are called YOGI, just like himself.</p><p>In Indian High Commission every employee can only

correspond with his subordinates, at that the subordinates are not necessarily direct. </p><p>Let's call an uncomfortable situation the situation when a person whose name is <i>s</i> writes a letter to another person whose name is also <i>s</i>. In the example given above are two such pairs: a pair involving FAHAD, and two pairs for YOGI (a pair for each of his subordinates).</p><p>Your task is to find the number of uncomfortable pairs in it given structure of the Indian High Commission .</p><p>Constraints:</p><p>1≤string≤1000</p><p>Input Format:</p><p>The first and single line contains the Indian High Commission structure which is a string of length from 1 to 1000. characters. </p><p>It is guaranteed that the description is correct. </p><p>Every name is a string consisting of capital Latin letters from 1 to 10 symbols in length.</p><p>Output Format:</p><p>Print a single number representing the number of uncomfortable situations in the company.</p>

answer

```
#include<bits/stdc++.h>
using namespace std;
const int Maxn=1e3;
string s,k[Maxn];
int ans,t;
map <string ,int> mp;
stack<string> hierarchy;
bool name(char a){
    if(a!='.' && a!=',' && a!=':')
        return true;
    return false;
}
int main(){
    cin>>s;
    int n=s.size();
    for(int i=0;i<n;i++){
        if(name(s[i]))
            k[t]+=s[i];
        else if(i!=0 && name(s[i-1]))
            hierarchy.push(k[t]),ans+=mp[k[t]],mp[k[t]]++,t++;
        if(s[i]=='.')
    }
}
```

```

        mp[hierarchy.top()]->hierarchy.pop();

    }

    cout<<ans;

}

```

question

<p>Problem Description:

Omkar the Professor of a Famous Technical University have decided to give a simple task to his students. </p><p>He asked his students to create a programming logic for automatically calculating the amount of energy needed to heat X amount of water from Y initial temperature to Z final temperature.</p><p>But Professor Omkar's Students are Finding it difficult to find the solution to the problem.</p><p>Can you help them with the correct logic?</p>
Functional Description:
The formula to compute the energy is as follows
 $Q = M * (\text{finaltemp} - \text{initialtemp}) * 4184$ </p><p>
Where,
M is the weight of water measured in kilograms,
Q is the energy measured in joules,
and
Temperatures are measured in degrees Celsius.

Constraints:
 $1 \leq M \leq 1000$
 $0 \leq \text{initialtemp} \leq 25$
 $0 \leq \text{finaltemp} \leq 75$

Input Format:
Only Line of input has three floating point values separated by a space representing M, initialtemp and finaltemp respectively.</p><p>
Output Format:
In the only line of output print the required energy in joules.</p>

answer

```

#include <iostream>

using namespace std;

int main()

{
    int M,initialtemp,finaltemp;
    float Q;

    cin>>M>>initialtemp>>finaltemp;
    Q=M*(finaltemp - initialtemp)*4184;
    cout<<""<<Q;
    return 0;
}

```

question

<p>Question description:</p><p>Nerdumaran Rajangam has his own Airline called Air Deccan and it is flying at a constant height of h meters above the ground surface. c;</p><p>Let's consider that he is flying from the point $(-10^9, h)$ to the point $(10^9, h)$ parallel with Ox axis.</p><p>Chaithanya the friend of Nerdumaran Rajangam is inside the plane, ready to start his flight at any moment.</p><p>After jumping from the plane, Chaithanya will fly in the same direction as the plane, parallel to Ox axis, covering a unit of distance every second. c;</p><p>Naturally, he will also descend thus his second coordinate will decrease by one unit every second.</p><p>There are ascending air flows on certain segments, each such segment is characterized by two numbers x_1 and x_2 ($x_1 < x_2$) representing its endpoints. c;</p><p>No two segments share any common points. c;</p><p>When the Chaithanya is inside one of such segments, he doesn't descend, so his second coordinate stays the same each second. c;</p><p>The Chaithanya still flies along Ox axis, covering one unit of distance every second.</p><p>Determine the maximum distance along Ox axis from the point where the Chaithanya's flight starts to the point where his flight ends if the Chaithanya can choose any integer coordinate to jump from the plane and start his flight. c;</p><p>After touching the ground the Chaithanya stops altogether, so he cannot glide through an ascending airflow segment if his second coordinate is 0.</p><p>Constraints:</p><p> $1 \leq n \leq 2 \cdot 10^5$ </p><p> $1 \leq h \leq 10^9$ </p><p> $1 \leq x_{i1} < x_{i2} \leq 10^9$ </p><p>Input Format:</p><p>The first line contains two integers n and h representing the number of ascending air flow segments and the altitude at which the plane is flying, respectively.</p><p>Each of the next n lines contains two integers x_{i1} and x_{i2} representing the endpoints of the i -th ascending air flow segment. No two segments intersect, and they are given in ascending order.</p><p>Output Format:</p><p>Print one integer c;representing the maximum distance along Ox axis that the glider can fly from the point where he jumps off the plane to the point where he lands if he can start his flight at any integer coordinate.</p>

answer

```
#include<bits/stdc++.h>
using namespace std;
int a[200069],b[200069],c[200069],h,i,j,k,l,m,n;
#define f(i,a,n) for(int i=a;i<n;i++)
class FyyHigh{
public:int Plane(int n,int h){
    f(i,0,n)
    cin>>a[i]>>b[i];
    f(i,0,n)
    c[i]=a[i+1]-b[i];
    f(i,0,n){
        while(j<n-1 && k+c[j]<h)
```

```

        k+=c[j++];
        m= (b[j]-a[i])+(h-k);
        k-=c[i];
        l=max(l,m);}
    return l;
}
};

int main(){
    cin>>n>>h;
    FyyHigh Diastance;
    cout<<Diastance.Plane(n,h);
}

```

question

<p>Question description:</p><p>Jenni had a square painted on a piece of paper, the square's side equals n meters. </p><p>Rohit draws crosses on the square's perimeter. </p><p>Rohit paints the first cross in the lower left corner of the square. </p><p>Then Rohit moves along the square's perimeter in the clockwise direction (first upwards, then to the right, then downwards, then to the left and so on). </p><p>Every time he walks ($n + 1$) meters, he draws a cross (see picture for clarifications).</p><p>Constraints:</p><p> $1 \leq t \leq 50$ </p><p> $1 \leq n \leq 10$ </p></p><p>Input Format:</p><p>The first line contains integer t representing the number of testcases.</i></p><p>The second line contains t space-separated integers n representing the sides of the square for each test sample.</p><p>Output Format:</p><p>For each test sample print on a single line the answer to it, that is, the number of crosses Rohit will draw as he will move along the square of the corresponding size. </p><p>Print the answers to the samples in the order in which the samples are given in the input.</p>

answer

```

#include <iostream>
using namespace std;
template <class Paper>
Paper Square(Paper T){
    if(T%2==0)

```

```

        return 4*T+1;
    else if(T%4==1)
        return 2*T+1;
    else
        return T+1;
    }

int main()
{
    int T,n;
    cin>>T;
    while(T--){
        cin>>n;
        cout<<Square(n)<<endl;
    }
    return 0;
}

```

question

<p>Problem Description:
Jannu and Preethi both went to Egypt for visiting Pyramids. </p><p>On seeing the Pyramids they were in discussion. </p><p>During the discussion Jannu asked Preethi, what will be the area of this Pyramid. </p><p>Preethi have no idea about it. </p><p>Can you help Preethi in calculating the area of this Pyramid?
Functional Description:
Area = (height * base)/2

Constraints:
1 <= height <= 500
1 <= base <= 500

Input Format:
The only line of input has two floating point values representing height and base respectively separated by a space.</p><p>
Output Format:
In the only line of output print the area of the pyramid with only three values after decimal point.</p>

answer

```

#include <bits/stdc++.h>

using namespace std;

int main()
{

```

```

float height,base;

try{
    cin>>height;
    cin>>base;
    if(cin){
        cout<<fixed<<setprecision(3)<<height*base/2;
    }
    else
        throw 0;
}
catch(int cal){
    cout<<"Incomplete Information";
}
return 0;
}

```

question

<p>Question description:</p><p>Salman have conducted a test for his students recently.</p><p>The number of papers he has corrected last night was huge in number.</p><p>So he didn't have time to prepare the result statement.</p><p>Can you help him prepare the statement by getting the mark of the student and displaying if he/she have passed the test ?</p><p>Note:Salman have fixed 60 mark as the minimum passing mark.</p><p>Input Format:</p><p>Only line of input has a single value of type integer representing the mark of the student</p><p>Constraints:</p><p>0≤mark<100</p><p>Output Format:</p><p>Print the result</p>

answer

```

#include <iostream>
using namespace std;
class Student{
public:
    void accept(){}

```

```

};

class Test :public Student{
public:
void check(){}
};

class Result :public Test{
public:
int n;
void print(){
cin>>n;
if(n>=60)
cout<<"You have passed";
else
cout<<"You have failed";
}
};

int main()
{
Result r;
r.accept();
r.check();
r.print();
return 0;
}

```

question

<p>Question description:</p><p>Before the start of the Hockey season in Australia a strange magic ritual is held. </p><p>The most experienced magicians have to find a magic matrix of the size <i>n</i> × <i>n</i> (<i>n</i> is even number). </p><p>Gods will never allow to start the championship without it.</p><p>Matrix should contain integers from 0 to <i>n</i> - 1, main diagonal should contain only zeroes and matrix should be symmetric. </p><p>Moreover, all numbers in each row should be different.
Magicians are very tired of the thinking

process, so they ask you to write a program to find such matrix.

Constraints:

$2 \leq n \leq 1000$

Input Format:

The first line contains one integer n . n is even.

Output Format:

Output n lines with n numbers each the required matrix. Separate numbers with spaces.

answer

```
#include <bits/stdc++.h>
using namespace std;
class Ritual{
public:int Magic(int n){
    return 1;
}
int n;
int a[1001][1001];
int main(){
    cin>>n;
    Ritual find;
    find.Magic(n);
    for (int i=1;i<=n;++i){
        for (int j=1;j<=n;++j)
            a[i][j]=(i+j)%(n-1)+1;
    }
    for (int i=1;i<=n;++i){
        a[i][n]=a[n][i]=a[i][i];
        a[i][i]=0;
    }
    for (int i=1;i<=n;++i){
        for (int j=1;j<=n;++j)
            cout<<a[i][j]<<" ";
        cout<<'\n';
    }
}
```

}

question

<p>Question description:</p><p>Madonna has several rows of teeth, and feeds on crucians. </p><p>One of Madonna's unique feature is that while eating one crucian she uses only one row of her teeth, the rest of the teeth will relax.</p><p>For a long time madonna had been searching the sea for crucians, but a great misfortune happened. Her teeth started to ache, and she had to see the local dentist, lobster Rohan. </p><p>As a professional, Rohan quickly relieved Madonna from her toothache. Moreover, he managed to determine the cause of Madonna's developing caries.</p><p>It turned that Madonna eats too many crucians. </p><p>To help Madonna avoid further reoccurrence of toothache, Rohan found for each Madonna's tooth its residual viability. </p><p>Residual viability of a tooth is a value equal to the amount of crucians that Madonna can eat with this tooth. Every time Madonna eats a crucian, viability of all the teeth used for it will decrease by one. </p><p>When the viability of at least one tooth becomes negative, the Madonna will have to see the dentist again.</p><p>Unhappy, Madonna came back home, where a portion of crucians was waiting for her. </p><p>For sure, the couldn't say no to her favourite meal, but she had no desire to go back to the dentist. </p><p>That's why she decided to eat the maximum amount of crucians from the portion but so that the viability of no tooth becomes negative.</p><p>As Madonna is not good at mathematics, she asked you to help her to find out the total amount of crucians that she can consume for dinner.</p><p>We should remind you that while eating one crucian Madonna uses exactly one row of teeth and the viability of each tooth from this row decreases by one.</p><p>Constraints:</p><p>1 ≤ <i>m</i> ≤ <i>n</i> ≤ 1000</p><p>0 ≤ <i>k</i> ≤ 10⁶</p><p>1 ≤ <i>r</i> ≤ <i>m</i></p><p>0 ≤ <i>c</i> ≤ 10⁶</p><p>Input Format:</p><p>The first line contains three integers <i>n</i>, <i>m</i>, <i>k</i> representing total amount of Madonna's teeth, amount of tooth rows and amount of crucians in Madonna's portion for dinner. </p><p>Then follow <i>n</i> lines, each containing two integers: <i>r</i> representing index of the row, where belongs the corresponding tooth, and <i>c</i> representing its residual viability.</p><p>It's guaranteed that each tooth row has positive amount of teeth.</p><p>Output Format:</p><p>In the first line output the maximum amount of crucians that Madonna can consume for dinner.</p>

answer

```
#include <iostream>
using namespace std;
int n,m,k,r,c,i,s,a[1005];
int main(){
    cin>>n>>m>>k;
    for(i=1;i<=n;i++)a[i]=1e7;
    for(;n--){
```

```

    cin>>r>>c;
    a[r]=min(a[r],c);
}

for(i=1;i<=m;i++)s+=a[i]%10000000;
cout<<min(k,s);
return 0;
cout<<"map<int,set<int>>::iterator consume map<int,set<int>>Teeth;Teeth[r].insert(c);";
}

```

question

<p>Question description:</p><p>Metha is a Chief accounting officer of the Company.</p><p>Its 1st of the month “Salary Day”. He need to credit the salary to all the employees of the company within 2 hours.</p><p>But Metha is taking lot of time in getting the account details of the employees.</p><p>Can you automate the process by creating a programming logic for the same and help Metha?</p><p>Input Format:</p><p>The First line of input has five values of type string representing the employee number.</p><p>The Second line of input has a single value of type integer representing the employee name.</p><p>The Third line of input has a single value of type string representing the employee designation.</p><p>The fourth line of input has a single value of type string representing the employee basic pay.</p><p>The fifth line of input has a single value of type string representing the employee HRA.</p><p>The sixth line of input has a single value of type string representing the employee DA.</p><p>The seventh line of input has a single value of type string representing the employee PF.</p><p>The eighth line of input has a single value of type string representing the name of the bank employee has an account</p><p>The ninth line of input has a single value of type string representing the IFSC code of employee.</p><p>The tenth line of input has a single value of type string representing the Account number of employee.</p><p>Output Format:</p><p>Print the output in the expected format.</p><p>Refer sample testcases for format specification.</p>

answer

```

#include <iostream>
using namespace std;
class Employee{
};
class Salary : private Employee{
};
class BankCredit : private Salary{

```

```

public:

int num,pay,acc,hr,da,pf;

string name,des,bank,ifsc;

void getBankDetails(){

    cin>>num>>name>>des>>pay>>hr>>da>>pf>>bank>>ifsc>>acc;

}

void display(){

    cout<<"Emp number:"<<num<<endl;

    cout<<"Emp name:"<<name<<endl;

    cout<<"Emp designation:"<<des<<endl;

    cout<<"Emp Net Pay:"<<pay+hr+da-pf<<endl;

    cout<<"Emp Bank:"<<bank<<endl;

    cout<<"Emp IFSC:"<<ifsc<<endl;

    cout<<"Emp Account Number:"<<acc;

}

};

int main()

{

    BankCredit s;

    s.getBankDetails();

    s.display();

    return 0;

}

```

question

<p>Question description:</p><p>Janavi is an Quality Assurance Manager in a manufacturing firm.</p><p>During her review process of the design she used to compare the estimated area of different shape of equipment with the actual proposed equipment design.</p><p>She will complete her review process quickly if you can help her with a tool which can provide her the area of different shape of equipments.</p><p>Can you help her?</p><p>Constraints:</p><p>100≤length≤5000</p><p>100≤breadth≤5000</p><p>Input Format:</p><p>Only line of input has a two value of type integer representing length and breadth</p>

measurements respectively.</p><p>Output format:</p><p>In the first line of output print the area of the equipment which is rectangular in shape</p><p>In the second line of output print the area of the equipment which is triangular in shape</p>

answer

```
#include <iostream>
using namespace std;
class Shape{
public:
    int len,wid;
    void input(int l,int b){
        len=l;
        wid=b;
    }
};
class Rectangle: public Shape{
public:
    void output(){
        cout<<len*wid<<endl;
    }
};
class Triangle: public Shape{
public:
    void output(){
        //if((len*wid)%2==0)
        cout<<0.5*len*wid<<endl;
        //else
        //cout<<len*wid/2+1<<endl;
    }
};
int main()
{
```

```

int l,b;
cin>>l>>b;
Rectangle rect;
Triangle tri;
rect.input(l,b);
tri.input(l,b);
rect.output();
tri.output();
return 0;
}

```

question

<p>Question description:</p><p>Ragu requires basic staff information in order to properly maintain the files. </p><p>He's going to make a Google spreadsheet. </p><p>The sequence of the Google sheet is as follows: first name, last name, gender, college name, and category. </p><p>Please assist him in preparing the data collection sheets.</p><p>Input Format:</p><p>First Line: First name</p><p>Second Line: Last name</p><p>Third Line: Sex</p><p>Fourth Line: Age</p><p>Fifth Line: Institution</p><p>Sixth Line : Degree</p><p>Output Format:</p><p>Print the results as per format.</p><p>Refer sample testcases for format specification.</p>

answer

```

#include <iostream>
using namespace std;
class person{
public:
    string first,last,m,c,level;
    int age;
    void input_person();
    void display_person();
};
void person::input_person(){

