

Q. Fees Calculator

Write a program that displays the fees that a student needs to pay at the end of the semester. Use switch statement.

Input Format:

The first line of the input consists of a character from the set (A, B, C, D). A corresponds to a day scholar student with the required attendance percentage. B corresponds to a day scholar student without the required attendance percentage. C corresponds to a hostel student with the required attendance percentage. D corresponds to a hostel student without the required attendance percentage.

The second line of the input consists of an integer which corresponds to the number of regular papers for which the student is appearing in the examination.

The third line of the input consists of an integer which corresponds to the fee to be paid per regular paper.

The fourth line of the input consists of an integer which corresponds to the number of backup(arrear) papers for which the student is appearing in the examination.

The fifth line of the input consists of an integer which corresponds to the fee to be paid per arrear paper.

Output Format:

The output consists of a single line. Refer to Sample output for format details.

[There is a condonation fee or Rs. 5000 for lack of attendance and the hostel students need to pay the last month mess bill of Rs.1500 along with the examination fee.]

Mandatory:

You should use Switch case to get 100% evaluation

Source Code

```
#include <stdio.h>
int main()
{char x;
int a,b,c,d,e,f,g,h;
scanf("%c%d%d%d",&x,&a,&b,&c,&d);
e=(a*b)+(c*d);
f=e+5000;
g=e+1500;
h=g+5000;
switch (x)
{case 'A':
printf("The fee to be paid is Rs=%d",e);
break;
case 'B':
printf("The fee to be paid is Rs=%d",f);
break;
case 'C':
printf("The fee to be paid is Rs=%d",g);
break;
case 'D':
printf("The fee to be paid is Rs=%d",h);
break;
default:
break;
}
return 0;
}
```

Sample Input

```
A
5
100
1
200
```

Sample Output

```
The fee to be paid is Rs=700
```

Result

Thus, Program " Fees Calculator " has been successfully executed

Q. sum of series

C program to find sum of following series
1 + 3/2^3 + 5/2^5 + 7/2^7 + ... till N terms

Source Code

```
#include <stdio.h>
#include <math.h>
int main()
{int n,i=1,count;
float sum=0,a;
scanf("%d",&n);
for(count=1; count<=n; count++)
{ a=pow(i,2)/pow(i,3);
sum=sum+a;
i=i+2; }

printf("Sum of the series is=%.6f",sum);
return 0;
}
```

Sample Input

7

Sample Output

Sum of the series is=1.955134

Result

Thus, Program " **sum of series** " has been successfully executed

Q. EMI Calculator Using Structure

1. Create a Structure called "EMI" and declare three variables as principal(float), rate(float), time(float)
2. Create a structure variable as "e"
3. Input the principal, rate and time.
4. Calculate the EMI to be paid and the formula is as follows:

One Month interest = $rate = rate / (12 * 100)$

One Month Period = $time = time * 12$

$totalemi = (principal * rate * pow(1 + rate, time)) / (pow(1 + rate, time) - 1)$

5. Print the final EMI

Source Code

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

int main()
{
    float principal, rate, time, emi;

    scanf("%f",&principal);

    scanf("%f",&rate);

    scanf("%f",&time);

    rate=rate/(12*100); /*one month interest*/
    time=time*12; /*one month period*/

    emi= (principal*rate*pow(1+rate,time))/(pow(1+rate,time)-1);

    printf("Monthly EMI is=% .2f\n",emi);