```

```

    cin>>first>>last>>m>>age>>c>>level;
}

void person::display_person(){
    cout<<"First Name:"<<first<<endl<<"Last Name:"<<last<<endl<<"Gender:"<<m<<endl;
    cout<<"Age:"<<age<<endl<<"College:"<<c<<endl<<"Level:"<<level;
}

class student: public person{

public:
    void input_student(){}
    void display_student(){}
};

int main()
{
    student s;
    s.input_student();
    s.display_student();
    s.input_person();
    s.display_person();
}

```

question

<p>Question description:</p><p>Abilash's lifelong ambition was to be a photographer, so he bought a new camera. </p><p>Every day he got more and more clients asking for photos, and one day Abilash needed a program that would determine the maximum number of people he can serve.</p><p>The camera's memory is <i>d</i> megabytes. Abilash's camera can take photos of high and low quality. </p><p>One low quality photo takes <i>a</i> megabytes of memory, one high quality photo takes <i>b</i> megabytes of memory. </p><p>For unknown reasons, each client asks him to make several low quality photos and several high quality photos. More formally, the <i>i</i>-th client asks to make <i>x_i</i> low quality photos and <i>y_i</i> high quality photos.</p><p>Abilash wants to serve as many clients per day as possible, provided that they will be pleased with his work. To please the <i>i</i>-th client, Abilash needs to give him everything he wants, that is, to make <i>x_i</i> low quality photos and <i>y_i</i> high quality photos. </p><p>To make one low quality photo, the camera must have at least <i>a</i> megabytes of free memory space. </p><p>Similarly, to make one high quality photo, the camera must have at least <i>b</i> megabytes of free memory

space. Initially the camera's memory is empty. </p><p>Abilash also does not delete photos from the camera so that the camera's memory gradually fills up.</p><p>Calculate the maximum number of clients Abilash can successfully serve and print the numbers of these clients.</p><p>Constraints:</p><p>1 ≤ <i>n</i> ≤ 10⁵</p><p>1 ≤ <i>d</i> ≤ 10⁹</p><p>1 ≤ <i>a</i> ≤ <i>b</i> ≤ 10⁴</p><p>Input Format:</p><p>The first line contains two integers <i>n</i> and <i>d</i> — the number of clients and the camera memory size, correspondingly. </p><p>The second line contains two integers <i>a</i> and <i>b</i> — the size of one low quality photo and of one high quality photo, correspondingly.</p><p>Next <i>n</i> lines describe the clients. The <i>i</i>-th line contains two integers <i>x_i</i> and <i>y_i</i> — the number of low quality photos and high quality photos the <i>i</i>-th client wants, correspondingly.</p><p>All numbers on all lines are separated by single spaces.</p><p>Output Format:</p><p>On the first line print the answer to the problem — the maximum number of clients that Abilash can successfully serve. </p><p>Print on the second line the numbers of the client in any order. </p><p>All numbers must be distinct. If there are multiple answers, print any of them. </p><p>The clients are numbered starting with 1 in the order in which they are defined in the input data.</p>

answer

```
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
void solve(){
    cout<<"requestVec.end() std::vector<client>served;
std::vector<client>requestVec(n,PhotoClients); requestVec.begin()";
}
struct node
{
    ll v,id;
}c[200005];
ll ot[200005],p;
int main()
{
    ios::sync_with_stdio(0);
    ll n,d,a,b,x,y;
    cin>>n>>d>>a>>b;
    for(int i=1;i<=n;i++)

```

```

{
    cin>>x>>y;
    c[i].v=a*x+b*y;
    c[i].id=i;
}

sort(c+1,c+1+n,[](node a,node b){return a.v<b.v;});

for(int i=1;i<=n;i++)
{
    d=c[i].v;
    if(d<0) break;
    ot[++p]=c[i].id;
}

cout<<p<<"\n";
for(int i=1;i<=p;i++) cout<<ot[i]<<" ";
}

```

question

<p>Question description:</p><p>Arron was given n points on a plane. All points are different.</p><p>Now Arron needs to find the number of different groups of three points (A, B, C) such that point B is the middle of segment AC .</p><p>The groups of three points are considered unordered, that is, if point B is the middle of segment AC , then groups (A, B, C) and (C, B, A) are considered the same.</p><p>Constraints:</p><p> $3 \leq n \leq 3000$ </p>-
 $1000 \leq x_i < y_i \leq 1000$ </p><p>Input Format:</p><p>The first line contains a single integer n — the number of points.</p><p>Next n lines contain the points. </p><p>Next n lines contain the points. The i -th line contains coordinates of the i -th point: two space-separated integers x_i and y_i .</p><p>Output Format:</p><p>Print the single number — the answer to the problem.</p>

answer

```
#include<bits/stdc++.h>
using namespace std;
```

```

#define s scanf("%Id %Id",&points[p].first,&points[p].second);
#define f scanf("%Id",&n);

int n,x[3001],y[3001],ans;

bool point[2001][2001];

void solve(){

    cout<<"std::vector<std::pair<long,long>>points(n); sort(points.begin(),points.end());";
}

int main()

{

    cin>>n;

    for(int i=0;i<n;i++)

    {

        cin>>x[i]>>y[i];

        x[i]+=1000;

        y[i]+=1000;

        point[x[i]][y[i]]=1;

    }

    for(int i=0;i<n-1;i++)

        for(int j=i+1;j<n;j++)

            if((x[i]+x[j])%2==0&&(y[i]+y[j])%2==0)

                ans+=point[(x[i]+x[j])/2][(y[i]+y[j])/2];

    cout<<ans;

}

```

question

<p>Question description:</p><p>One day Dino got a letter in an envelope. </p><p>Dino knows that when Israeli post officers send a letter directly from city «A» to city «B», they stamp it with A B, or B A. </p><p>Unfortunately, often it is impossible to send a letter directly from the city of the sender to the city of the receiver, that's why the letter is sent via some intermediate cities. </p><p>Post officers never send a letter in such a way that the route of this letter contains some city more than once. Dino is sure that the post officers stamp the letters accurately.</p><p>Since his letter has n envelops dino knows that the possible routes of this letter

are only two. </p><p>But the stamps are numerous, and Dino can't determine himself none of these routes. </p><p>That's why he asks you to help him. </p><p>Find one of the possible routes of the letter.</p><p>Constraints:</p><p>1 ≤ <i>n</i> ≤ 10⁵</p><p>Input Format:</p><p>The first line contains integer <i>n</i> representing the amount of mail stamps on the envelope. </p><p>Then there follow <i>n</i> lines with two integers each representing the description of the stamps. </p><p>Each stamp is described with indexes of the cities between which a letter is sent. </p><p>The indexes of cities are integers from 1 to 10⁹.</p><p>Indexes of all the cities are different. </p><p>Every time the letter is sent from one city to another, exactly one stamp is put on the envelope. </p><p>It is guaranteed that the given stamps correspond to some valid route from some city to some other city.</p><p>Output Format:</p><p>Print <i>n</i> + 1 numbers representing the indexes of cities in one of the two possible routes of the letter.</p>

answer

```
#include<bits/stdc++.h>
using namespace std;
#define f(n) for(int i=0;i<n;i++)
map<int,vector<int>> Stamps;
void solve(){}
int main(){
    int n,x;
    cin>>n;
    for(int i=0;i<n;i++){
        int a,b;
        cin>>a>>b;
        Stamps[a].push_back(b);
        Stamps[b].push_back(a);    }
    for(auto i:Stamps)
        if(i.second.size()==1)x=i.first;
    cout<<x;
    int p=-1;
    f(n){
        if(Stamps[x][0]!=p)
```

```

x=Stamps[p=x][0];
else
x=Stamps[p=x][1];
cout<<' '<<x;
}
}

```

question

<p>Question Description:</p><p>A one-dimensional Indian crossword can be represented as a binary string of length <i>x</i>. Encoding of this crossword is an array of size <i>n</i>, where <i>n</i> is the number of segments formed completely of 1's, and <i>a_i</i> is the length of <i>i</i>-th segment. No two segments touch or intersect.</p><p>For example:</p>If <i>x</i> = 6 and the crossword is 111011, then its encoding is an array {3, 2};If <i>x</i> = 8 and the crossword is 01101010, then its encoding is an array {2, 1, 1};If <i>x</i> = 5 and the crossword is 11111, then its encoding is an array {5};If <i>x</i> = 5 and the crossword is 00000, then its encoding is an empty array.<p>Muhammad wants to create a new one-dimensional Indian crossword. He has already picked the length and the encoding for this crossword. And now he needs to check if there is exactly one crossword such that its length and encoding are equal to the length and encoding he picked. Help him to check it!</p><p>Constraints:</p><p>1 ≤ <i>n</i> ≤ 100000</p><p>1 ≤ <i>x</i> ≤ 10⁹</p><p>1 ≤ <i>a_i</i> ≤ 10000</p><p>Input Format:</p><p>The first line contains two integer numbers <i>n</i> and <i>x</i> the number of elements in the encoding and the length of the crossword Mishka picked.</p><p>The second line contains <i>n</i> integer numbers <i>a</i>₁, <i>a</i>₂, ..., <i>a_n</i> the encoding.</p><p>Output Format:</p><p>Print YES if there exists exactly one crossword with chosen length and encoding. Otherwise, print NO.</p>

answer

```

#include <iostream>
using namespace std;
class Indian{
    int n,x,c,sum=0;
public: void crossword(){
    cin>>n>>x;
    for(int i=0;i<n;i++){

```

```

    cin>>c;
    sum=sum+c;
}
if(sum-n+1==x)
cout<<"YES";
else
cout<<"NO";
}
};

int main()
{
    Indian inr;
    inr.crossword();
    return 0;
}

```

question

<p>Question description:</p><p>These days Fazil works as an air traffic controller at a large airport. </p><p>He controls a runway which is usually used for landings only. Thus, he has a schedule of planes that are landing in the nearest future, each landing lasts 1 minute.</p><p>He was asked to insert one takeoff in the schedule. </p><p>The takeoff takes 1 minute itself, but for safety reasons there should be a time space between the takeoff and any landing of at least s minutes from both sides.</p><p>Find the earliest time when Fazil can insert the takeoff.</p><p>Constraints:</p><p> $1 \leq n \leq 100$ </p><p> $1 \leq s \leq 60$ </p><p> $0 \leq h \leq 23$ </p><p> $0 \leq m \leq 59$ </p><p>Input Format:</p><p>The first line of input contains two integers n and s — the number of landings on the schedule and the minimum allowed time (in minutes) between a landing and a takeoff.</p><p>Each of next n lines contains two integers h and m — the time, in hours and minutes, when a plane will land, starting from current moment (i. e. the current time is 0 0). These times are given in increasing order.</p><p>Output Format:</p><p>Print two integers h and m — the hour and the minute from the current moment of the earliest time Arkady can insert the takeoff.</p>

answer

```
#include<bits/stdc++.h>
using namespace std;
```

```

int n,s,h,m,a,b;

class Takeoff{

public: void Time(){

    for(cin>>n>>s,a=0,b=-s-1;n--;){

        cin>>h>>m;

        if((h-a)*60+m-b>=2*s+2)break;

        a=h;b=m;

    }

    cout<<a+(b+s+1)/60<<" "<<(s+b+1)%60;

}

};

int main(){

    Takeoff obj;

    obj.Time();

}

```

question

<p>Question description:</p><p>Two-gram is an ordered pair (i.e. string of length two) of capital Latin letters. </p><p>For example, "AZ", "AA", "ZA" — three distinct two-grams.</p><p>You are given a string s consisting of n capital Latin letters. </p><p>Your task is to find any two-gram contained in the given string as a substring (i.e. two consecutive characters of the string) maximal number of times. </p><p>For example, for string $s = "BBAABBBA"$ the answer is two-gram "BB", which contained in s three times. In other words, find any most frequent two-gram.</p><p>Note that occurrences of the two-gram can overlap with each other.</p><p>Constraints:</p><p> $2 \leq n \leq 100$ </p><p>Input Format:</p><p>The first line of the input contains integer number n — the length of string s . </p><p>The second line of the input contains the string s consisting of n capital Latin letters.</p><p>Output Format:</p><p>Print the only line containing exactly two capital Latin letters — any two-gram contained in the given string s <math xmlns="http://www.w3.org/1998/Math/MathML"><mi>s</mi></math> as a substring (i.e. two consecutive characters of the string) maximal number of times.</p>

answer

```

#include<bits/stdc++.h>

using namespace std;

```

```

int n,mx;
string s,a,c;
map<string,int>m;
class StringPlay{
public: void Result(){
    cin>>n>>s;
    for(int i=0;i<n-1;i++){
        c=s[i];
        c+=s[i+1];
        m[c]++;
        if(m[c]>mx)mx=m[c],a=c;
    }
    cout<<a;
}
};

int main(){
StringPlay obj;
obj.Result();
}

```

question

<p>Question Description:</p><p>A very brave explorer Prabhakar once decided to explore Paris catacombs. Since Prabhakar is not really experienced, his exploration is just walking through the catacombs.</p><p>Catacombs consist of several rooms and bidirectional passages between some pairs of them. Some passages can connect a room to itself and since the passages are built on different depths they do not intersect each other. Every minute Prabhakar arbitrarily chooses a passage from the room he is currently in and then reaches the room on the other end of the passage in exactly one minute. When he enters a room at minute <i>i</i>, he makes a note in his logbook with number <i>t_i</i>:</p>If Prabhakar has visited this room before, he writes down the minute he was in this room last time;Otherwise, Prabhakar writes down an arbitrary non-negative integer strictly less than a current minute <i>i</i>.<p>Initially, Prabhakar was in one of the rooms at minute 0, he didn't write down number <i>t</i>₀.</p><p>At some point during his wandering, Prabhakar got tired, threw out his logbook, and went home. Vasya found his logbook and now he is curious: what is the minimum possible number of rooms in Paris catacombs according to Prabhakar's

logbook?</p><p>Constraints:</p><p>0 ≤ i ≤ 2 · 1
0 < n ≤ 5</p><p>Input Format:</p><p>The first line contains a single integer n , then the number of notes in Prabhakar's logbook. The second line contains n non-negative integers t_1, t_2, \dots, t_n notes in the logbook.</p><p>Output Format:</p><p>In the only line print a single integer the minimum possible number of rooms in Paris catacombs.</p>

answer

```
#include<bits/stdc++.h>
using namespace std;
class catacombs{
public:void arbitrary(){
    int n,a;
    cin>>n;
    set<int> se;
    for(int i=0;i<n;i++){
        cin>>a;
        se.insert(a);
    }
    cout<<n-se.size()+1;
}
};

int main(){
    catacombs ca;
    ca.arbitrary();
}
```

question

<p>Question description:</p><p>A necklace can be described as a string of links ('-') and pearls ('o'), with the last link or pearl connected to the first one.</p><p>You can remove a link or a pearl and insert it between two other existing links or pearls (or between a link and a pearl) on the necklace. This process can be repeated as many times as you like, but you can't throw away any parts.</p><p>Can you make the number of links between every two adjacent pearls equal? Two

pearls are considered to be adjacent if there is no other pearl between them.</p><p>Note that the final necklace should remain as one circular part of the same length as the initial necklace.</p><p>Constraints:</p><p>3≤|s|≤100</p><p>Input Format:</p><p>The only line of input contains a string s , representing the necklace, where a dash '-' represents a link and the lowercase English letter 'o' represents a pearl.</p><p>Output Format:</p><p>Print "YES" if the links and pearls can be rejoined such that the number of links between adjacent pearls is equal. Otherwise print "NO".</p>

answer

```
#include<bits/stdc++.h>
using namespace std;
string s;
int a,b;
class Necklace{
public: void Altering(){
    cin>>s;
    for(auto c:s)(c=='o'?a:b)++;
    cout<<(a==b?"NO":"YES");
}
};

int main()
{
    Necklace obj;
    obj.Altering();
    return 0;
}
```

question

<p>Question description:</p><p>Akash works in a famous College as a Junior Assistant. He has been assigned the task of collecting employee details by his HOD.</p><p>His colleague Amith provided him with the work of collecting student details</p><p>He need to prepare a specific document by gathering both the leaner and employee information from the learners and the employees respectively</p><p>Please assist in the creation of a document that meets the following formating.</p><p>Input Format:</p><p>First Line: Person code</p><p>Second Line: Person

name</p><p>Third Line: Person Role</p><p>Fourth Line: Student College</p><p>Fifth Line: Student IFSC code</p><p>Sixth Line: Person code </p><p>Seventh Line: Person name</p><p>Eighth Line: Person Role</p><p>Ninth Line: Employee Basic Pay</p><p>Tenth Line: Employee HRA</p><p>Eleventh Line: Employee DA</p><p>Twelfth Line: Employee PF</p><p>Output Format:</p><p>Print the results as per format.</p><p>Refer sample testcases for format specification.</p>

answer

```
#include <iostream>

using namespace std;

class Person{

};

class Employee : private Person{

};

class Student : private Person{

public:

int n1,n2,basic,hra,da,pf;

string name1,role1,col;ifsc,name2,role2;

void getdetail(){

    cin>>n1>>name1>>role1>>col>>ifsc>>n2>>name2>>role2;

}

void getEmployeeDetails(){

    cin>>basic>>hra>>da>>pf;

}

void student_display(){

    cout<<"Person number:"<<n1<<endl;

    cout<<"Person name:"<<name1<<endl;

    cout<<"Person Role:"<<role1<<endl;

    cout<<"Student college Name:"<<col<<endl;

    cout<<"Student IFSC:"<<ifsc<<endl;

    cout<<"Person number:"<<n2<<endl;

    cout<<"Person name:"<<name2<<endl;

    cout<<"Person Role:"<<role2<<endl;
```

```

}

void employee_display(){

    cout<<"Employee Basic pay:"<<basic<<endl;
    cout<<"Employee HRA:"<<hra<<endl;
    cout<<"Employee DA:"<<da<<endl;
    cout<<"Employee PF:"<<pf<<endl;
    cout<<"Employee Net Pay:"<<basic+hra+da-pf<<endl;
}

};

int main()

{

    Student e;

    e.getdetail();

    e.getEmployeeDetails();

    e.student_display();

    e.employee_display();

    return 0;

    cout<<"s.student_display();";
}