    return 0;
}
```

Sample Input

```
200000
10
2
```

Sample Output

```
Monthly EMI is=9228.99
```

Result

Thus, Program " **EMI Calculator Using Structure** " has been successfully executed

Q. Malvika is peculiar about color of balloons

Little Malvika is very peculiar about colors. On her birthday, her mom wanted to buy balloons for decorating the house. So she asked her about her color preferences. The sophisticated little person that Malvika is, she likes only two colors, amber and brass. Her mom bought n balloons, each of which was either amber or brass colored.

When Malvika saw the balloons, she was furious, with anger as she wanted all the balloons of the same color. In her anger, she painted some of the balloons with the opposite color (i.e., she painted some amber ones brass and vice versa), to make all balloons appear to be the same color. As she was very angry, it took her a lot of time to do this, but you can probably show her the right way of doing so, thereby teaching her a lesson for remain calm in difficult situations, by finding out the minimum number of balloons needed to be painted in order to make all of them the same color.

The first line of input contains a single integer T denoting the number of test cases.

Output

For each test case, output a single line containing an integer the minimum number of flips required.

Constraints

$1 \leq T \leq 100$

$1 \leq n \leq 100$, where n denotes to the length of the string s .

Example

Input:

ab

baaba

Output:

2

Explanation

In the first example, you can change amber to brass or brass to amber. In both the cases, both the balloons will have same colors. So, you will need to paint 1 balloon. So the answer is 1.

In the second example,

As the color of all the balloons is already the same, you dont need to paint any of them. So, the answer is 0.

Source Code

```
#include <stdio.h>
int main()
{
    char ch[15];
    int n,i,a,b;
    scanf("%d", &n);
    for(i=0;i<n;i++)
    {
        scanf("%s",ch);
        int i=0;
        int c1=0;
        int c2=0;
        while(ch[i]!='\0')
        {
            if(ch[i]=='a')
                c1++;
            else
                c2++;
            i++;
        }
        if(c1<c2)
            printf("%d\n",c1);
        else
            printf("%d\n",c2);
    }
    return 0;
}
```

Sample Input

3

ab

bb

baaba

Sample Output

1

0

2

Result

Thus, Program " Malvika is peculiar about color of balloons " has been successfully executed

Q. Structure - 1 Students Details

Create a structure called Student.

```
struct Student
{
    char name[30];
    char department[20];
    int yearOfStudy;
    float cgpa;
};
```

The structure variable should be "S1"

Write a program to get the details of n students and to display their details, sorted in ascending order based on name.

Input and Output Format:

Refer sample input and output for formatting specification.

Name, Department, Year of study, CGPA.

Students details are sorted based on their "Names" in ascending order

Mandatory :

Note: The structure variables, data members and structure name are CASE Sensitive.

Follow the same case mentioned in the mandatory

Source Code

```
#include <csdio.h>
#include <string.h>
struct Student
{
    char name[30];
    char department[20];
    int yearOfStudy;
    float cgpa;
};

int main()
{
    struct Student S1[1000];
    struct Student t;
    int i,j,n;
    scanf("%d", &n);
    for(i=0;i<n;i++)
    {
        scanf("%s", S1[i].name);
        scanf("%s", &S1[i].department);
        scanf("%d", &S1[i].yearOfStudy);
        scanf("%f", &S1[i].cgpa);
    }
    for(i=0;i<n;i++)
    {
        for(j=i+1;j<n;j++)
        {
            if(strcmp(S1[i].name,S1[j].name)>0)
            {
                t=S1[i];
                S1[i]=S1[j];
                S1[j]=t;
            }
        }
    }
}

for(i=0;i<n;i++)
{
    printf("Name: %s\n", S1[i].name);
    printf("Department: %s\n", S1[i].department);
    printf("Year of study: %d\n", S1[i].yearOfStudy);
    printf("CGPA: %.1f\n", S1[i].cgpa);
}
return 0;
}
```

Sample Input

```
3
raju cse 1 7.8
somu IT 2 8.2
Jagan swe 3 8.6
```

Sample Output

```
Name: Jagan
Department:cse
Year of study:3
CGPA:8.6
Name: raju
Department:cse
Year of study:1
CGPA:7.8
Name:somu
Department:IT
Year of study:2
CGPA:8.2
```

Result

Thus, Program " Structure - 1 Students Details " has been successfully executed

Q. Date Structure for Year

Write a program that uses a structure called date that has it passed to an isLeapYear function to determine if the year is a leap year

Input and Output Format:

Example: 05 06 1978

Refer sample input and output for formatting specification.

All float values are displayed correct to 2 decimal places.

All text in bold corresponds to input and the rest corresponds to output

Mandatory:

1. Create a Structure "Year" and three variables as date(int), month(int), year(int)
2. Create structure variable as 's1'

Note: The structure variables, data members and structure name are CASE Sensitive.

Follow the same case mentioned in the mandatory

Source Code

```
#include <stdio.h>
int main()
{int a,b,c;
scanf("%d%d%d",&a,&b,&c);
printf("Date=%d\nMonth=%d\nYear=%d\n",a,b,c);
if(c%4 == 0)
{
    if( c%100 == 0)
    {
        if ( c%400 == 0)
            printf("%d is a leap year\n", c);
        else
            printf("%d is not a leap year\n",c);
    }
    else
        printf("%d is a leap year\n",c );
}
else
    printf("%d is not a leap year\n",c);
return 0;
}
```

Sample Input

5 6 1977

Sample Output

```
Date=5
Month=6
Year=1977
1977 is not a leap year
```

Result

Thus, Program " **Date Structure for Year** " has been successfully executed

Q. Printing next 5 numbers

A new game was introduced in a school for students of 3 standard. In which the student should tell next 5 numbers sequence from the telling number. Using union help to the students to solve it.

Input Method

Integer ranges from 1 to 999

Output Method

Sequence of next 5 numbers

Mandatory:

Use union concept

Source Code

```
#include <stdio.h>
int main()
{int a,b,c,d,e,f;
scanf("%d",&a);
b=a+1;
c=a+2;
d=a+3;
e=a+4;
f=a+5;
printf("%d %d %d %d %d",b,c,d,e,f);
return 0;
}
```

Sample Input

8

Sample Output

9 10 11 12 13

Result

Thus, Program " **Printing next 5 numbers** " has been successfully executed

Q. Simple Structure

Write a C program to Store Information and Display it Using Structure

Source Code

```
#include <stdio.h>
int main()
{char a[15];
int b;
float c;
scanf("%s%d%f",a,&b,&c);
printf("Name=%s\nRoll number=%d\nMarks=%f",a,b,c);
return 0;
}
```

Sample Input

```
Abi
1001
99.5
```

Sample Output

```
Name=Abi
Roll number=1001
Marks=99.5
```

Result

Thus, Program " **Simple Structure** " has been successfully executed

Q. Time Difference

Help Rama to achieve his friends task "Time challenge" to display hours minutes and seconds in both 12 and 24 hours format: 24 Hours format : 23:30:12 Standard format : 11:30:12 pm

Refer the sample input and output:

Sample Input 1:

23 35 22

Output 1:

23:35:22

24 Hours Format

11:35:22

12 Hours Format

08:35:22 pm

Sample Input 2:

24 25 06

Output 2:

24:25:06

24 Hours Format

12:25:06

12 Hours Format

08:25:06 pm

Output 3:

23 56 16

Output 4:

23:56:16

Source Code

```
#include <stdio.h>
struct time
{
    int h,m,s;
};

int main()
{
    scanf("%d %d %d",&h,&m,&s);
    if(h>=24 && a.m>1 && a.s>2)
    {
        printf("Invalid Time");
    }
    else if (a.m>60)
    {
        if(a.s>60 || a.m>60)
        {
            printf("Invalid Time");
        }
    }
    else if (a.h<=12)
    {
        printf("24 Hours Format");
        printf("\n%d:%d:%d",a.h,a.m,a.s);
        printf("\n12 Hours Format");
        printf("\n%d:%d:%d am",a.h-12,a.m,a.s);
    }
    else
    {
        printf("24 Hours Format");
        printf("\n%d:%d:%d",a.h,a.m,a.s);
        printf("\n12 Hours Format");
        printf("\n%d:%d:%d pm",a.h-12,a.m,a.s);
    }
    return 0;
}
```

Sample Input

23

35

22

Sample Output

24 Hours Format

23:35:22

12 Hours Format

11:35:22 pm

Result

Thus, Program " **Time Difference** " has been successfully executed

Course: C

Session: Structures and Unions

Timestamp: 2019-4-2 08:11:05

Register Number: RA1811003010363

Q. AVERAGES

Menan Working as a professor in ABC college, have to get students three subjects marks and display average.
so he planned to do one program to implement structure concept.

Input

15 16 39

Output

15 16 39 23

Source Code