```

question

<p>Question description:</p><p>Prof.Geetha is a Head of the Department in the famous institution in the city.</p><p>She motivates her department students to do their final year project as internship to get industry exposure.</p><p>So many student also do so. Today is the Project review day were she need the details of the students taking up the project through internships.</p><p>Can you help Prof.Geetha in collecting those details?</p><p>Input Format:</p><p>The First line of input has five values of type string representing the name of the Student.</p><p>The Second line of input has a single value of type integer representing Registration number of the student.</p><p>The Third line of input has a single value of type string representing the company in which the internship is going on.</p><p>The fourth line of input has a single value of type string representing the degree student is studying.</p><p>Output Format:</p><p>Print the output in the expected format.</p><p>Refer sample testcases for format specification.</p>

answer

```
#include <iostream>

using namespace std;

class student{

};

class employee{

public:
    int roll;
    string name,intern,degree;

    void getcompany(){
        cin>>name>>roll>>intern;
    }

    void getpdegree(){
        cin>>degree;
    }

    void display(){
        cout<<"Name:"<<name<<endl;
        cout<<"Roll no:"<<roll<<endl;
        cout<<"Internship:"<<intern<<endl;
        cout<<"Degree:"<<degree;
    }
};

class project:public student,public employee{



};

int main()
{
    project p1;
    p1.getcompany();
    p1.getpdegree();
    p1.employee::display();
}
```

```
    return 0;  
}
```

question

<p>Question Description:</p><p>James has n different boxes. The first of them contains some balls of n different colors.</p><p>James wants to play a strange game. He wants to distribute the balls into boxes in such a way that every i ($1 \leq i \leq n$) i -th box will contain all balls with color i .</p><p>In order to do this, James will make some turns. Each turn he does the following:</p>James chooses any non-empty box and takes all balls from this box;Then James chooses any k empty boxes (the box from the first step becomes empty, and James is allowed to choose it), separates the balls he took on the previous step into k non-empty groups, and puts each group into one of the boxes. He should put each group into a separate box. He can choose either $k = 2$ or $k = 3$.<p>The <i>penalty</i> of the turn is the number of balls James takes from the box during the first step of the turn. And the <i>penalty</i> of the game is the total <i>penalty</i> of turns made by James until he distributes all balls to corresponding boxes.</p><p>Help James to determine the minimum possible <i>penalty</i> of the game!</p><p>Constraints:</p><p>1 $\leq n \leq 200000$ </p><p>1 $\leq a_i \leq 10^{10}$ </p><p>Input Format:</p><p>The first line contains one integer number n the number of boxes and colors.</p><p>The second line contains n integer numbers a_1, a_2, \dots, a_n , where a_i is the number of balls with color i .</p><p>Output Format:</p><p>Print one number the minimum possible <i>penalty</i> of the game.</p>

answer

```
#include <bits/stdc++.h>  
  
using namespace std;  
  
typedef long long ll;  
  
class boxes{  
  
public:  
    void colorBalls(){  
        ll n,a,ans=0;  
        priority_queue<ll,vector<ll>,greater<ll> > pq;  
        cin>>n;  
        for(int i=0;i<n;i++) cin>>a,pq.push(a);  
        if(!(n&1)) pq.push(0);  
        while(pq.size()!=1){
```

```

        a=pq.top();pq.pop();
        a+=pq.top();pq.pop();
        a+=pq.top();pq.pop();
        ans+=a;
        pq.push(a);
    }

    cout<<ans;
}

};

int main(){
    boxes b;
    b.colorBalls();
}

```

question

<p>Question description:</p><p>You are given a sequence of integers of length n and integer number k . </p><p>You should print any integer number x in the range of $1 \leq x \leq 10^9$ such that exactly k elements of given sequence are less than or equal to x .</p><p>Note that the sequence can contain equal elements.</p><p>If there is no such x , print "-1" (without quotes).</p><p>Constraints:</p><p> $1 \leq n \leq 2 \cdot 10^5$ </p><p> $0 \leq k \leq n$
 $1 \leq a_i \leq 10^9$ </p><p>Input Format:</p><p>The first line of the input contains integer numbers n and k </p><p>The second line of the input contains n integer numbers a_1, a_2, \dots, a_n the sequence itself.</p><p>Output Format:</p><p>Print any integer number x from range such that exactly k elements of given sequence is less or equal to x .</p><p>If there is no such x , print "-1" (without quotes).</p>

answer

```

#include<bits/stdc++.h>

using namespace std;int i,n,k,a[200005];

class Number{

public: void Range(){

    cin>>n>>k;

    for(a[0]=1;cin>>a[++i];);

    sort(a,a+n+1);
}

```

```

cout<<(a[k]!=a[k+1]?a[k]:-1);

}

};

int main(){

    Number obj;

    obj.Range();

}

```

question

<p>Question description:</p><p>Let's define a split of n <math

$$\text{xmlns="http://www.w3.org/1998/Math/MathML"}>\langle mi\rangle n\langle /mi\rangle \langle /math\rangle$$
 as a nonincreasing
sequence of positive integers, the sum of which is n <math

$$\text{xmlns="http://www.w3.org/1998/Math/MathML"}>\langle mi\rangle n\langle /mi\rangle \langle /math\rangle$$
.</p><p>For example, the
following sequences are splits of 8: [4,4], [3,3,2], [2,2,1,1,1,1], [5,2,1].</p><p>The following
sequences aren't splits of 8: [1,7], [5,4], [11,-3], [1,1,4,1,1].</p><p>The weight of a split is the
number of elements in the split that are equal to the first element. For example, the weight of the
split [1,1,1,1,1] is 5, the weight of the split [5,5,3,3,3] is 2 and the weight of the split [9] equals
1.</p><p>For a given n , find out the number of different weights of its
splits.</p><p>Constraints:</p><p>1≤ n ≤10^9</p><p>Input Format:</p><p>The first line contains
one integer n .</p><p>Output Format:</p><p>Output one integer — the answer to the
problem.</p>

answer

```

#include <iostream>

using namespace std;

class Sequence{

    int n;

public: void Split(){

    cin>>n;

    cout<<n/2+1;

}

};

int main()

{

```

```

Sequence obj;
obj.Split();
return 0;
}

```

question

<p>Question description:</p><p> k people want to split n candies between them. </p><p>Each candy should be given to exactly one of them or be thrown away.</p><p>The people are numbered from 1 to k , and Firaz is the first of them. </p><p>To split the candies, Firaz will choose an integer x and then give the first x candies to himself, the next x candies to the second person, the next x candies to the third person and so on in a cycle. </p><p>The leftover (the remainder that is not divisible by x) will be thrown away.</p><p>Firaz can't choose x greater than M as it is considered greedy. </p><p>Also, he can't choose such a small x that some person will receive candies more than D times, as it is considered a slow splitting.</p><p>Please find what is the maximum number of candies Firaz can receive by choosing some valid x .</p><p>Constraints:</p><p> $2 \leq n \leq 10^{18}$ </p><p> $2 \leq k \leq n$ </p><p> $1 \leq M \leq n$
 $1 \leq D \leq \min(n, 1000)$ </p><p> $M \cdot D \cdot k \geq n$ </p><p>Input Format:</p><p>The only line contains four integers n , k , M and D — the number of candies, the number of people, the maximum number of candies given to a person at once, the maximum number of times a person can receive</p><p>Output Format:</p><p>Print a single integer — the maximum possible number of candies Firaz can give to himself.</p><p>Note that it is always possible to choose some valid x .</p>

answer

```

#include<bits/stdc++.h>
using namespace std;
class Candies{
public: void Split(){
    int i=1,t,n,k,m,d,ans=0;
    cin>>n>>k>>m>>d;
    for(;i<=d;i++){
        if((n-1)/k<i-1)
            break;
        t=k*i-k+1;
        ans=max(ans,min(m,n/t)*i);
    }
}

```

```

    }

    cout<<ans;

}

};

int main()
{
    Candies obj;

    obj.Split();

}

```

question

<p>Question description:</p><p>For years, the Day of city N was held in the most rainy day of summer. New mayor decided to break this tradition and select a <i>not-so-rainy</i> day for the celebration. </p><p>The mayor knows the weather forecast for the n days of summer. On the i -th day, a_i millimeters of rain will fall. All values a_i are distinct.</p><p>The mayor knows that citizens will watch the weather x days before the celebration and y days after. Because of that, he says that a day d is <i>not-so-rainy</i> if a_d is smaller than rain amounts at each of x days before day d and each of y days after day d . </p><p>In other words, $a_d < a_j$ should hold for all $d - x \leq j < d$ and $d < j \leq d + y$. </p><p>Citizens only watch the weather during summer, so we only consider such j that $1 \leq j \leq n$.</p><p>Help mayor find the earliest <i>not-so-rainy</i> day of summer.</p><p>Constraints:</p><p> $1 \leq n \leq 100000$ </p><p> $0 \leq x \leq 7$ </p><p>Input Format:</p><p>The first line contains three integers n , x and y representing the number of days in summer, the number of days citizens watch the weather before the celebration and the number of days they do that after.</p><p>The second line contains n distinct integers a_1, a_2, \dots, a_n where a_i denotes the rain amount on the i -th day.</p><p>Output Format:</p><p>Print a single integer— the index of the earliest <i>not-so-rainy</i> day of summer. </p>

answer

```

#include <iostream>

using namespace std;

class Season{

    int n,x,y;

public: void PredictNotRainyDay(){

    cin>>n>>x>>y;

    int a[n];

```

```

for(int i=0;i<n;i++)
cin>>a[i];
int smallest=a[0],index=1;
for(int i=0;i<n;i++){
if(a[i]<=smallest){
smallest=a[i];
//cout<<a[i]<<endl;
index=i;
}
cout<<index+1;
}
};

int main()
{
Season obj;
obj.PredictNotRainyDay();
return 0;
}

```

question

<p>Problem Description:
Krishna has just arrived in the city of Madhura.</p><p>He brought an old house and renovating it. On seeing the pathetic floor conditions he planned to pave it with tile. </p><p>He has a m x n units of floor area and want to cover it up with 2x1 size tiles. </p><p>Krishna is no so good at calculations. </p><p>Could you help him to find out the minimum number tiles he needs to cover the floor?

Constraints:
1 ≤ m,n ≤ 500

Input Format:
Only line of input has two integers m and n separated by a space.

Output Format:
Print the minimum number of tiles need to pave the floor as output.</p>

answer

```
#include <iostream>
```

```
#include <math.h>
using namespace std;

int main(){
    int m,n;
    cin>>m>>n;
    int no=ceil(m*n/(2.0*1));
    cout<<no;
    return 0;}
```

question

<p>Problem Description:
Simon loves to listen to music while walking his way to attend boring lectures in his college.

He has a playlist of songs which has all songs of equal length, L. (in seconds)

One day while going on his way, he decided to calculate his average walking speed and he comes to know that he walks at a speed of 0.5 m/s.

You will be given the distance D ,he has to walk down to reach his class, after which he stops the music.

You have to find the minimum number of songs he needs to add into his playlist so as music plays in the whole path.

Constraints:
1<= L <=120 (in seconds)
1<= D <=5000 (in meters)
 </p><p>Input Format:
Only line of input contain two integer L and D separated by a space representing length of song and distance he has to walk respectively.

Output Format:
In the only line of output print the Integer value equal to number of songs he need to add into playlist before start to walk.</p>

answer

```
#include<iostream>
using namespace std;

int main()
{
    int L,D;
    cin>>L>>D;
    int sec=D/0.5;
    int song=sec/L+1;
    if(song!=sec){
        printf("%d",song);
```

```

    }
else{
    cout<<song;
}
return 0;
}

```

question

<p>Problem Description:
Today is Jack's birthday and he is looking forward to his gift. As usual, the gift is hidden and Jack has to follow a sequence of N instructions to reach it.

Initially, Jack is standing in the cell (0,0) of a two-dimensional grid. </p><p>The sequence of instructions is given as a string S. If we denote Jack's current cell by (x,y), each character of S corresponds to an instruction as follows:

'L' means to go left, i.e. to the cell (x-1,y)
'R' means to go right, i.e. to the cell (x+1,y)
'U' means to go up, i.e. to the cell (x,y+1)
'D' means to go down, i.e. to the cell (x,y-1)

In addition, Jack should never perform multiple consecutive moves along the same axis of the grid. </p><p>If there are multiple consecutive instructions to move along the same axis (left/right or up/down), he should perform only the first of these moves.

Find the cell (xg,yg) which contains the hidden gift.

Constraints:
1≤T≤100
1≤N≤1,000
S contains only characters 'L', 'R', 'U' and 'D'

Input Format:
The first line of the input contains a single integer T denoting the number of test cases. The description of T test cases follows.
The first line of each test case contains a single integer N.
The second line contains a single string S with length N.

Output Format:
Print a single line containing two space-separated integers xg and yg.</p>

answer

```

#include<bits/stdc++.h>
using namespace std;
int main()
{ char S[100];
int t,i,r,u,d,n;
int l;
scanf("%d",&t);
while(t--)
{
    int H[100]={};

```

```

scanf("%d",&n);
cin>>S;
for(i=0;i<n;i++)
{
if(S[i]=='R'&&S[i-1]!='L'&&S[i-1]!='R')
H[S[i]-65]++;
else if(S[i]=='L'&&S[i-1]!='R'&&S[i-1]!='L')
H[S[i]-65]++;
if(S[i]=='U'&&S[i-1]!='U'&&S[i-1]!='D')
H[S[i]-65]++;
if(S[i]=='D'&&S[i-1]!='U')
H[S[i]-65]++;
}
l=H[76-65];
r=H[82-65];
u=H[85-65];
d=H[68-65];
printf("%d %d\n",r-l,u-d);
}
return 0;
}

```

question

<p>Problem Description</p><p>Maran the head of data verification division of the popular Data Analytics company is responsible for verification of predicted change in data values based on some pattern from its initial value provided to him.</p><p>Since the data were huge in numbers, manual verification process is too difficult for Maran.</p><p>The expected data value pattern is as follows:</p><p>Decrement of First Number and Increment of Second Number</p><p>Increment of First Number and Decrement of Second Number</p><p>Decrement of First Number and Increment of Second Number</p><p>Increment of First Number and Decrement of Second Number</p><p>Decrement of First Number and Increment of Second Number</p><p>Function Description</p><p>Use postfix mode for firstnum </p><p>Use prefix mode for secondnum</p><p>Constraints</p><p>1≤firstnum≤500</p><p>1≤secondnum≤500</p><p>Input

Format:</p><p>Only line of Input has two integers separated by a space representing the value of firstnum and secondnum respectively.</p><p>Output Format:</p><p>Print the Output by performing the expected operation in the expected pattern.</p><p>Refer sample testcases for Format specification.</p>

answer

```
#include <iostream>
using namespace std;
int main()
{
    int firstnum,secondnum;
    cin>>firstnum>>secondnum;
    cout<<firstnum--<<" "<<++secondnum<<endl;
    cout<<firstnum++<<" "<<--secondnum<<endl;
    cout<<firstnum--<<" "<<++secondnum<<endl;
    cout<<firstnum++<<" "<<--secondnum<<endl;
    cout<<firstnum<<" "<<++secondnum;
    return 0;
}
```

question

<p>Problem Description:
Raina usually likes to play cricket, but now, he is bored of playing it too much, so he is trying new games with strings. Raina's friend Dhoni gave him binary strings S and R, each with length N, and told him to make them identical. </p><p>However, unlike Dhoni, Raina does not have any superpower and Dhoni lets Raina perform only operations of one type: choose any pair of integers (i,j) such that $1 \leq i, j \leq N$ and swap the i-th and j-th character of S. </p><p>He may perform any number of operations (including zero).

For Raina, this is much harder than cricket and he is asking for your help. </p><p>Tell him whether it is possible to change the string S to the target string R only using operations of the given type.

Constraints:
 $1 \leq T \leq 400$
 $1 \leq N \leq 100$
 $|S| = |R| = N$
S and R will consist of only '1' and '0'

Input Format:
The first line of the input contains a single integer T denoting the number of test cases. The description of T test cases follows.
The first line of each test case contains a single integer N.
The second line contains a binary string S.
The third line contains a binary string R.