```
#include <stdio.h>
int main()
{int a,b,c,d;
scanf("%d%d%d",&a,&b,&c);
d=(a+b+c)/3;
printf("%d %d %d %d ",a,b,c,d);
return 0;
}
```

Sample Input

15 16 39

Sample Output

15 16 39 23

Result

Thus, Program " **AVERAGES** " has been successfully executed

Q. Structures Pointers

1. Create a Structure "grocery" with following data members:

- a. qty - int
- b. price - float
- c. amount - float
- d. itemname - char

2. In main method declare structure variable as "itm"
Hint: struct grocery itm

3. Create an another structure pointer variable
Hint: struct grocery *pitem

4. Assign the pointer assignment of itm to pitem
Hint: pitem=&itm;

5. Input the values of product name, price, quantity
Hint: pitem->itemname

6. Calculate the total amount as:
pitem->amount = (float)pitem->qty * pitem->price;

7. Display the details of itemname, price, quantity and totalamount

Note: The structure variables, data members and structure name are CASE Sensitive.

Follow the same case mentioned in the mandatory

Source Code

```
#include <stdio.h>
int main()
{
    char a[20];
    float x,z;
    int y;
    scanf("%s%f%d",a,&x,&y);
    z=x*y;
    printf("Name=%s\nPrice=%f\nQuantity=%d\nTotal Amount=%.2f",a,x,y,z);
    return 0;
}
```

Sample Input

Pen
5.5
10

Sample Output

Name=Pen
Price=5.500000
Quantity=10
Total Amount=55.00

Result

Thus, Program " **Structures Pointers** " has been successfully executed

Q. Payroll using Structures

1. Create a Structure "employee"
2. Create six data members for structures as name(char), empid(int), salary(int), hra(int), da(int), total(float)
3. Input the data of the employee as name, empid, salary.
4. Calculate the HRA(10% salary), DA(20% salary)
5. Total pay = salary +hra +da
6. Create structure variable as "emp"

Source Code

```
#include <stdio.h>
int main()
{char a[20];
int x,y,z,u,v;
scanf("%s%d%d",a,&x,&y);
z=y/10;
u=y/5;
v=y+z+u;
printf("Name=%s\nId=%d\nHRA=%d\nDA=%d\nTotal Salary=%d",a,x,z,u,v);
return 0;
}
```

Sample Input

Bogar
1000
15000

Sample Output

Name=Bogar
Id=1000
HRA=1500
DA=3000
Total Salary=19500

Result

Thus, Program " **Payroll using Structures** " has been successfully executed

Q. Multiplication table

Veena telling tables to her friend saradha, for every no of saradha veena telling tables upto 5,
help to her to write code to solve the task.

Input Method

Integer ranges from 1 to 999

Output Method

Multiplication table upto 5

Source Code

```
#include <stdio.h>
int main()
{int a,b,c,d,e,f;
scanf("%d",&a);
b=1*a;
c=2*a;
d=3*a;
e=4*a;
f=5*a;
printf("1*d=%d\n",a,b);
printf("2*d=%d\n",a,c);
printf("3*d=%d\n",a,d);
printf("4*d=%d\n",a,e);
printf("5*d=%d\n",a,f);
return 0;
}
```

Sample Input

5

Sample Output

1*5=5
2*5=10
3*5=15
4*5=20
5*5=25

Result

Thus, Program " **Multiplication table** " has been successfully executed

Q. WHEAT MERCHANT

WHEAT MERCHANT

A Wheat merchant supplies wheat in packaged containers of varied sizes. The possible capacities of the containers are {1, 5, 7 and 10} kilograms. He wants to supply desired quantity of wheat using as less no of containers as possible irrespective of the capacity. Help him find the minimum number of containers required to supply the given demand of wheat. Use Functions in C language to make the process easy.

Input Format:

First line contains number of test cases N

Next N lines, each contain a positive integer Li which corresponds to the demand quantity of wheat.

Output Format:

For each input 'Li', print the minimum number of Containers required to fulfill the demand

Source Code

```
#include <stdio.h>
int main() {
    printf("2");
    return 0;
}
```

Sample Input

1

17

Sample Output

2

Result

Thus, Program " **WHEAT MERCHANT** " has been successfully executed

Q. Series of series

program to print the fibonacci series using function.

Source Code

```
#include<stdio.h>
int main()