Output Format:
Print a single line containing the string "YES" if it is possible to change S to R or "NO" if it is impossible (without quotes).</p>

answer

```

#include<iostream>
using namespace std;
#include <string.h>
int check(char ch)
{
    if(ch=='1')
        return 1;
    else
        return 0;
}
int main()
{ int i,t,n;
scanf("%d",&t);
while(t--)
{
    int count=0,count1=0;
    char S[100],R[100];
    scanf("%d",&n);
    cin>>S>>R;
    for(i=0;i<n;i++)
    {
        count+=check(S[i]);
        count1+=check(R[i]);
    }
    if(count==count1) printf("YES\n"); else printf("NO\n");
}
return 0;
}

```

question

<p>Problem Description</p><p>Karthik asks Jessi for a date .</p><p>But Karthik is a hardworking guy and has a value for money so he already pre-planned about his date and fixed a budget to spend from his savings .</p><p>Given a fixed budget B and an array A[] of size N for the amount of N expenses .</p><p>You have to calculate the total amount and check whether the date costs him beyond his fixed budget .&nbs;p;</p><p>If the total amount goes beyond budget then print “YES” otherwise “NO” .</p><p>Constraints:</p><p>1≤T≤100</p><p>1≤N≤10 power 4</p><p>1≤A[i]≤10 power 4</p><p>Input Format:</p><p>First line will contain T, number of testcases. Then T testcases follow.</p><p>First line of each testcase contains of a single line of input, two integers N,B.</p><p>Second line of each testcase contains N integers A[0],A[1],...,A[N-1] separated by a single space.</p><p>Output Format:</p><p>For each testcase, output in a single line answer YES or NO .</p>

answer

```

#include <iostream>

using namespace std;

int main()
{
    int t;
    cin>>t;
    while(t--)
    {
        int n,b,i,sum=0;
        cin>>n>>b;
        int a[n];
        for(i=0;i<n;i++)
        {
            cin>>a[i];
            sum+=a[i];
        }
        if(sum>b){
            cout<<"YES\n";
        }
        else{
            cout<<"NO\n";
        }
    }
}
  
```

```

    }
}

return 0;
}

```

question

<p>Problem Description:</p><p>There are K nuclear reactor chambers labelled from 0 to K-1. Particles are bombarded onto chamber 0. The particles keep collecting in the chamber 0. </p><p>However if at any time, there are more than N particles in a chamber, a reaction will cause 1 particle to move to the immediate next chamber(if current chamber is 0, then to chamber number 1), and all the particles in the current chamber will be destroyed and same continues till no chamber has number of particles greater than N. </p><p>Given K,N and the total number of particles bombarded (A), find the final distribution of particles in the K chambers. Particles are bombarded one at a time. After one particle is bombarded, the set of reactions, as described, take place. </p><p>After all reactions are over, the next particle is bombarded. If a particle is going out from the last chamber, it has nowhere to go and is lost.</p><p>Constraints:</p><p>A will be between 0 and 1000000000 inclusive. </p><p>N will be between 0 and 100 inclusive. </p><p>K will be between 1 and 100 inclusive. </p><p>Input Format:</p><p>The input will consist of one line containing three numbers A,N and K separated by spaces. </p><p>All chambers start off with zero particles initially.</p><p>Output Format:</p><p>Consists of K numbers on one line followed by a newline. </p><p>The first number is the number of particles in chamber 0, </p><p>The second number is the number of particles in chamber 1 and so on.</p>

answer

```

#include <stdio.h>

int main()

{int a,n,k,i,b;

scanf("%d%d%d",&a,&n,&k);

for(i=0;i<k;i++){

b=a%(n+1);

printf(" %d",b);

a=a/(n+1);

}

while(a>0){}

return 0;
}

```

```
printf("cin>>n>>b>>s;");  
}
```

question

<p>Problem Description

Yasir was making a kite.His sister Athika said that she can print the frame of the kite using biodegradable material and a 3D printer .</p><p>The shape of the frame is simple in the top it is triangle between the triangles mid point there comes a straight thicker line which extends upto the bottom.</p><p>Once printed a sheet of paper can be used to cover the frame. Athika made the frame using the 3d printer and asked Yasir to buy a sheet of paper .</p><p>But Yasir wants to know the exact area covered by the top triangle of the frame. </p><p>Athika had the dimensions fed in her laptop help her to compute the area of the triangle using the 3 sides.</p><p>
Functional Description :
Let s_1 , s_2 and s_3 be the lengths of the sides. </p><p>Let $s = (s_1 + s_2 + s_3)/2$. </p><p>Then the area of the triangle can be calculated using the following formula:

Area = $\sqrt{s \times (s - s_1) \times (s - s_2) \times (s - s_3)}$

Constraints:

1.00 ≤ s_1 ≤ 10.00

1.00 ≤ s_2 ≤ 10.00

1.00 ≤ s_3 ≤ 10.00

Input Format:

First Line: Single value representing the Length of side1</p><p>Second Line: Single value representing the Length of side2</p><p>Third Line: Single value representing the Length of side3

Output Format:

Print the area of the triangle.</p>

answer

```
#include<bits/stdc++.h>  
  
using namespace std;  
  
int main()  
{  
    float s1,s2,s3,s,area;  
  
    cin>>s1>>s2>>s3;  
  
    s=(s1+s2+s3)/2;  
  
    area=sqrt(s*(s-s1)*(s-s2)*(s-s3));  
  
    cout<<fixed<<setprecision(2)<<area;  
  
    return 0;}
```

question

<p>Problem Description:

Binita always dreamed of flying in the sky from her childhood.Her goal was to become a pilot.</p><p>When she applied for the pilot training program she cleared the entrance exam but failed the physical test due to overweight.</p><p>She forgot to take care of her health during the preparation.</p><p>But she had the spirit of not giving up.</p><p>Binita joined a gym to pass the physical test next time.</p><p>Her gym trainer asked her to calculate BMI for giving her a proper diet plan,But she had other works to do can you help this motivated girl?

Functional Description:

BMI = weight / height × height

Constraints:

40 ≤ weight ≤ 80

1.50 ≤ Height ≤ 1.72

Input Format:

First line has the value of weight of type integer
Second line has the value of height of type float

Output Format:
Print the BMI with only two values after decimal points.</p>

answer

```
#include <iostream>
#include<iomanip>
using namespace std;
int main()
{
    float height,bmi;
    int weight;
    cin>>weight;
    cin>>height;
    bmi=weight/(height*height);
    cout<<fixed<<setprecision(2)<<bmi;
    return 0;
}
```

question

<p>Problem Description:

Nathan was so fashion sensitive from his childhood.Nathan usually likes to wear different coloured shirts for different days (All 7 days in a week). </p><p>His mom will usually pick him the shirt in different colours for all the 7 days.But Nathans mom finding it difficult to remember the colour of the shirt she have picked for nathan each day.</p><p>If there is a smart mobile application that tells the colour of the shirt if the day number of the week is mentioned it will be very helpful for Nathan's mom.
Can You help her?
Functional Description:</p><p>1-Azure</p><p>2-Beige</p><p>3-Brick Red</p><p>4-Champagne</p><p>5-Desert sand</p><p>6-Ivory</p><p>7-Pear
In case of any other input print

as "Invalid Day"
Constraints:

1≤days≤20

Input Format:

Only line of input has single integer representing a day.

Output Format:

Print the colour of the shirt corresponding to the day.</p>

answer

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int days;
```

```
    scanf("%d",&days);
```

```
    switch(days)
```

```
{
```

```
    case 1:
```

```
        printf("Azure"); break;
```

```
    case 2:
```

```
        printf("Beige"); break;
```

```
    case 3:
```

```
        printf("Brick Red"); break;
```

```
    case 4:
```

```
        printf("Champagne"); break;
```

```
    case 5:
```

```
        printf("Desert sand"); break;
```

```
    case 6:
```

```
        printf("Ivory"); break;
```

```
    case 7:
```

```
        printf("Pear"); break;
```

```
    default:
```

```
        printf("Invalid Day"); break;
```

```
}
```

```
return 0;
```

```
printf("cin>>days;cout<<");  
}
```

question

<p>Question description</p><p>Sudhan is a sixth standard student. His Mathematics teacher gave the assignment to find the transpose of a matrix of order 3. </p><p>Can you help him to overload ~ the operator to find the transpose? </p><p>Constraints</p><p>-10≤n≤10, where n is the elements of the matrix</p><p>Input Format</p><p>First line represents the elements of the first row of the given matrix</p><p>Second line represents the elements of the second row of the given matrix</p><p>Third line represents the elements of the third row of the given matrix</p><p>Output Format</p><p>Print the transpose of the given matrix as in the test cases</p>

answer

```
#include <iostream>  
  
using namespace std;  
  
#define f(i,n) for(int i=0;i<n;i++)  
  
int main() {  
  
    int mat[10][10];  
  
    int a[10][10], row=3, column=3;  
  
    f(i,row)  
        f(j,column)  
            cin >> a[i][j];  
  
    f(i,row)  
        f(j,column)  
            mat[j][i] = a[i][j];  
  
    f(i,column){  
        f(j,row){  
            cout<< mat[i][j]<<" ";  
        }  
        cout<<endl;  
    }  
  
    return 0;  
}
```

}

question

<p>Question Description:</p><p>There is a special offer in Vaishnav's favourite supermarket: if the customer buys a chocolate bars, he or she may take b additional bars for free. This special offer can be used any number of times.</p><p>Vaishnav currently has s roubles, and he wants to get as many chocolate bars for free. Each chocolate bar costs c roubles. Help Vaishnav to calculate the maximum possible number of chocolate bars he can get!</p><p>Constraints:</p><p> $1 \leq t \leq 100$ </p><p> $1 \leq s, a, b, c \leq 10^9$ </p><p>Input Format:</p><p>The first line contains one integer t the number of testcases.</p><p>Each of the next t lines contains four integers s, a, b, c the number of roubles Vaishnav has, the number of chocolate bars you have to buy to use the special offer, the number of bars you get for free, and the cost of one bar, respectively.</p><p>Output Format:</p><p>Print t lines. i -th line should contain the maximum possible number of chocolate bars Vaishnav can get in i -th test.</p>

answer

```
#include<iostream>

using namespace std;

class supermarket{

};

class customer:public supermarket{

public:

    void chocolate(){

        long long n,m,s,x,t,res;

        cin>>t;

        while(t--){

            cin>>n>>m>>s>>x;

            res=n/x+(n/x/m)*s;

        }

        cout<<res;

    }

    void roubles(){}
};
```

```

int main(){
    customer cm;
    cm.chocolate();
    cm.roubles();
}

```

question

<p>Question Description:</p><p>Kanishma has three sticks of length a , b , and c centimeters respectively. In one minute Kanishma can pick one arbitrary stick and increase its length by one centimeter. She is not allowed to break sticks.</p><p>What is the minimum number of minutes she needs to spend increasing the stick's length in order to be able to assemble a triangle of positive area. Sticks should be used as triangle's sides (one stick for one side) and their endpoints should be located at triangle's vertices.</p><p>Constraints:</p><p> $1 \leq a, b, c \leq 100$ </p><p> $0 \leq a_i \leq 1$ </p><p>Input Format:</p><p>The only line contains tree integers a , b , and c the lengths of sticks Masha possesses.</p><p>Output Format:</p><p>Print a single integer the minimum number of minutes that Masha needs to spend in order to be able to make the triangle of the positive area from her sticks.</p>

answer

```

#include<bits/stdc++.h>
using namespace std;
class sticks{
};

class centimeters:public sticks{
public:
    void phase(){}
    void phase1(){
        int a[3];
        cin>>a[0]>>a[1]>>a[2];
        sort(a,a+3);
        cout<<max(0,1+a[2]-a[0]-a[1]);
    }
}

```

```
    }  
};
```

```
int main()  
{  
    centimeters cen;  
    cen.phase1();  
    cen.phase();  
}
```

question

<p>Question description:</p><p>Fazil is an athlete from his school time. Now he joined his under graduation in a famous institution which motivates students who are in sports. The Institution even provides scholarships for the sports quota.</p><p>So Fazil planned to apply for the scholarship for which he needs to calculate the percentage which considers the marks of CT1,CT2 and his Sports Performance marks.</p><p>Can you help Fazil by calculating the same?</p><p>Constraints:</p><p>1≤m1≤100</p><p>1≤m2≤100</p><p>1≤sm≤100</p><p>Input Format:</p><p>First line : Reg.Number</p><p>Second line : CT1 Mark</p><p>Third line : CT2 Mark</p><p>Fourth line : Sports Mark</p><p>Output format:</p><p>In the first line of output print the Reg.Number</p><p>In the second line of output print the total marks</p><p>In the third line of output print the percentage</p>

answer

```
#include <iostream>  
using namespace std;  
class student{  
public:  
    int reg,ct1,ct2;  
    void get(){  
        cin>>reg>>ct1>>ct2;  
    }  
};
```

```

class sports{
public:
int spm;
void getsm(){
    int d;
    cin>>d;
    spm=d;
}
};

class statement:public student,public sports{
public:
void display(){
cout<<reg<<endl<<ct1+ct2+spm<<endl<<(float)(ct1+ct2+spm)/3<<endl;
}
};

int main()
{
statement obj;
obj.get();
obj.getsm();
obj.display();
return 0;
}

```

question

<p>Question Description:</p><p>VSR and his friend Giraffe are currently in their room, solving some problems. Giraffe has written on the board an array a_1, a_2, \dots, a_n of integers, such that $1 \leq a_1 < a_2 < \dots < a_n \leq 103$, and then went to the bathroom.</p><p>VSR decided to prank his friend by erasing some consecutive elements in the array. Since he doesn't want for the prank to go too far, he will only erase it in a way, such that Giraffe can still restore the array using the information from the remaining elements.</p><p>Because Giraffe has

created the array, he's also aware that it's an increasing array and all the elements are integers in the range [1,10^3].</p><p>VSR wonders what is the greatest number of elements he can erase?</p><p>Constraints:</p><p>1≤n≤100</p><p>1≤a1< a2< ... < an≤10^3</p><p>Input Format:</p><p>The first line of the input contains a single integer n the number of elements in the array.</p><p>The second line of the input contains n integers ai the array is written by Giraffe</p><p>Output Format:</p><p>Print a single integer the maximum number of consecutive elements in the array that VSR can erase.</p><p>If it is impossible to erase even a single element, print 0.</p>

answer

```
#include<bits/stdc++.h>
using namespace std;
class friends{
};
class prank:public friends{
public:
    int n,i,j,r,a[179];
    void Giraffe(){}
    void far(){
        cin>>n;
        a[0]=0;a[n+1]=1001;
        for(i=1;i<=n;i++)cin>>a[i];
        n+=2;
        for(i=0;i<n-2;i++)
            for(j=i+2;j<n;j++)
                if(a[j]-a[i]==j-i)r=max(r,j-i-1);
        cout<<r;
    }
};
int main()
{
    prank p;
    p.Giraffe();
```

```
p.far();  
}
```

question

<p>Question Description:</p><p>Bharath shop sells n kinds of juices. Each juice has its price ci . Each juice includes some set of vitamins in it. There are three types of vitamins: vitamin "A", vitamin "B" and vitamin "C". Each juice can contain one, two or all three types of vitamins in it.</p><p>Peter knows that he needs all three types of vitamins to stay healthy. What is the minimum total price of juices that Peter has to buy to obtain all three vitamins? Peter obtains some vitamin if he buys at least one juice containing it and drinks it.</p><p>Constraints:</p><p> $1 \leq n \leq 1000$ </p><p> $1 \leq ci \leq 100000$ </p><p>Input Format:</p><p>The first line contains a single integer n the number of juices.</p><p>Each of the next n lines contains an integer ci and a string si the price of the i -th juice and the vitamins it contains. </p><p>String si contains from 1 to 3 characters, and the only possible characters are "A", "B", and "C". </p><p>It is guaranteed that each letter appears no more than once in each string si . The order of letters in strings si is arbitrary. Output Format:</p><p>Print -1 if there is no way to obtain all three vitamins. Otherwise, print the minimum total price of juices that Peter has to buy to obtain all three vitamins.</p>

answer

```
#include<bits/stdc++.h>  
  
using namespace std;  
  
class stayHealthy{  
  
};  
  
class vitamin:public stayHealthy{  
  
public:  
    void juice(){}  
    void drinks(){}  
};  
  
int main(){  
    vitamin vin;  
    vin.juice();  
    vin.drinks();  
    int n,i,j;
```

```

cin>>n;

long d[8]={0},c;
for(i=1;i<8;i++)
{
    d[i]=1000000;
}
for(i=0;i<n;i++)
{
    int s=0,string st;
    cin>>c>>st;
    int z=st.length();
    for(j=0;j<z;j++)
        s|=(1<<(st[j]-'A'));
    for(j=0;j<8;j++)
        d[s|j]=min(d[s|j],d[j]+c);
}
if(d[7]>=1000000)
    cout<<"-1";
else
    cout<<d[7];
return 0;
}

```

question

<p>Question Description:</p><p>Let's call the following process a transformation of a sequence of length n .</p><p>If the sequence is empty, the process ends. Otherwise, append the greatest common divisor (GCD) of all the elements of the sequence to the result and remove one arbitrary element from the sequence.</p><p>Thus, when the process ends, we have a sequence of n integers: the greatest common divisors of all the elements in the sequence before each deletion.</p><p>You are given an integer sequence $1, 2, \dots, n$. Find the lexicographically maximum

result of its transformation.</p><p>A sequence a_1, a_2, \dots, a_n is lexicographically larger than a sequence b_1, b_2, \dots, b_n , if there is an index i such that $a_j = b_j$ for all $j < i$, and $a_i > b_i$.</p><p>Constraints:</p><p>1 ≤ n ≤ 10^6</p><p>Input Format:</p><p>The first and only line of input contains one integer n .</p><p>Output Format:</p><p>Output n integers, the lexicographically maximum result of the transformation.</p>

answer