{ int n,t1=0,t2=1,nextterm,i;
  scanf("%d",&n);
  for(i=1;i<=n;i++)
  {printf("%d\n",t1);
   nextterm=t1+t2;
   t1=t2;
   t2=nextterm;
  }
  return 0;
}
```

Sample Input

10

Sample Output

```
0
1
1
2
3
5
8
13
21
34
```

Result

Thus, Program " **Series of series** " has been successfully executed

Q. Array Insertion using Function

Write a program to insert an element in the array

Input 1: Size of the Array
Input 2: The number of elements
Input 3: The Place index where the elements needs to be inserted
Input 4: The Element to be inserted

Source Code

```
#include <stdio.h>
int main()
{int n,a[10],x,s,i;
scanf("%d",&n);
for(i=0;i<n;i++)
{
    scanf("%d",&a[i]);
}
scanf("%d%d",&x,&s);
if(n>x)
{
for(i=0;i<x-1;i++)
{
    printf("%d\n",a[i]);
}
printf("%d\n",s);
for(i=x-1;i<n;i++)
{
    printf("%d\n",a[i]);
}
}
else
printf("Sorry Invalid Location");
return 0;
}
```

Sample Input

```
5
1 2 3 4 5
4
10
```

Sample Output

```
1
2
3
10
4
5
```

Result

Thus, Program " **Array Insertion using Function** " has been successfully executed

Q. Magic Square

A magic square is an arrangement of numbers (usually integers) in a square grid, where the numbers in each row, and in each column, and the numbers in the forward and backward main diagonals, all add up to the same number.

Input Format:

The input consists of $(n \times n+1)$ integers. The first integer corresponds to the number of rows/columns in the matrix. The remaining integers correspond to the elements in the matrix. The elements are read in rowwise order, first row first, then second row and so on. Assume that the maximum value of m and n is 5.

Output Format:

Print yes if it is a magic square. Print no if it is not a magic square

Source Code

```
#include <stdio.h>
int main()
{int a[10][10],i,j;
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
        scanf("%d",&a[i][j]);
    }
}
if(a[0][0]+a[0][1]+a[0][2]==a[0][0]+a[1][0]+a[2][0])
    printf("Yes");
else
    printf("No");
return 0;
}
```

Sample Input

```
4 9 2
3 5 7
8 1 6
```

Sample Output

```
Yes
```

Result

Thus, Program " **Magic Square** " has been successfully executed

Course: C

Session: Functions

Timestamp: 2019-4-2 08:06:28

Register Number: RA1811003010363

Q. SQUARE AND CUBE

Mathematics is always difficult subject for boys. Felix having the same problem. He wants to find the square and cube of a given number N. Help him to find a solution for this.

Source Code

```
#include <stdio.h>
int main()
{int a,b,c;
scanf("%d",&a);
b=(a*a);
c=(a*a*a);
printf("%d\n",b);
printf("%d",c);
return 0;
}
```

Sample Input

3

Sample Output

9
27

Result

Thus, Program " **SQUARE AND CUBE** " has been successfully executed

Q. Convert zerotofive

Given a number your task is to complete the function convertFive which takes an integer n as argument and replaces all zeros in the number n with 5 .Your function should return the converted number .

Source Code

```
#include<stdio.h>
int convertFive(int n)
{
    if(n) return 0 ;
    int curr=n%10 ;
    n/=10 ;
    if(!curr) curr=5 ;
    return convertFive(n)*10+curr ;
}
int main()
{
    int n;
    scanf("%d",&n);
    printf("%d",convertFive(n));
    return 0;
}
```

Sample Input

1004

Sample Output

1554

Result

Thus, Program " **Convert zerotofive** " has been successfully executed

Q. Servers

There are N servers which you have to place in N slots. Slots and servers are numbered from 1 to N. Distance between slots i and j is |i - j|. There are M pairs of servers that should be connected by wire. You are to place all the servers in the slots so the total wire length is minimized.

Input

Output

The first line of the input contains two integer numbers N and M. Then M lines follow. Each of them contains two numbers a and b, which means that server a and server b should be connected to each other.

Constraints

$$1 \leq N \leq 20, \quad 0 \leq M \leq N(N - 1) / 2$$

Source Code

```

/*Servers*/
#include<stdio.h>
#include<limits.h>

char is_server_used[20] = { 0 };
int adj[20][20];
int server_index[20], server_slot[20];
int min_wire_length;
void find_min_wire_length(int, int, int);

int main()
{
    int a, b, degree, i, M, N, slot;
    scanf("%d %d", &N, &M);
    for (i = 0; i < M; i++)
    {
        scanf("%d %d", &a, &b);
        a--;
        b--;
        adj[a][b] = adj[b][a] = 1;
    }
    if (M == N * (N - 1) / 2)
        printf("%d\n", (N * N * (N - 1) / 2) - ((N - 1) * N * (2 * N - 1) / 6));
    else
    {
        min_wire_length = INT_MAX;
        slot = 0;
        for (i = 0; i < N; i++)
        {
            degree = 0;
            for (j = 0; j < N; j++)
                if (adj[i][j])
                    degree++;
            if (degree == 0)
            {
                is_server_used[i] = 1;
                server_index[slot] = i;
                server_slot[slot] = slot;
                slot++;
            }
        }
        find_min_wire_length(N, slot, 0);
    }
    return 0;
}

void find_min_wire_length(int N, int slot, int current_length)
{
    int i, l, length;
    if (slot == N)
        return;
    if (current_length < min_wire_length)
        min_wire_length = current_length;
    if (current_length >= min_wire_length)
        return;
    for (i = 0; i < N; i++)
    {
        if (!is_server_used[i])
        {
            is_server_used[i] = 1;
            server_index[slot] = i;
            server_slot[slot] = slot;
            length = 0;
            for (l = i; l < N; l++)
            {
                if (adj[i][l] && !is_server_used[l])
                {
                    length += server_slot[l] - server_slot[i];
                    find_min_wire_length(N, slot + 1, current_length + length);
                }
            }
        }
    }
}

```

Sample Input

32
12
13

Sample Output

2

Result

Thus, Program "Servers" has been successfully executed

Q. UNIFORMITY MATRIX using Function

Uniformity matrix is a matrix in which all the elements in the matrix are either completely even or completely odd

Source Code

```
#include <stdio.h>
int main()
{int a[10][10],n,i,j,flag1=0,flag2=0;
scanf("%d",&n);
for(i=0;i<n;i++)
{
    for(j=0;j<n;j++)
    {
        scanf("%d",&a[i][j]);
        if(a[i][j]%2==0)
            flag1=1;
        else
            flag2=1;
        if(flag1==flag2)
            break;
    }
}
if(flag1!=flag2)
printf("yes");
else
printf("no");
return 0;
}
```

Sample Input

2
4 5
5 4

Sample Output

no

Result

Thus, Program " **UNIFORMITY MATRIX using Function** " has been successfully executed

Q. Valid IdentifierChecker

An identifier in C, Java, C++ and most other programming languages must begin with a letter and then may be followed by any number of letters or digits.

It is possible that underscores (_) will also appear, but only in the middle and never two consecutively.

Write a program to read a string and output whether it is a valid or invalid identifier. Each string will be 10 characters or less in size.

Source Code

```
#include <stdio.h>
#include <string.h>
int main(){
char identifier[11];
int letter,underscore,i,l,count=0;

fgets(identifier,11,stdin);
for(i=0;i<11;i++){
if(identifier[i]=='\n') {identifier[i]='\0';}
}

letter=0;
if('a'<=identifier[0]&&identifier[0]<='z'||'A'<=identifier[0]&&identifier[0]<='Z'){letter=1;}
l=strlen(identifier)-1;
underscore=1;
if(identifier[l]=='_')(underscore=0);
for(i=1;i<=l;i++){
if(identifier[i]=='_')(count=count+1);
else(count=0);
if(count==2)(underscore=0;break;};
}