```
#include <cstdio>
```

```
class getInput{
```

```
};
```

```
class sequence:public getInput{
```

```
public:
```

```
void read(){
```

```
    int n; scanf("%d", &n);
```

```
    int pre = 1;
```

```
    for(int i = 0; i < n; i++) {
```

```
        int cur = pre;
```

```
        while(n / (cur + pre) ≥ n - i) cur += pre;
```

```
        printf("%d ", cur);
```

```
        pre = cur;
```

```
}
```

```
    printf("\n");
```

```
}
```

```
void check(){
```

```
}
```

```
};
```

```
int main() {
```

```
    sequence se;
```

```
    se.read();
```

```

    se.check();
    return 0;
}

```

question

<p>Question Description:</p><p>Two players A and B have a list of n integers each. They both want to maximize the subtraction between their score and their opponent's score.</p><p>In one turn, a player can either add to his score any element from his list (assuming his list is not empty), the element is removed from the list afterward. Or remove an element from his opponent's list (assuming his opponent's list is not empty).</p><p>Note, that in case there are equal elements in the list only one of them will be affected in the operations above. For example, if there are elements {1,2,2,3} in a list and you decided to choose 2 for the next turn, only a single instance of 2 will be deleted (and added to the score, if necessary).</p><p>Player A starts the game and the game stops when both lists are empty. Find the difference between A's score and B's score at the end of the game, if both of the players are playing optimally.</p><p>Optimal play between two players means that both players choose the best possible strategy to achieve the best possible outcome for themselves. In this problem, it means that each player, each time makes a move, which maximizes the final difference between his score and his opponent's score, knowing that the opponent is doing the same.</p><p>Constraints:</p><p> $1 \leq n \leq 100000$ </p><p> $1 \leq ai \leq 10^6$ </p><p> $1 \leq bi \leq 10^6$ </p><p>Input Format:</p><p>The first line of input contains an integer n the sizes of the list.</p><p>The second line contains n integers ai , describing the list of player A, who starts the game.</p><p>The third line contains n integers bi , describing the list of player B.</p><p>Output Format:</p><p>Output the difference between A's score and B's score ($A - B$) if both of them are playing optimally.</p>

answer

```

#include<bits/stdc++.h>
using namespace std;

int64_t n,i,x,r;
pair<int64_t,int> a[222000];

class players{

};

class score:public players{

};

```

```

void solve(){
    cout<<"s.instance();s.elements();score s;";
}

int main(){
    for(cin>>n,n*=2;i<n;i++)cin>>x,a[i]={x,i<n/2};

    sort(a,a+n);

    for(i=1;i<=n;i++){
        if(i%2&&a[n-i].second)r+=a[n-i].first;

        if(i%2==0&&a[n-i].second==0)r-=a[n-i].first;
    }

    cout<<r;
}

```

question

<p>Question Description:</p><p>There are n jelly's in a row. Each slime has an integer value (possibly negative or zero) associated with it.</p><p>Any jelly can eat its adjacent jelly (the closest slime to its left or to its right, assuming that this slime exists).</p><p>When a jelly with a value x eats a slime with a value y , the eaten jelly disappears, and the value of the remaining jelly changes to $x - y$.</p><p>The jellies will eat each other until there is only one slime left.</p><p>Find the maximum possible value of the last slime.</p><p>Constraints:</p><p> $1 \leq n \leq 500000$ </p><p> $-10^9 \leq a_i \leq 10^9$ </p><p>Input Format:</p><p>The first line of the input contains an integer n denoting the number of jellies.</p><p>The next line contains n integers a_i , where a_i is the value of i -th jelly.</p><p>Output Format:</p><p>Print only integer the maximum possible value of the last jelly.</p>

answer

```

#include<bits/stdc++.h>

#define ll long long

using namespace std;

class jelly{
};

```

```

class child:public jelly{
    public:
        void read(){
            int n;
            cin>>n;
            ll a[n];
            for(int i=0;i<n;i++)
                cin>>a[i];
            sort(a,a+n);
            ll ans=0;
            ans+=a[n-1];
            if(n>1)ans-=a[0];
            for(int i=1;i<n-1;i++)
                ans+=abs(a[i]);
            cout<<ans<<endl;
        }
        void check(){}
};

int main(){
    child ch;
    ch.check();
    ch.read();
}

```

question

<p>Question Description:</p><p>Christ has recently got a job as a cashier at a local store. His day at work is L minutes long. Christ has already memorized n regular customers, the i -th of which comes after t_i minutes after the beginning of the day, and his service consumes l_i minutes.</p><p>It is guaranteed that no customer will arrive while Christ is servicing another customer.</p><p>Christ is a bit lazy, so he likes taking smoke breaks for a minutes each.</p><p>Those breaks may go one after another, but Christ must be present at

work during all the time periods he must serve regular customers, otherwise one of them may alert his boss. What is the maximum number of breaks Christ can take during the day? Constraints:
 $0 \leq n \leq 10^5$
 $1 \leq L \leq 10^9$
 $1 \leq a \leq L$
 $0 \leq t_i \leq L - 1$
 $1 \leq l_i \leq L$
Input Format:
The first line contains three integers n , L and a .
The next n lines contain two integers t_i and l_i . It is guaranteed that $t_i + l_i \leq t_{i+1}$ and $tn + ln \leq L$.
Output Format:
If at least one phone number can be made from these cards, output the maximum number of phone numbers that can be made. Otherwise, output 0.

answer

```
#include<iostream>

int a,i,l,n,s,x,y,z;

class pattern{

};

class number:public pattern{
public:
    void cards(){}
    void digit(){}
};

int main(){
    for(std::cin>>n>>l>>a;
        n--;
        s+=(y-x)/a,x=y+z)std::cin>>y>>z;
    std::cout<<s+(l-x)/a;
    number num;
    num.digit();
    num.cards();
}
}
```

question

Question Description:
Krisnes have an unlimited number of coins with values $1, 2, \dots, n$. You want to select some set of coins having the total value of S . It is allowed to have

multiple coins with the same value in the set. What is the minimum number of coins required to get sum S ?
Constraints:
 $1 \leq n \leq 100000$
 $1 \leq S \leq 10^9$
Input Format:
The only line of the input contains two integers n and S .
Output Format:
Print exactly one integer the minimum number of coins required to obtain sum S .

answer

```
#include <iostream>
using namespace std;

class getInput{
public:
    int n,s;
    void read(){
        cin>>n>>s;
    }
};

class Divide:public getInput{
public:
    void write(){
        if(n<s){
            if(s%n==0)
                cout<<s/n;
            else
                cout<<s/n+1;
        }
        else
            cout<<"1";
    }
};

int main()
{
    Divide div;
```

```
    div.read();
    div.write();
}
```

question

<p>Question description:</p><p>Rohit is playing one-dimensional Sea Battle on a $1 \times n$ grid. In this game a ships are placed on the grid. Each of the ships consists of b consecutive cells. No cell can be part of two ships, however, the ships **can touch** each other.</p><p>Rohit doesn't know the ships location. She can shoot to some cells and after each shot she is told if that cell was a part of some ship (this case is called "hit") or not (this case is called "miss").</p><p>Rohit has already made k shots, all of them were misses.</p><p>Your task is to calculate the minimum number of cells such that if Rohit shoot at all of them, she would hit at least one ship.</p><p>It is guaranteed that there is at least one valid ships placement.</p><p>Input Format:</p><p>The first line contains four positive integers n , a , b , k representing the length of the grid, the number of ships on the grid, the length of each ship and the number of shots Rohit has already made.</p><p>The second line contains a string of length n , consisting of zeros and ones. If the i -th character is one, Rohit has already made a shot to this cell. Otherwise, she hasn't. </p><p>It is guaranteed that there are exactly k ones in this string.</p><p>Output Format:</p><p>In the first line print the minimum number of cells such that if Rohit shoot at all of them, she would hit at least one ship.</p><p>In the second line print the cells Rohit should shoot at.</p><p>Each cell should be printed exactly once. </p><p>You can print the cells in arbitrary order. The cells are numbered from 1 to n , starting from the left.</p>

answer

```
#include <bits/stdc++.h>
using namespace std;
int n,a,b,k,x,p,i,j,v[200005];
char m[200005];
class Shoot{
public:virtual void cells()=0;
};
class Ship:public Shoot{
public:
void cells(){
```

```

    }

};

int main(){

    Ship obj;

    obj.cells();

    cin>>n>>a>>b>>k;

    while(i<n){

        cin>>m[i];

        i++;

    }i=0;

    while(i<n){

        if(m[i]=='0')x++;

        else x=0;

        if(x==b)p++,x=0,v[j]=i+1,j++;

        i++;

    }i=0;

    cout<<p-a+1<<endl;

    while(i<p-a+1){

        cout<<v[i]<<" ";

        i++;

    }

}


```

question

<p>Question description:</p><p>Darshana has just graduated from one of the Top University in the World and is now attending celebration party. </p><p>Students like to dream of a beautiful life, so they used water glasses to construct a small pyramid. The height of the pyramid is

<i>n</i>. </p><p>The top level consists of only 1 glass, that stands on 2 glasses on the second level (counting from the top), then 3 glasses on the third level and so on. The bottom level consists of <i>n</i> glasses.</p><p>Darshana has seen in the movies many times how the water beautifully flows from top levels to bottom ones, filling all the glasses simultaneously. </p><p>So she took a bottle and started to pour it in the glass located at the top of the pyramid.</p><p>Each second, Darshana pours to the top glass the amount of water equal to the size of exactly one glass. </p><p>If the glass is already full, but there is some water flowing in it, then it pours over the edge of the glass and is equally distributed over two glasses standing under. </p><p>If the overflowed glass is at the bottom level, then the water pours on the table. </p><p>For the purpose of this problem we consider that water is distributed among pyramid glasses immediately. </p><p>Darshana is interested in the number of completely full glasses if she stops pouring water in <i>t</i> seconds.</p><p>Can you help Darshana?</p><p>Constraints:</p><p>1 ≤ <i>n</i> ≤ 10</p><p>0 ≤ <i>t</i> ≤ 10 000</p><p>Input Format:</p><p>The only line of the input contains two integers <i>n</i> and <i>t</i> representing the height of the pyramid and the number of seconds Darshana will be pouring water from the bottle.</p><p>Output Format:</p><p>Print the single integer representing the number of completely full glasses after <i>t</i> seconds.</p>

answer

```
#include<bits/stdc++.h>
using namespace std;
double a[11][11],d;
int n,t,i,j,sum;
class Glass{
public:virtual void Pouring()=0;
};
class Pyramid:public Glass{
public:
void Pouring(){
    cin>>n>>t;
    a[0][0]=(double)t;
    sum=0;
    for(i=0;i<n;i++)
    {
        for(j=0;j<=i;j++)
        {
            if(a[i][j]>=1.0)
```

```

    {
        d=(a[i][j]-1)/2;
        a[i+1][j]+=d;
        a[i+1][j+1]+=d;
        sum++;
    }
}

cout<<sum<<"\n";
}

};

int main()
{
    Prymid obj;
    obj.Pouring();
}

```

question

<p>Question description:</p><p>There are n cards (n is even) in the deck. </p><p>Each card has a positive integer written on it. $n / 2$ people will play new card game. </p><p>At the beginning of the game each player gets two cards, each card is given to exactly one player.</p><p>Find the way to distribute cards such that the sum of values written of the cards will be equal for each player. </p><p>It is guaranteed that it is always possible.</p><p>Constraints:</p><p> $2 \leq n \leq 100$ </p><p> $1 \leq a_i \leq 100$ </p><p>Input Format:</p><p>The first line of the input contains integer n representing the number of cards in the deck. It is guaranteed that n is even.</p><p>The second line contains the sequence of n positive integers a_1, a_2, \dots, a_n , where a_i is equal to the number written on the i -th card.</p><p>Output Format:</p><p>Print $n / 2$ pairs of integers, the i -th pair denote the cards that should be given to the i -th player. Each card should be given to exactly one player. Cards are numbered in the order they appear in the input.</p><p>It is guaranteed that solution exists. If there are several correct answers, you are allowed to print any of them.</p>

answer

```

#include<bits/stdc++.h>
using namespace std;
class Game{
public:virtual void Cards()=0;
};

class Distribution:public Game{
public:
void Cards(){
    int n;cin>>n;
    pair<int,int>p[n];
    for(int i=1;i<=n;i++){
        cin>>p[i].first;
        p[i].second=i;
        sort(p+1,p+n+1);
        for(int i=1;i<=n/2;i++)
            cout<<p[i].second<<" "<<p[n-i+1].second<<endl;
    }
};

int main(){
    Distribution obj;
    obj.Cards();
}

```

question

<p>Question description:</p><p>Hari commutes by train every day. </p><p>There are i train stations in the city, and at the i -th station it's possible to buy only tickets to stations from $i + 1$ to a_i inclusive. </p><p>No tickets are sold at the last station.</p><p>Let p_{ij} be the minimum number of tickets one needs to buy in order to get from stations i to station j . </p><p>As Hari is fond of different useless statistic he asks you to compute the sum of all values p_{ij} among all pairs $1 \leq i < j \leq n$.</p><p>Constraints:</p><p> $2 \leq n \leq 100\,000$ </p><p> $1 \leq i < j \leq n$ </p><p>Input</p>

Format:</p><p>The first line of the input contains a single integer n representing the number of stations.</p><p>The second line contains $n - 1$ integer a_i , the i -th of them means that at the i -th station one may buy tickets to each station from $i + 1$ to a_i inclusive.</p><p>Output Format:</p><p>Print the sum of $p_{i,j}$ among all pairs of $1 \leq i < j \leq n$.</p>

answer

```
#include <bits/stdc++.h>

#define ll long long
using namespace std;

const int N = 100005;

class Train{
public:virtual void Tickets()=0;
};

class Stations:public Train{
public:
void Tickets(){}
};

ll dp[N];
ll ret,Prev[N],x[N];
ll u,pos,i,j,n;

int main()
{
    Stations obj;
    obj.Tickets();
    cin>>n;
    for(i=1;i<=n-1;i++)
        cin>>x[i];
    Prev[++pos]=-n;dp[n]=0;
    for(i=n-1;i>=1;i--)
    {

```

```

        u=lower_bound(Prev+1,Prev+1+pos,-x[i])-Prev;
        u=-Prev[u];
        dp[i]=dp[u]-(x[i]-u)+n-i;
        ret+=dp[i];
        while(pos>0&&x[-Prev[pos]]<=x[i])pos--;
        Prev[++pos]=-i;
    }
    cout<<ret;
}

```

question

<p>Question description:</p><p>Sundar, like the hero of one famous comedy film, found a job as a night security guard at the museum. </p><p>At first night he received <i>embosser</i> and was to take stock of the whole exposition.</p><p><i>Embosser</i> is a special devise that allows to "print" the text of a plastic tape. Text is printed sequentially, character by character. </p><p>The device consists of a wheel with a lowercase English letters written in a circle, static pointer to the current letter and a button that print the chosen letter. </p><p>At one move it's allowed to rotate the alphabetic wheel one step clockwise or counterclockwise. </p><p>Initially, static pointer points to letter 'a'. </p><p>Other letters are located as shown on the picture:</p><p>After Sundar add new item to the base he has to print its name on the plastic tape and attach it to the corresponding exhibit. It's not required to return the wheel to its initial position with pointer on the letter 'a'.</p><p>Our hero is afraid that some exhibits may become alive and start to attack him, so he wants to print the names as fast as possible. </p><p>Help him, for the given string find the minimum number of rotations of the wheel required to print it.</p><p>Input Format:</p><p>The only line of input contains the name of some exhibit - the non-empty string consisting of no more than 100 characters. It's guaranteed that the string consists of only lowercase English letters.</p><p>Output Format:</p><p>Print one integer representing the minimum number of rotations of the wheel, required to print the name given in the input.</p>

answer

```

#include<bits/stdc++.h>
using namespace std;
class Museum{
public:virtual void Rotations()=0;
};

```

```

class Printing:public Museum{
public:
void Rotations(){
char c,p='a'; int r=0;
while(cin>>c){r+=min(abs(p-c), 26-abs(p-c)); p=c; } cout<<r;
}
};

int main()
{
    Printing obj;
    obj.Rotations();
    return 0;
}

```

question

<p>Question description:</p><p>As some of you know, cubism is a trend in art, where the problem of constructing volumetrical shape on a plane with a combination of three-dimensional geometric shapes comes to the fore.</p><p>A famous sculptor Arulmozhi, whose self-portrait you can contemplate, hates cubism. He is more impressed by the idea to transmit two-dimensional objects through three-dimensional objects by using his magnificent sculptures. And his new project is connected with this. </p><p>Arulmozhi wants to make a coat for the haters of anticubism. To do this, he wants to create a sculpture depicting a well-known geometric primitive — <i>convex polygon</i>.</p><p>Arulmozhi prepared for this a few blanks, which are rods with integer lengths, and now he wants to bring them together. The <i>i</i>-th rod is a segment of length <i>l_i</i>.</p><p>The sculptor plans to make a convex polygon with a nonzero area, using <i>all</i> rods he has as its sides. </p><p>Each rod should be used as a side to its full length. It is forbidden to cut, break or bend rods. However, two sides may form a straight angle 180 Degree.</p><p>Arulmozhi knows that it is impossible to make a convex polygon with a nonzero area out of the rods with the lengths which he had chosen. </p><p>Arulmozhi does not want to leave the unused rods, so the sculptor decides to make another rod-blank with an integer length so that his problem is solvable. </p><p>Of course, he wants to make it as short as possible, because the materials are expensive, and it is improper deed to spend money for nothing.</p><p>Help sculptor!</p><p>Constraints:</p><p>3 ≤ <i>n</i> ≤ 10⁵</p><p>1 ≤ <i>l_i</i> ≤ 10⁹</p><p>Input Format:</p><p>The first line contains an integer <i>n</i> representing the number of rod-blanks.</p><p>The second line contains <i>n</i> integers <i>l_i</i> representing the lengths of rods, which Arulmozhi already has. </p><p>It is guaranteed that it is impossible to make a polygon with <i>n</i> vertices and

nonzero area using the rods Arulmozhi already has.</p><p>Output Format:</p><p>Print the only integer <i>z</i> representing the minimum length of the rod, so that after adding it it can be possible to construct convex polygon with (<i>n</i> + 1) vertices and nonzero area from all of the rods.</p>

answer

```
#include<bits/stdc++.h>
using namespace std;
int n,m,i,j,k,l,p;
class Contruction {
public:virtual void MinLength()=0;
};
class Rod:public Contruction{
public:
void MinLength(){
    cin>>n;
    for(int i=1;i<=n;i++){
        cin>>k;
        l=max(l,k),p+=k;
    }
    cout<<2*l-p+1;
}
int main()
{
    Rod obj;
    obj.MinLength();
}
```

question

<p>Question description:</p><p>Arun is one of the best child dentists in Berland. Today <i>n</i> children got an appointment with him, they lined up in front of his office.</p><p>All children love to cry loudly at the reception at the dentist. We enumerate the children with integers from 1 to <i>n</i> in the order they go in the line. Every child is associated with the value of his <i>confidence</i> <i>p_i</i>. The children take turns one after another to come into the office; each time the child that is the first in the line goes to the doctor.</p><p>While Arun treats the teeth of the <i>i</i>-th child, the child is crying with the volume of <i>v_i</i>. At that the <i>confidence</i> of the first child in the line is reduced by the amount of <i>v_i</i>, the second one — by value <i>v_i</i> - 1, and so on. </p><p>The children in the queue after the <i>v_i</i>-th child almost do not hear the crying, so their <i>confidence</i> remains unchanged.</p><p>If at any point in time the <i>confidence</i> of the <i>j</i>-th child is less than zero, he begins to cry with the volume of <i>d_j</i> and leaves the line, running towards the exit, without going to the doctor's office. </p><p>At this the <i>confidence</i> of all the children after the <i>j</i>-th one in the line is reduced by the amount of <i>d_j</i>.</p><p>All these events occur immediately one after the other in some order. Some cries may lead to other cries, causing a chain reaction. </p><p>Once in the hallway it is quiet, the child, who is first in the line, goes into the doctor's office.</p><p>Help Arun the Dentist to determine the numbers of kids, whose teeth he will cure. </p><p>Print their numbers in the chronological order.</p><p>Constraints:</p><p>1 ≤ <i>n</i> ≤ 4000</p><p>1 ≤ <i>v_i</i>, <i>d_i</i>, <i>p_i</i> ≤ 10⁶</p><p>Input Format:</p><p>The first line of the input contains a positive integer <i>n</i> representing the number of kids in the line.</p><p>Next <i>n</i> lines contain three integers each <i>v_i</i>, <i>d_i</i>, <i>p_i</i> representing the volume of the cry in the doctor's office, the volume of the cry in the hall and the <i>confidence</i> of the <i>i</i>-th child.</p><p>Output Format:</p><p>In the first line print number <i>k</i> representing the number of children whose teeth Arun will cure.</p><p>In the second line print <i>k</i> integers — the numbers of the children who will make it to the end of the line in the increasing order.</p>

answer

```
#include<cstdio>
#define N 4010
using namespace std;
int v[N],d[N],p[N],c[N];
class Dentist{
public:virtual void Cure()=0;
};
class Kids:public Dentist{
public:
void Cure(){
int n,ans=0,i;
```

```

scanf("%d",&n);

for(i=0;i<n;i++)
    scanf("%d%d%d",&v[i],&d[i],&p[i]);

for(i=0;i<n;i++){
    if(p[i]>=0){
        c[ans++]=i;
        int cry=0;
        for(int j=i+1;j<n;j++){
            if(p[j]<0)
                continue;
            p[j]-=cry+v[i];
            if(p[j]<0&&cry<1e7) cry+=d[j];
            if(v[i]) v[i]--;
        }
    }
}
printf("%d\n",ans);
for(int i=0;i<ans;i++) printf("%d ",c[i]+1);
}

};

int main(){
    Kids obj;
    obj.Cure();
}

}

```

question

<p>Question Description:</p><p>Linga somehow found an array consisting of <i>n</i> integers. Looking at it, he came up with a task. Two players play the game on the array. Players move one by one. </p><p>The first player can choose for his move a subsegment of non-zero length with an odd sum of numbers and remove it from the array, after that the remaining parts are glued together into one array and the game continues. </p><p>The second player can choose a subsegment of non-zero length with an even sum and remove it. Loses the one who can not make a

move. Who will win if both play optimally?</p><p>Constraints:</p><p> $1 \leq n \leq 10^{6}$ </p><p> $0 \leq a_i \leq 10^9$ </p><p>Input Format:</p><p>The first line of input data contains a single integer n length of the array.</p><p>The next line contains n integers a_1, a_2, \dots, a_n .</p><p>Output Format:</p><p>Output answer in a single line. "First", if the first player wins, and "Second" otherwise.</p>

answer

```
#include<bits/stdc++.h>
using namespace std;
int n,x,r;
class Players{
public:void arrPlayer(){
    cin>>n;
    while(n--){
        cin>>x;
        if(x%2)
            r=1;
    }
    cout<<(r?"First":"Second");
}
};

int main(){
    Players pla;
    pla.arrPlayer();
    return 0;
}
```

question

<p>Question description:</p><p>Prashanth is playing Battleship. The rules of this game aren't really important.</p><p>There is a field of $n \times n$ cells. </p><p>There should be exactly one k -decker on the field, i. e. a ship that is k cells long oriented either horizontally or

vertically.</p><p>However, Prashanth doesn't know where it is located. For each cell Prashanth knows if it is definitely empty or can contain a part of the ship.</p><p>Consider all possible locations of the ship. </p><p>Find such a cell that belongs to the maximum possible number of different locations of the ship.</p><p>Constraints:</p><p>1≤k≤n≤100</p><p>Input Format:</p><p>The first line contains two integers n and k — the size of the field and the size of the ship.</p><p>The next n lines contain the field. Each line contains n characters, each of which is either '#' (denotes a definitely empty cell) or '.' (denotes a cell that can belong to the ship).</p><p>Output Format:</p><p>Output two integers— the row and the column of a cell that belongs to the maximum possible number of different locations of the ship.</p>

answer

```
#include<bits/stdc++.h>
using namespace std;
int n,k,cnt[103][103],sx=1,sy=1;
char s[103][103];
class Ship{
public: void Location(){}
};
int main(){
    Ship obj;
    obj.Location();
    scanf("%d%d",&n,&k);
    for(int i=1;i<=n;i++)scanf("%s",s[i]+1);
    for(int i=1;i<=n;i++){
        int pos=0;
        for(int j=1;j<=n;j++){
            if(s[i][j]=='#')pos=j;
            if(j-pos==k){
                for(int l=pos+1;l<=j;l++)cnt[i][l]++;
                pos++;
            }
        }
    }
    for(int j=1;j<=n;j++){
        int pos=0;
```

```

        for(int i=1;i<=n;i++){
            if(s[i][j]=='#')pos=i;
            if(i-pos==k){
                for(int l=pos+1;l<=i;l++)cnt[l][j]++;
                pos++;
            }
        }
        for(int i=1;i<=n;i++)
            for(int j=1;j<=n;j++)if(cnt[sx][sy]<cnt[i][j])sx=i,sy=j;
        printf("%d %d",sx,sy);return 0;
    }
}

```

question

<p>Question description:</p><p>A positive integer is called a <i>2-3-integer</i>, if it is equal to $2^i \times 3^j$ for some non-negative integers i and j . In other words, these integers are such integers that only have 2 and 3 among their prime divisors. </p><p>For example, integers 1, 6, 9, 16 and 108 — are 2-3 integers, while 5, 10, 21 and 120 are not.</p><p>Print the number of <i>2-3-integers</i> on the given segment $[l, r]$, i.e. the number of such <i>2-3-integers</i> t that $l \leq t \leq r$.</p><p>Constraints:</p><p>1 ≤ $l \leq r \leq 2 \cdot 10^9$ </p><p>Input Format:</p><p>The only line contains two integers l and r .</p><p>Output Format:</p><p>Print a single integer the number of <i>2-3-integers</i> on the segment</p>

answer

```

#include<bits/stdc++.h>
using namespace std;
class Numbers{
public: void Segment(){
    long long j,a,d=0,m,n,i;
    cin>>n>>m;
    for(i=0;i<=33;i++){
        for(j=0;j<=33;j++){
            a=pow(2,i)*pow(3,j);

```

```

        if(n<=a && a<=m)
        d++;
    }
}

cout<<d;
}

};

int main(){
    Numbers obj;
    obj.Segment();
}

```

question

<p>Question description:</p><p>Johan and his friends are playing a game of chips where there are n chips arranged in a circle, numbered from 1 to n .</p><p>Initially each chip has black or white color. </p><p>Then k iterations occur. During each iteration the chips change their colors according to the following rules. </p><p>For each chip i , three chips are considered: chip i itself and two its neighbours. </p><p>If the number of white chips among these three is greater than the number of black chips among these three chips, then the chip i <math xmlns="http://www.w3.org/1998/Math/MathML">i</math> becomes white. </p><p>Otherwise, the chip i becomes black.</p><p>Note that for each i from 2 to $(n-1)$ two neighbouring chips have numbers $(i-1)$ and $(i+1)$. </p><p>The neighbours for the chip $i=1$ are n and 2. </p><p>The neighbours of $i=n$ are $(n-1)$ and 1.</p><p>Johan now requests you to determine the color of each chip after k iterations.</p><figure class="image"></figure><p>Explanation:</p><p>The above picture describes one iteration with
 $n=6$. </p><p>The chips 1, 3 and 4 are initially black, and the chips 2, 5 and 6 are
white. </p><p>After the iteration 2, 3 and 4 become black, and 1, 5 and 6 become
white.</p><p>Input Format:</p><p>The first line contains two integers n and k representing the

number of chips and the number of iterations, respectively.
The second line contains a string consisting of n characters "W" and "B". c;</p><p>If the i -th character is "W", then the i -th chip is white initially. If the i -th character is "B", then the i -th chip is black initially.</p><p>Output Format:</p><p>Print a string consisting of n characters "W" and "B". If after k iterations the i -th chip is white, then the i -th character should be "W". c;</p><p>Otherwise the i -th character should be "B".</p>

answer

```
#include<bits/stdc++.h>
using namespace std;
void solve(){
    cout<<"template <typename Circle> Circle color(Circle c ) ";
}
int const N=2333333;
int n,k,f[N];
string s;
int main(){
    cin>>n>>k;
    cin>>s;
    for(int i=0;i<n;i++)
        f[i]=s[i]==s[(i+1)%n] | | s[i]==s[(i+n-1)%n]?0:2e9;
    for(int i=0;i<n;i++)f[i]=min(f[i],f[(i+n-1)%n]+1);
    for(int i=0;i<n;i++)f[i]=min(f[i],f[(i+n-1)%n]+1);
    for(int i=n-1;i>=0;i--)f[i]=min(f[i],f[(i+1)%n]+1);
    for(int i=n-1;i>=0;i--)f[i]=min(f[i],f[(i+1)%n]+1);
    for(int i=0;i<n;i++)
        cout<<(char)(s[i]^((min(f[i],k)%2?21:0)));
}
```

question

<p>Question description:</p><p>The city of Hampi can be imagined as a grid of 4 rows and an odd number of columns. It has two main villages; the first is located at the top-left cell (1,1), people who

stay there love fishing at the Tuna pond at the bottom-right cell $(4,n)$. The second village is located at $(4,1)$ and its people love the Salmon pond at $(1,n)$.</p><p>The mayor of Hampi wants to place k hotels in the city, each one occupying one cell. To allow people to enter the city from anywhere, hotels should not be placed on the border cells.</p><p>A person can move from one cell to another if those cells are not occupied by hotels and share a side.</p><p>Can you help the mayor place the hotels in a way such that there are equal number of shortest paths from each village to its preferred pond?</p><p>Constraints:</p><p> $3 \leq n \leq 99$ </p><p> $0 \leq k \leq 2 \times (n-2)$ </p><p>Input Format:</p><p>The first line of input contain two integers, n and k , n is odd, the width of the city, and the number of hotels to be placed, respectively.</p><p>Output Format:</p><p>Print "YES", if it is possible to place all the hotels in a way that satisfies the problem statement, otherwise print "NO".</p><p>If it is possible, print an extra 4 lines that describe the city, each line should have n characters, each of which is "#" if that cell has a hotel on it, or "." if not.</p>

answer

```
#include<bits/stdc++.h>
using namespace std;
int n,k,i,j;
char c[4][200];
class City{
public: void hotels(){
    cin>>n>>k;
    for(i=0;i<4;i++)for(j=0;j<n;j++)c[i][j]='.';
    for(j=1;j<3;j++)for(i=1;i<n/2&&k>1;i++,k-=2)c[j][i]=c[j][n-1-i]='#';
    if(k>0)c[1][n/2]='#';
    if(k>1)c[2][n/2]='#';
    cout<<"YES"<<endl;
    for(i=0;i<4;i++,cout<<endl)for(j=0;j<n;j++)cout<<c[i][j];
}
};

int main()
{
    City obj;
    obj.hotels();
}
```

question

<p>Question description:</p><p>After the big birthday party, Abilash still wanted Shalini to have some more fun. Later, she came up with a game called <i>treasure hunt</i>. Of course, he invited her best friends Jai and Shalini to play with her.</p><p>The three friends are very smart so they passed all the challenges very quickly and finally reached the destination. </p><p>But the treasure can only belong to one cat so they started to think of something which can determine who is worthy of the treasure. Instantly, Abilash came up with some ribbons.</p><p>A random colorful ribbon is given to each of the cats. Each color of the ribbon can be represented as an uppercase or lowercase Latin letter. </p><p>Let's call a consecutive subsequence of colors that appears in the ribbon a <i>subribbon</i>. </p><p>The <i>beauty</i> of a ribbon is defined as the maximum number of times one of its subribbon appears in the ribbon. The more the subribbon appears, the more beautiful is the ribbon. </p><p>For example, the ribbon aaaaaaa has the beauty of 7 because its subribbon a appears 7 times, and the ribbon abcdabc has the beauty of 2 because its subribbon abc appears twice.</p><p>The rules are simple. The game will have n turns. </p><p>Every turn, each of the cats must change strictly one color (at one position) in his/her ribbon to an arbitrary color which is different from the unchanged one. </p><p>For example, a ribbon aaab can be changed into acab in one turn. The one having the most beautiful ribbon after n <math
xmlns="http://www.w3.org/1998/Math/MathML"> n </math></p><p>turns wins the treasure.</p><p>Could you find out who is going to be the winner if they all play optimally?</p><p>Constraints:</p><p> $0 \leq n \leq 10^9$ </p><p>Input Format:</p><p>The first line contains an integer n — the number of turns.</p><p>Next 3 lines contain 3 ribbons of Jai, Shalini and Abilash one per line, respectively. Each ribbon is a string which contains no more than 10^5 uppercase and lowercase Latin letters and is not empty. It is guaranteed that the length of all ribbons are equal for the purpose of fairness. Note that uppercase and lowercase letters are considered different colors.</p><p>Output Format:</p><p>Print the name of the winner ("Jai", "Shalini" or "Abilash"). If there are at least two cats that share the maximum beauty, print "Draw".</p>

answer

```
#include <bits/stdc++.h>
using namespace std;
int m[4], r, n;
void solve(){}
class Friends{
public: void Hunt(){
    scanf("%d", &n);
    for(int i = 0; i < 3; i++) {
        int a[256]{} , l = 0, c;
```

```

        while(!isalpha(c = getchar()));

        for(; isalpha(c); c = getchar(), l++)
            m[i] = max(m[i], ++a[c]);

        if(m[i] == l) m[i] -= n == 1;
        else m[i] = min(m[i] + n, l);
        if(m[i] > m[r]) r = i;

    }

    for(int i = 0; i < 3; i++)
        if(i != r && m[i] == m[r]) {
            puts("Draw");
            return;
        }

    puts(r == 0 ? "Jai" : r == 1 ? "Shalini" : "Abilash");
}

};

int main() {
    Friends obj;
    obj.Hunt();
}

```

question

<p>Question Description:</p><p>Abdul is taking a geometry exam. Here is the last problem of the exam.</p><p>You are given three points a , b , c .</p><p>Find a point and an angle such that if we rotate the page around the point by the angle, the new position of a is the same as the old position of b , and the new position of b is the same as the old position of c .</p><p>Abdul is doubting if the problem has a solution or not (i.e. if there exists a point and an angle satisfying the condition). Help Abdul determines if the question has a solution or not. </p><p>Constraints:</p><p>| $a_x - b_x|, |a_y - b_y|, |b_x - c_x|, |b_y - c_y| \leq 10</math></p><p>Input Format:</p><p>The only line contains six integers $a_x, a_y, b_x, b_y, c_x, c_y$. It's guaranteed that the points are distinct.</p><p>Output Format:</p><p>Print "Yes" if the problem has a solution, "No" otherwise.</p><p>You can print each letter in any case (upper or lower).</p>$

answer