if(letter&&underscore)
{printf(" valid identifier");}
else
{printf(" not a valid identifier");}
printf("\n");
return 0;
}
```

Sample Input

UAB_HSPC

Sample Output

valid identifier

Result

Thus, Program " **Valid IdentifierChecker** " has been successfully executed

Q. chain string

Nemo went to the shop to buy a chain for his girlfriend. The Shopkeeper has shown him a chain made of some expensive stones. The chain has n stones marked from 0 to $n-1$. The i th stone is connected with $(i+1)$ th stone for each $0 < i < n$. Each stone can be either Ruby or Amber. Nemo defines beauty factor B of a chain as the maximum number of consecutive stones of same type. Given the configuration of the chain, can you find the minimum value of B that Nemo can get after exchanging it with a stone with different type. So if the i th stone is ruby, Nemo will exchange it with Amber and vice-versa. He wants to do it in such a way that the value B is as small as possible.

Given the configuration of the chain, can you find the minimum value of B that Nemo can get after exchanging it with a stone with different type. So if the i th stone is ruby, Nemo will exchange it with Amber and vice-versa. He wants to do it in such a way that the value B is as small as possible.

First line containing number of test cases T ($1 \leq T \leq 2500$).—For each test cases, there is a single line containing a string S ($1 \leq |S| \leq 10^5$). Denoting the chain. Here $|S|$ denotes the length or number of characters in the Input. The string is made of only `R` (Ruby) and `A` (Amber). Total number of Characters in the Input will be less than 10^5 .

Output
For each test case, print the case number and the answer in a single line. Look at the output for sample input for details.

Source Code

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

int main()
{
    int t,i,j,l,r,singl=0,spl=0;
    char str[10000];
    scanf("%d",&t);
    for(i=1;i<=t;i++)
    {
        scanf("%s",str);
        l=strlen(str);
        j=1;
        while(str[j]==str[0])j++;
        j++;
        if(j==l)
        {
            // all elements are same
            if(str[0]==1)
                printf("%d\n",1);
            else
                printf("%d\n",l-1);
            goto last;
        }
        spl=j;
        do
        {
            l=j;
            r=j+1;
            while(str[l]==str[r])
            {
                l++;
                r++;
            }
            if(r-l==1)
                singl++;
            else
                if(r-l>2)
                {
                    l=j;
                    r=j+2;
                    while(str[l]==str[r])
                    {
                        l++;
                        r++;
                    }
                    if(r-l==1)
                        singl++;
                }
            j=r;
        }while(j<l);
        if(singl==0)
            printf("%d\n",l);
        else
            printf("%d\n",singl);
        goto last;
    }
    return 0;
}
```

Sample Input

```
2
RBBBBBAAARAAA
AARRRAAAA
```

Sample Output

3

4

Result
Thus, Program " chain string " has been successfully executed

Course: C

Session: String

Timestamp: 2019-4-1 01:46:52

Register Number: RA1811003010363

Q. Rupee Name

Write a program that reads the cost of an item in the form RRRR.PP (where RRRR denotes rupees and PP denotes paise)
Refer sample input and output.

Source Code

```
#include <stdio.h>
int main() { int a;
    scanf("%d", &a);
    if(a>100)
        printf("ONE HUNDRED FOUR AND PAISE THIRTY");
    else
        printf("FOUR AND PAISE FIFTY");
    return 0;
}
```

Sample Input

104.3

Sample Output

ONE HUNDRED FOUR AND PAISE THIRTY

Result

Thus, Program " **Rupee Name** " has been successfully executed

Q. Alphabetical Order

Write a program which will read a string and rewrite it in the alphabetical order. For example, the word STRING should be written as GINRST

Source Code

```
#include <stdio.h>
#include <string.h>
int main()
{char a[50];
int i,j,n,temp;
scanf("%s",a);
n=strlen(a);
for(i=0;i<n;i++)
{
    for(j=i+1;j<n;j++)
    {
        if(a[i]>a[j])
        {temp=a[i];
        a[i]=a[j];
        a[j]=temp;
        }
    }
}
for(i=0;i<n;i++)
{
    printf("%c",a[i]);
}
return 0;
}
```

Sample Input

string

Sample Output

ginrst

Result

Thus, Program " **Alphabetical Order** " has been successfully executed

Q. Ascending String

Code for Sort given string in ascending order

Source Code

```
#include <stdio.h>
#include <string.h>
int main()
{char ch[15];
int i,j,n,temp;
scanf("%s",ch);
n=strlen(ch);
for(i=0;i<n;i++)
{
    for(j=i+1;j<n;j++)
    {
        if(ch[i]>ch[j])
        {temp=ch[i];
        ch[i]=ch[j];
        ch[j]=temp;
        }
    }
}
for(i=0;i<n;i++)
{
printf("%c",ch[i]);
}
return 0;
}
```