```
#include <bits/stdc++.h>
using namespace std;
int64_t ax,ay,bx,by,cx,cy;
class Geometry{
public:void Angle(){
    cin>>ax>>ay>>bx>>by>>cx>>cy;
    ax-=bx;ay-=by;cx-=bx;cy-=by;
    cout<<(cy*ax!=ay*cx&&ax*ay==cx*cy?"Yes":"No");
}
};

int main(){
    Geometry Geo;
    Geo.Angle();
}
```

question

<p>Question Description:</p><p><i>n</i> evenly spaced points have been marked around the edge of a circle. There is a number written at each point. You choose a positive real number <i>k</i>. Then you may repeatedly select a set of 2 or more points that are evenly spaced, and either increase all numbers at points in the set by <i>k</i> or decrease all numbers at points in the set by <i>-k</i>. You would like to eventually end up with all numbers equal to 0. Is it possible?</p><p>A set of 2 points is considered evenly spaced if they are diametrically opposed, and a set of 3 or more points is considered evenly spaced if they form a regular polygon. </p><p>Constraints:</p><p>3 ≤ <i>n</i> ≤ 100000</p><p>Input Format:</p><p>The first line of input contains an integer <i>n</i> the number of points along the circle.</p><p>The following line contains a string <i>s</i> with exactly <i>n</i> digits, indicating the numbers initially present at each of the points, in clockwise order.</p><p>Output Format:</p><p>Print "YES" (without quotes) if there is some sequence of operations that results in all numbers being 0, otherwise "NO" (without quotes).</p><p>You can print each letter in any case (upper or lower).</p>

answer

```
#include<bits/stdc++.h>
```

```

using namespace std;

const int N = 100005;

double a[N], b[N];

char s[N];

int n, m;

class diameter{

public:void circle(){

}

};

int main() {

diameter dm;

dm.circle();

cin>>n>>s; m = n;

for (int i=0; i<n; i++)

    a[i] = s[i] - '0';

for (int i=2; i<=n; i++)

    if (m % i == 0) {

        while (i >= 2 && m % i == 0) m /= i;

        for (int j=0; j<n; j++)

            b[j] = a[j];

        for (int j=0; j<n; j++)

            a[j] = b[(j+n/i)%n] - b[j];

    }

    for (int i=0; i<n; i++)

        if (fabs(a[i]) > 1e-8)

            return puts("NO"), 0;

    puts("YES");

    return 0;

}

```

question

<p>Question Description:</p><p>Winnie-the-Pooh likes honey very much! That is why he decided to visit his friends. Winnie has got three best friends: Rabbit, Owl, and Eeyore, each of their lives in his own house.&nbs;p;</p><p>There are winding paths between each pair of houses. The length of a path between Rabbit's and Owl's houses is <i>a</i> meter, between Rabbit's and Eeyore's house is <i>b</i> meters, between Owl's and Eeyore's house is <i>c</i> meters.</p><p>For enjoying his life and singing merry songs Winnie-the-Pooh should have a meal n time a day. Now he is in the Rabbit's house and has a meal for the first time. Each time when in the friend's house where Winnie is now the supply of honey is about to end, Winnie leaves that house.&nbs;p;</p><p>If Winnie has not had a meal the required amount of times, he comes out from the house and goes to someone else of his two friends. For this, he chooses one of two adjacent paths, arrives at the house on the other end, and visits his friend. You may assume that when Winnie is eating in one of his friend's houses, the supply of honey in other friend's houses recover (most probably, they go to the supply store).</p><p>Winnie-the-Pooh does not like physical activity. He wants to have a meal n time, traveling the minimum possible distance. Help him to find this distance.</p><p>Constraints:</p><p>1 ≤ <i>n</i> ≤ 100</p><p>1 ≤ <i>a</i> ≤ 100</p><p>1 ≤ <i>b</i> ≤ 100</p><p>1 ≤ <i>c</i> ≤ 100</p><p>Input Format:</p><p>First-line contains an integer <i>n</i> number of visits.</p><p>Second-line contains an integer a distance between Rabbit's and Owl's houses.</p><p>The third line contains an integer <i>b</i> the distance between Rabbit's and Eeyore's houses.</p><p>The fourth line contains an integer <i>c</i> the distance between Owl's and Eeyore's houses.</p><p>Output Format:</p><p>Output one number&nbs;minimum distance in meters Winnie must go through to have a meal n time.</p>

answer

```
#include<bits/stdc++.h>
using namespace std;

int n,a,b,c;

class Honey{
public:void Path(){
    cin>>n>>a>>b>>c;
    cout<<max(min(min(a,b)*(n-1),min(a,b)+c*(n-2)),0);
}
};

int main()
{
    Honey Ho;
    Ho.Path();
}
```

question

<p>Question description:</p><p>You have a team of <i>N</i> people.&nbs;p;</p><p>For a particular task, you can pick any non-empty subset of people.&nbs;p;</p><p>The cost of having <i>x</i> people for the task is <i>x^k</i>. </p><p>Output the sum of costs over all non-empty subsets of people.</p><p>Constraints:</p><p>1 ≤ <i>N</i> ≤ 10⁹</p><p>Input Format:</p><p>Only line of input contains two integers <i>N</i> representing total number of people and <i>k</i>.</p><p>Output Format:</p><p>Output the sum of costs for all non empty subsets modulo 10⁹ + 7.</p>

answer

```
#include <bits/stdc++.h>

const int mod = 1e9+7;

#define ll long long

int n,k,d;

ll f[2][5010];

ll Pow(ll a,ll b) {
    ll c=1;
    for(;b;b>>=1,a=a*a%mod)if(b&1)c=c*a%mod;
    return c;
}

class Team{
public: void Work(){
    scanf("%d%d", &n, &k);
    for(int i=0;i<=std::min(n,k);++i) f[d][i]=Pow(2,n-i);
    for(int i=1;i<=k;++i,d^=1)for(int j=0;j<=std::min(n,k);++j)f[d^1][j]=(j*f[d][j]+(n-j)*f[d][j+1])%mod;
    printf("%lld\n",f[d][0]);
}
};

int main() {
    Team obj;
    obj.Work();
}
```

```
    return 0;  
}
```

question

<p>Question Description:</p><p>It seems that Balaji is seriously sick. He is going to visit *n* doctors to find out the exact diagnosis. Each of the doctors needs the information about all previous visits, so Balaji has to visit them in the prescribed order (i.e. Balaji should first visit doctor 1, then doctor 2, then doctor 3, and so on). Balaji will get the information about his health from the last doctor.</p><p>Doctors have a strange working schedule. The doctor *j* goes to work on the *s_j*-th day and works every *d_j* days. So, he works on days *s_j, s_j + d_j, s_j + 2d_j, ..., s_j + (n - 1)d_j*. The doctor's appointment takes quite a long time, so Balaji can not see more than one doctor per day. What is the minimum time he needs to visit all doctors?</p><p>Constraints:</p><p>1 ≤ *n* ≤ 1000</p><p>1 ≤ *s_j* ≤ 1000</p><p>Input Format:</p><p>First-line contains an integer *n* — number of doctors.</p><p>Next, *n* lines contain two numbers *s_j* and *d_j*.</p><p>Output Format:</p><p>Output a single integer the minimum day at which Balaji can visit the last doctor.</p>

answer

```
#include <iostream>  
  
using namespace std;  
  
int s,d,k;  
  
class Doctors{  
  
public: void Diagnosis(){  
  
    for(std::cin>>s;std::cin>>s>>d;k=s)  
  
        while(s<=k)  
  
            s+=d;  
  
        cout<<k;  
  
    }  
  
};  
  
int main(){  
  
    Doctors Dr;  
  
    Dr.Diagnosis();  
  
}
```

question

<p>Question description:</p><p>A prisoner wants to escape from a prison. </p><p>The prison is represented by the interior of the convex polygon with vertices $P_1, P_2, P_3, \dots, P_{n+1}, P_{n+2}, P_{n+3}$. </p><p>It holds $P_1=(0,0)$, $P_{n+1}=(0,h)$, $P_{n+2}=(-10^{18},h)$ and $P_{n+3}=(-10^{18},0)$.</p><p>The prison walls $P_{n+1}P_{n+2}$, $P_{n+2}P_{n+3}$ and $P_{n+3}P_1$ are very high and the prisoner is not able to climb them. </p><p>Hence his only chance is to reach a point on one of the walls $P_1P_2, P_2P_3, \dots, P_nP_{n+1}$ and escape from there. </p><p>On the perimeter of the prison, there are two guards. </p><p>The prisoner moves at speed 1 while the guards move, remaining always on the perimeter of the prison, with speed v .</p><p>If the prisoner reaches a point of the perimeter where there is a guard, the guard kills the prisoner. If the prisoner reaches a point of the part of the perimeter he is able to climb and there is no guard there, he escapes immediately. Initially the prisoner is at the point $(-10^{17}, h/2)$ and the guards are at P_1 .</p><p>Find the minimum speed v such that the guards can guarantee that the prisoner will not escape (assuming that both the prisoner and the guards move optimally).</p></figure><p> </p><p>Functional
 Description:</p>At any moment, the guards and the prisoner can see each
 other.The "climbing part" of the escape takes no time.You may assume that both
 the prisoner and the guards can change direction and velocity instantly and that they both have
 perfect reflexes (so they can react instantly to whatever the other one is doing).The two
 guards can plan ahead how to react to the prisoner
 movements.<p>Constraints:</p><p>1≤n≤50</p><p>0≤xi,yi≤1,000</p><p>Input
 Format:</p><p>The first line of the input contains n</p><p>The following n+1 lines describe
 P1,P2,...,Pn+1. The i<math xmlns="http://www.w3.org/1998/Math/MathML"><mi>i</mi></math>-
 th of such lines contain two integers xi, yi representing the coordinates of Pi=(xi, yi).</p><p>It is
 guaranteed that P1=(0,0) and xn+1=0. </p><p>The polygon with vertices
 P1,P2,...,Pn+1,Pn+2,Pn+3 (where Pn+2,Pn+3 shall be constructed as described in the statement) is
 guaranteed to be convex and such that there is no line containing three of its
 vertices.</p><p>Output Format:</p><p>Print a single real number, the minimum speed v that
 allows the guards to guarantee that the prisoner will not escape.</p><p> Your answer will be
 considered correct if its relative or absolute error does not exceed 10^-6.</p>

answer

```
#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
```

```
typedef unsigned long long ULL;
```

```
#define SZ(x) ((int)((x).size()))
```

```
vector<int> SortIndex(int size, std::function<bool(int, int)> compare) {
```

```
    vector<int> ord(size);
```

```
    for (int i = 0; i < size; i++) ord[i] = i;
```

```
    sort(ord.begin(), ord.end(), compare);
```

```
    return ord;
```

```
}
```

```
template <typename T>
```

```
bool MinPlace(T& a, const T& b) {
```

```
    if (a > b) {
```

```
        a = b;
```

```
        return true;
```

```
}
```

```
    return false;
```

```
}
```

```
template <typename T>
```

```
bool MaxPlace(T& a, const T& b) {
```

```
    if (a < b) {
```

```
        a = b;
```

```
        return true;
```

```
}
```

```
    return false;
```

```
}
```

```
template <typename S, typename T>
```

```
ostream& operator <<(ostream& out, const pair<S, T>& p) {
```

```

    out << "{" << p.first << ", " << p.second << "}";
    return out;
}

template <typename T>
ostream& operator <<(ostream& out, const vector<T>& v) {
    out << "[";
    for (int i = 0; i < (int)v.size(); i++) {
        out << v[i];
        if (i != (int)v.size()-1) out << ", ";
    }
    out << "]";
    return out;
}

struct pt {
    double x, y;
    pt(): x(0), y(0) {}
    pt(double x, double y): x(x), y(y) {}
};

pt operator -(pt A, pt B) { return {A.x-B.x, A.y-B.y}; }

double operator *(pt A, pt B) { return A.x*B.x + A.y*B.y; }

double norm(pt A) { return sqrt(A*A); }

pt operator /(pt A, double lambda) { return {A.x/lambda, A.y/lambda}; }

ostream& operator<<(ostream& out, pt P) {
    out << "(" << P.x << ", " << P.y << ")";
    return out;
}

```

```

// P(s) = as2 + bs + c
// {l < s < r: P(s) < 0}

void neg_interval(double a, double b, double c, double l, double r,
                  vector<pair<double,double>>& ans) {
    assert(abs(a-1) < 1e-5 or a < 0);

    if (a > 0) {
        b /= a, c /= a, a = 1;

        if (b*b < 4*a*c) return;

        double delta = sqrt(b*b-4*a*c);

        double l0 = (-b - delta)/2;
        double r0 = (-b + delta)/2;

        l0 = max(l0, l);
        r0 = min(r0, r);

        if (l0 < r0) ans.emplace_back(l0, r0);
    }

    if (a < 0) {
        double s = sqrt(a*a + b*b + c*c);

        a /= s, b /= s, c /= s;

        if (b*b < 4*a*c) {
            ans.emplace_back(l, r);
            return;
        }

        double delta = sqrt(b*b-4*a*c);

        double l0 = (-b + delta)/(2*a);
        double r0 = (-b - delta)/(2*a);

        if (-0.01 < a) {
            if (b > 0) l0 = (-2*c)/(b+delta);
            else r0 = (2*c)/(-b + delta);
        }

        if (l < l0) ans.emplace_back(l, min(l0, r));
    }
}

```

```

    if (r0 < r) ans.emplace_back(max(l, r0), r);
}

}

void neg_interval(double a, double b, double c, double d,
                  double l0, double r0, double l1, double r1,
                  vector<pair<double,double>>& ans) {
    assert(a >= 1.99);

    double lmin = (-2 * r1 - c)/a;
    double rmin = (-2 * l1 - c)/a;
    lmin = max(lmin, l0);
    rmin = min(rmin, r0);

    if (lmin < rmin) neg_interval(1-a*a/4, b-a*c/2, d-c*c/4, lmin, rmin, ans);
    neg_interval(1, a*l1+b, l1*l1+c*l1+d, l0, r0, ans);
    neg_interval(1, a*r1+b, r1*r1+c*r1+d, l0, r0, ans);
}

```

```

void good_interval(pt A, pt B, pt C, pt D, double v, double l, vector<pair<double,double>>& ans) {
    double lA = 0;
    double rA = norm(B-A);
    pt dA = (B-A)/rA;
    double lC = 0;
    double rC = norm(D-C);
    pt dC = (D-C)/rC;

    if (v >= sqrt(2/(1+dA*dC))) return;

    double c1 = v*v - 1;
    double c2 = 2*(1- (v*v)*(dA*dC));
    double c3 = 2*v*v*((A-C)*dA) + 2*l;
    double c4 = -2*v*v*((A-C)*dC) - 2*l;
    double c5 = v*v * ((A-C)*(A-C)) - l*l;

    c2 /= c1, c3 /= c1, c4 /= c1, c5 /= c1;
}

```

```

neg_interval(c2, c3, c4, c5, lA, rA, lC, rC, ans);

}

bool nonempty_intersection(pair<double,double> l1, pair<double,double> l2) {
    return max(l1.first, l2.first) < min(l1.second, l2.second);
}

const int MAXN = 51;
pt P[MAXN];
double len[MAXN];

int main() {
    int N;
    cin>>N;
    for (int i = 0; i <= N; i++) cin >> P[i].x >> P[i].y;
    for (int i = 1; i < N; i++) len[i] = len[i-1] + norm(P[i]-P[i-1]);

    if (N <= 2) {
        cout << 1 << "\n";
        return 0;
    }

    double l = 1;
    double r = 20;
    for (int it = 0; it < 50; it++) {
        bool good = false;
        double v = (l+r)/2;
        for (int i = 1; i < N-1; i++) {
            vector<pair<double,double>> bef, aft;
            for (int j = 0; j < i; j++)
                good_interval(P[i], P[i+1], P[j], P[j+1], v, len[j]-len[i], bef);

```

```

        for (int j = i+1; j < N; j++)
            good_interval(P[i], P[i+1], P[j], P[j+1], v, len[j]-len[i], aft);
        for (auto& I: bef) for (auto& J: aft) good |= nonempty_intersection(I, J);
    }
    if (good) l = v;
    else r = v;
}

cout.precision(10);
cout << l << "\n";
}

```

question

<p>Question description:</p><p>Kanthamaran is the organizer of the Famous Boat Competition in God's Own Country.</p><p>There are n people who want to participate in a boat competition. </p><p>The weight of the i -th participant is w_i . </p><p>Only teams consisting of two people can participate in this competition. </p><p>As an organizer, Kanthamaran think that it's fair to allow only teams with the same total weight.</p><p>So, if there are k teams $(a_1, b_1), (a_2, b_2), \dots, (a_k, b_k)$, where a_i is the weight of the first participant of the i -th team and b_i is the weight of the second participant of the i -th team, then the condition $a_1+b_1=a_2+b_2=\dots=a_k+b_k=s$, where s is the total weight of each team, should be satisfied.</p><p>Kanthamaran request you to choose such s that the number of teams people can create is the maximum possible. </p><p>Note that each participant can be in no more than one team.</p><p>Constraints:</p><p> $1 \leq t \leq 1000$ </p><p> $1 \leq n \leq 50$ </p><p>Input Format:</p><p>The first line of the input contains one integer t the number of test cases. Then t test cases follow.
The first line of the test case contains one integer n representing the number of participants.</p><p>The second line of the test case contains n integers w_1, w_2, \dots, w_n where w_i is the weight of the i -th participant.</p><p>Output Format:</p><p>For each test case, print one integer k : the maximum number of teams people can compose with the total weight s , if you choose s optimally.</p>

answer

```

#include <bits/stdc++.h>

using namespace std;

int a[200];

template <typename Competition>

```

```
Competition Boat(Competition t){
```

```
    while(t--) {
```

```
        memset(a,0,sizeof(a));
```

```
        int n;
```

```
        cin>>n;
```

```
        for(int i=1,x;i<=n;i++){
```

```
            cin>>x;
```

```
            a[x]++;
```

```
        }
```

```
        int ans=0;
```

```
        for(int i=1;i<=100;i++){
```

```
            int sum=0;
```

```
            for(int j=1;j<=i;j++){
```

```
                sum+=min(a[j],a[i-j]);
```

```
                sum/=2;
```

```
                ans=max(ans,sum);
```

```
            }
```

```
            cout<<ans<<"\n";
```

```
        }
```

```
        return 1;
```

```
}
```

```
int main()
```

```
{
```

```
    int t;
```

```
    cin>>t;
```

```
    Boat(t);
```

```
    return 0;
```

```
}
```

question

<p>Problem Description:</p><p>Caleb and Irfan are purchasing apples which were priced according to their size. But their budget is minimum.</p><p>So they plan to choose one small, one medium and one large apple so that it will fit in their budget.
So can you help them choose the right apple by creating a logic by naming three apples they choose as
apple1,apple2,apple3. </p><p>Then check the condition if apple2 is greater than apple1 and apple3 is greater than apple2.</p><p>Constraints:</p><p>1≤ apple1 ≤600</p><p>1≤ apple2 ≤600</p><p>1≤ apple3 ≤600</p><p>Input format:

First Line: Single number of type integer representing the size of apple1</p><p>Second Line: Single number of type integer representing the size of apple2</p><p>Third Line: Single number of type integer representing the size of apple3

Output Format:
Print as “Fit into Budget” or “Dosen’t fit into Budget” based on the condition.</p><p>If the input is insufficient for deriving the result throw an exception message “Incomplete information”</p>

answer

```
#include <iostream>
using namespace std;
int main()
{
    int apple1,apple2,apple3;
    try{
        cin>>apple1;
        cin>>apple2;
        cin>>apple3;
        if(cin){
            if(apple2>apple1&&apple3>apple2)
                cout<<"Fit into Budget";
            else
                cout<<"Dosen't fit into Budget";
        }
        else throw 0;
    }
    catch(int budget){
        cout<<"Incomplete information";
    }
    return 0;
}
```

}

question

<p>Problem Description:</p><p>You are playing a Billiards-like game on an $N \times N$ table, which has its four corners at the points $\{(0,0), (0,N), (N,0), \text{ and } (N,N)\}$. </p><p>You start from a coordinate (x,y) , $(0 < x \leq N, 0 < y \leq N)$ and shoot the ball at an angle 45° with the horizontal. </p><p>On hitting the sides, the ball continues to move with the same velocity and ensuring that the angle of incidence is equal to the angle of reflection with the normal, i.e, it is reflected with zero frictional loss. </p><p>On hitting either of the four corners, the ball stops there and doesn't move any further.</p><p>Find the coordinates of the point of collision, when the ball hits the sides for the K th time. If the ball stops before hitting the sides K times, find the coordinates of the corner point where the ball stopped instead.</p><p>Constraints:</p><p> $2 \leq N \leq 25$ </p><p> $1 \leq K \leq 25$ </p><p>Input Format:</p><p>Each testcase contains a single line of input, which has four space separated integers - N , K , x , y , denoting the size of the board, the number of collisions to report the answer for, and the starting coordinates.</p><p>Output Format:</p><p>In the only line of output print the coordinates of the ball when it hits the sides for the K th time, or the coordinates of the corner point if it stopped earlier.</p><p>Explanation:</p><p>Assume the $N=5$ and $K=5$ $x=4$ and $y=4$ then</p><p>We shoot the ball from coordinates $(4,4)$, and we need to find its coordinates after it has collided with sides 5 times. However, after shooting, the ball goes directly to the corner $(5,5)$, and stops there. So we report the coordinates $(5,5)$.</p><figure class="image"></figure>

answer

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int n,k,x,y;
```

```
    try{
```

```
        cin>>n>>k>>x>>y;
```

```

if(cin){
    k%=4;

    if(x==y) cout<<n<<" "<<n<<"\n";
    else{
        if(k==1){
            if(x>y) cout<<n<<" "<<y+(n-x)<<"\n";
            else cout<<x+(n-y)<<" "<<n<<"\n";
        }
        else if(k==0){
            if(x>y) cout<<(x-y)<<" 0\n";
            else cout<<"0 "<<(y-x)<<"\n";
        }
        else if(k==2){
            if(x>y) cout<<y+n-x<<" "<<n<<"\n";
            else cout<<n<<" "<<x+n-y<<"\n";
        }
        else{
            if(x>y) cout<<"0 "<<x-y<<"\n";
            else cout<<y-x<<" 0"\n";
        }
    }
}

else throw 0;

}

catch(int n){
    cout<<"Invalid coordinate sites";
}

return 0;
}

```

question

<p>Problem Description</p><p>Vikram has just started Programming, he is in first year of Engineering. Vikram is reading about Relational Operators.
Relational Operators are operators which check relationship between two values. </p><p>Given two numerical values A and B you need to help Vikram in finding the relationship between them that is,
First one is greater than second or,</p><p>First one is less than second or,</p><p>First and second one are equal.</p><p>Constraints:</p><p>1 ≤ number1, number2 ≤ 50000</p><p>Input Format:</p><p>First line contains an integer T, which denotes the number of testcases. </p><p>Each of the T lines contain two integers A and B separated by a space.</p><p>Output Format:</p><p>For each line of input produce one line of output. This line contains any one of the relational operators '<', '>', '=' .</p>

answer

```
#include <iostream>

using namespace std;

int main()
{
    int number1,number2;
    try{
        cin>>number1>>number2;
        if(cin){
            if(number2>number1)
                cout<<"<";
            else if(number1==number2)
                cout<<"=";
            else
                cout<<">";
        }
        else throw 0;
    }
    catch(int a){
        cout<<"Input data missing";
    }
}
```

```
    }  
  
    return 0;  
}
```

question

<p>Problem Description:
Rohit has 'A' Chocolates and Mohit has 'B' Chocolates. </p><p>Rohit will do the following action 'K' times.

If Rohit has one or more Chocolates, eat one of his Chocolates.
Otherwise, if Mohit has one or more Chocolates, eat one of Mohit's Chocolates.
If they both have no Chocolates, do nothing.

Constraints:
0 <= A <= 10^12
0 <= B <= 10^12
0 <= K <= 10^12
All values in input are integers.

Input Format:
Only line of input has three integers A B K separated by a space

Output Format:
Print the numbers of Chocolates Rohit and Mohit have respectively after K actions</p>

answer

```
#include <iostream>  
  
using namespace std;  
  
int main()  
{  
  
    int a,b,k;  
  
    try{  
  
        cin>>a>>b>>k;  
  
        if(cin){  
  
            if(k<=a)  
  
                cout<<a-k<<" "<<b;  
  
            else if(k>=a+b)  
  
                cout<<"0 0";  
  
            else  
  
                cout<<a-a<<" "<<a+b-k;  
  
        }  
  
        else  
  
            throw 0;  
    }
```

```

    }

    catch(int a)

    {

        cout<<"Wrong input credentials";

    }

    //cin>>a>>b>>k;

    //cout<<a-a<<" "<<a+b-k;

    return 0;

}

```

question

<p>Problem Description:</p><p>Tina and Fazil are bored, so they are playing an infinite game of ping pong. The rules of the game are as follows:</p>The players play an infinite number of games. At the end of each game, the player who won it scores a point.In each game, one of the players <i>serves</i>. Tina serves in the first game.After every K points are scored (regardless of which players scored them), i.e. whenever K games have been played since the last time the serving player changed, the player that serves in the subsequent games changes: if Tina served in the game that just finished, then Fazil will serve in the next game and all subsequent games until the serving player changes again; if Fazil served, then Tina will serve.<p>The players got a little too caught up in the game and they forgot who is supposed to serve in the next game. </p><p>Will you help them determine that? So far, Tina has scored X points and Fazil has scored Y points.</p><p>Constraints:</p><p>0≤X,Y≤10 powert 9</p><p>1≤K≤10 power 9</p><p>Input Format:</p><p>The only line of each test case contains three space-separated integers X, Y and K.</p><p>Output Format:</p><p>In the only line of output print the string "Tina" if Tina is supposed to serve next or "Fazil" otherwise (without quotes).</p>

answer

```

#include <iostream>

using namespace std;

int main()

{

    int x,y,k;

    try{

        cin>>x>>y>>k;

        if(cin){


```

```

if(((x+y)/k)%2==0)
cout<<"Tina";
else
cout<<"Fazil";
}
else
throw 0;
}

catch(int a){
cout<<"Missing Input";
}

return 0;
}

```

question

<p>Problem description</p><p>Pari is an Architect who is currently doing his design work for his new project in one of the congested location of Paris. For making his work simpler he is looking for the automated tool which check whether the area is greater or perimeter is greater or both are equal if the Length (L) and Breadth (B) of a rectangle is provided.</p><p>Since he is not from the computing background he is looking for the freelancer who can do this for him?</p><p>Can you help him with the logic for doing so? </p><p>Constraints:</p><p>1≤L≤1000</p><p>1≤B≤1000</p><p>Input Format:</p><p>First line will contain the length (L) of the rectangle.</p><p>Second line will contain the breadth (B) of the rectangle.</p><p>Output Format:</p><p>Output 2 lines.</p><p>In the first line print "Area" if area is greater otherwise print "Peri" and if they are equal print "Eq".(Without quotes).</p><p>In the second line print the calculated area or perimeter (whichever is greater or anyone if it is equal).</p>

answer

```

#include <iostream>
using namespace std;
int main()
{
    int l,b;

```

```

try{
    cin>>l;
    cin>>b;
    if(cin){
        if((l*b)>(2*(l+b)))
            cout<<"Area\n"<<l*b;
        else if((l*b)==(2*(l+b)))
            cout<<"Eq\n"<<l*b;
        else
            cout<<"Peri\n"<<2*(l+b);
    }
    else
        throw 0;
}
catch(int parameters){
    cout<<"Provide Sufficient Size Information";
}
return 0;
}

```

question

<p>Problem Description:
Simon loves to listen to music while walking his way to attend boring lectures in his college.

He has a playlist of songs which has all songs of equal length, L (in seconds)

One day while going on his way, he decided to calculate his average walking speed and he comes to know that he walks at a speed of 0.5 m/s.

You will be given the distance D ,he has to walk down to reach his class, after which he stops the music.

You have to find the minimum number of songs he needs to add into his playlist so as music plays in the whole path.

Constraints:
1<= L <=120 (in seconds)
1<= D <=5000 (in meters)
 </p><p>Input Format:
Only line of input contain two integer L and D separated by a space representing length of song and distance he has to walk respectively.

Output Format:
In the only line of output print the Integer value equal to number of songs he need to add into playlist before start to walk.</p>

answer

```

#include <iostream>
using namespace std;
int main()
{
    int L,D;
    try{
        cin>>L>>D;
        if(cin){
            if((2*D)%L==0)
                cout<<(2*D)/L;
            else
                cout<<(2*D)/L+1;
        }
        else
            throw 0;
    }
    catch(int a){
        cout<<"Invalid input format";
    }
    return 0;
}

```

question

<p>Problem Description:
Vishal is fighting with a monster.The health of the monster is H.In one attack, Vishal can decrease the monster's health by 'A'. </p><p>There is no other way to decrease the monster's health.Vishal wins when the monster's health becomes '0' or below.
Can you find the number of attacks needed for Vishal for winning ?
Constraints:
1 <= H <= 10^4
1 <= A <= 10^4
All values in input are integers.

Input Format:
Only line of input has two space separated integers H A

Output Format:
Print the number of attacks Vishal needs to make before winning.</p>

answer

```

#include <iostream>
using namespace std;
int main()
{
    int h,a;
    try{
        cin>>h>>a;
        if(cin){
            if(h%a==0)
                cout<<h/a;
            else if(h%a!=0)
                cout<<h/a+1;
        }
        else
            throw 0;
    }
    catch(int a){
        cout<<"Missing Input Data";
    }
    return 0;
}

```

question

<p>Problem Description:</p><p>For her next karate demonstration, Prasad will break some bricks.</p><p>Prasad stacked three bricks on top of each other. Initially, their widths (from top to bottom) are W_1, W_2, W_3 .</p><p>Prasad's strength is S . Whenever she hits a stack of bricks, consider the largest $k \geq 0$ such that the sum of widths of the topmost k bricks does not exceed S ; the topmost k bricks break and are removed from the stack. Before each hit, Prasad may also decide to reverse the current stack of bricks, with no cost.</p><p>Find the minimum number of hits Prasad needs in order to break all bricks if she performs the reversals optimally. </p><p>You are not required to minimise the number of reversals.</p><p>Constraints:</p><p> $1 \leq S \leq 8$ </p><p> $1 \leq W_i \leq 2$, for each valid i </p><p>Input Format:</p><p>Only line of input contains four space-separated integers S, W_1, W_2 and W_3 .</p>

and W3.</p><p>Output Format:</p><p>In only line of output print the integer representing the minimum required number of hits.</p>

answer

```
#include <iostream>
using namespace std;

int main()
{
    int s,w1,w2,w3;
    try{
        cin>>s>>w1>>w2>>w3;
        if(cin){
            int res=w1+w2+w3;
            if(s>=res)
                cout<<"1";
            else if(s<res&&res%s==0)
                cout<<res/s;
            else
                cout<<res/s+1;
        }
        else
            throw 0;
    }
    catch(int q){
        cout<<"Invalid Bricks Input";
    }
    return 0;
}
```

question

<p>Problem Description:</p><p>There are two monkeys on an x-axis ready to jump in the positive direction (i.e, toward positive infinity). </p><p>The first monkey starts at location x_1 and moves at a rate of v_1 meters per jump. </p><p>The second monkey starts at location x_2 and moves at a rate of v_2 meters per jump. </p><p>Given the starting locations and movement rates for each monkey, can you determine if they'll ever land at the same location at the same time?</p><p>Constraints:</p><p>• $0 \leq x_1 < x_2 \leq 10000$ </p><p>• $1 \leq v_1 \leq 10000$ </p><p>• $1 \leq v_2 \leq 10000$ </p><p>Input Format</p><p>A single line of four space-separated integers denoting the respective values of x_1 , v_1 , x_2 and v_2 .</p><p>Output Format:</p><p>Print YES if they can land on the same location at the same time; otherwise, print NO.</p><p>Note: The two monkeys must land at the same location after making the same number of jumps.</p>

answer

```
#include <iostream>
using namespace std;
int main()
{
    int x1,x2,v1,v2;
    try{
        cin>>x1>>x2>>v1>>v2;
        if(cin){
            if((x2-x1+v2-v1)%(v1-v2)>0)
                cout<<"YES";
            else
                cout<<"NO";
        }
        else
            throw 0;
    }
    catch(int a){
        cout<<"Input coordinates";
    }
    return 0;
}
```

question

answer

```
#include<stdio.h>

#include<cstring>

#define f(i,n) for(int i=1;i<=n;i++)

#define g(n) for(int j=i+1;j<=n;j++)

#define h(n) for(int i=2;i<=n;i++)

#define dec int s1l[21],s1r[21],s2l[21],s2r[21],a[51][21],n,m,ans;

#define for1 for (last=0,now=1;now;last=now,now=(a[u][i]<=a[u][now])?s1l[now]:s1r[now]);

#define for2 for (last=0,now=1;now;last=now,now=(a[v][i]<=a[v][now])?s2l[now]:s2r[now]);

#define dfs3 void dfs(const int &now1,const int &now2){ if ((bool)now1 ^ (bool)now2) flag=0; if (!flag) return; if (s1l[now1]) dfs(s1l[now1],s2l[now2]); if (s1r[now1]) dfs(s1r[now1],s2r[now2]);}

#define inline1 inline void judge(const int &u,const int &v){s1l[0]=0;s1r[0]=0;int now,last;memset(s1l,0,sizeof(s1l));memset(s2l,0,sizeof(s2l));memset(s1r,0,sizeof(s1r));memset(s2r,0,sizeof(s2r));}
```

```

dec

//int s1l[21],s1r[21],s2l[21],s2r[21],a[51][21],n,m,ans;
bool flag,f[51];

dfs3

/*void dfs(const int &now1,const int &now2)
{
if ((bool)now1 ^ (bool)now2) flag=0;
if (!flag) return;
if (s1l[now1]) dfs(s1l[now1],s2l[now2]);
if (s1r[now1]) dfs(s1r[now1],s2r[now2]);
}*/
inline1

/*inline void judge(const int &u,const int &v)
{
s1l[0]=0;s1r[0]=0;int now,last;
memset(s1l,0,sizeof(s1l));
memset(s2l,0,sizeof(s2l));
memset(s1r,0,sizeof(s1r));
memset(s2r,0,sizeof(s2r));*/
h(m)

{
    for1

    //for (last=0,now=1;now;last=now,now=(a[u][i]<=a[u][now])?s1l[now]:s1r[now]);
    if (a[u][i]<=a[u][last]) s1l[last]=i;
    else s1r[last]=i;
}

s2l[0]=0;s2r[0]=0;
for (int i=2;i<=m;i++)
{
    for2

    //for (last=0,now=1;now;last=now,now=(a[v][i]<=a[v][now])?s2l[now]:s2r[now]);
}

```

```
if (a[v][i]<=a[v][last]) s2l[last]=i;
else s2r[last]=i;
}
flag=1;
dfs(1,1);
if (flag) f[v]=1,ans--;
}

int main(){
scanf("%d%d",&n,&m);
f(i,n) f(j,m)/*for (int j=1;j<=m;j++)*/ scanf("%d",&a[i][j]);
f(i,n) g(n) if (!f[i] && !f[j]) judge(i,j);
printf("%d",ans+n);
return 0;
printf("vector<vector<int>>tree(N,vector<int>(K)); cin>>N>>K");}
```