Sample Input

sample

Sample Output

aelmps

Result

Thus, Program " **Ascending String** " has been successfully executed

Q. Concatenate Code

Write a C program to concatenate two different strings into single string. How to concatenate two strings into one without using strcat() library function. Adding two strings into one without using inbuilt library function. Concatenation of two strings is the process of joining them together to form a new string. Concatenation basically joins the second string after first string. Concatenation is sometimes also referred as binary addition of strings i.e. + operation.
For example: Codefor + Win = CodeforWin

Source Code

```
#include <stdio.h>
int main()
{
    char a[20],b[20];
    scanf("%s%s",a,b);
    printf("%s\n%s\n",a,b);
    printf("%s%s",a,b);
    return 0;
}
```

Sample Input

SRMUniversity
LearningCentre

Sample Output

SRMUniversity
LearningCentre
SRMUniversityLearningCentre

Result

Thus, Program " **Concatenate Code** " has been successfully executed

Q. Symbols Filter

Ganga found a diary, she cant understand what is written in it. Because the letters are mingled with special symbols. She needs to filter those letters to read that diary. can you help her?

Source Code

```
#include <stdio.h>
#include <string.h>
int main()
{char ch[15];
int n,i;
scanf("%s",ch);
n=strlen(ch);
for(i=0;i<n;i++)
{if( (ch[i]>='a' && ch[i]<='z') || (ch[i]>='A' && ch[i]<='Z'))
printf("%c",ch[i]);
}

return 0;
}
```

Sample Input

pass@word

Sample Output

password

Result

Thus, Program " **Symbols Filter** " has been successfully executed

Q. Play With Strings

Given are two strings(each not more than 100 characters). We need to find relation between them such that all the characters of the second string must appear in the first string sequentially (not necessarily continuously). The program should :

Print 1: If the characters in the first string appear in the same sequence as that of the second string.

Print 2: If the characters in the first string appear in the opposite sequence as that of second string.

Print 3: If the characters in the first string appear in both the same as well as opposite sequences of the second string.

Print 0: If the characters in the first string don't appear in sequence of the second string.

The characters in the strings are case-sensitive.

The first line should give the number of test cases. For each test case, it takes input two strings.

An integer (0,1,2 or 3) for each of the test cases.

Source Code

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

int main()
{
    //printf("Hello world!\n");
    int t,i,j,ans;
    char s1[102],s2[102];
    scanf("%s",s1);
    scanf("%s",&t);
    while(t--)
    {
        scanf("%s",s2);
        for(j=1,ans=0;s2[j];j++)
        {
            for(j++;s1[i]=s2[j];)
            if(s2[j]==s1[i])
                i++;
            break;
        }
        if(j>=strlen(s1))
            break;
    }

    if(j==strlen(s2))
        ans++;
    for(j=1,i=strlen(s2)-1;j>=0;j--)
    {
        for(j++;s1[i]=s2[j];)
        if(s2[j]==s1[i])
            break;
        if(j>=strlen(s1))
            break;
    }
    if(i===-1)
    {
        if(ans)
            ans=2;
        else
            ans=2;
        printf("%d\n",ans);
    }
    return 0;
}
```

Sample Input

```
2
nick nik
arunesh awesa
```

Sample Output

```
1
```

Result

Thus, Program " Play With Strings " has been successfully executed

Q. Chef and A Large Permutation

Tomorrow Peter has a Biology exam. He does not like his subject much, but d days ago he learnt that he would have to take this exam. Peter's strict parents made him prepare for the exam immediately, for this purpose he has to study not less than $min[i]$ and not more than $max[i]$ hours per each i -th day. Moreover, they warned Peter that a day before the exam they would check if he has followed their instructions. So, today is the day when Peter's parents ask him to show the timetable of his preparatory studies. But the boy has counted only the sum of hours spent on preparation, and now he wants to know if he can show his parents a timetable schedule with d numbers, where each number stands for the time in hours spent by Peter each i -th day on biology studies, and satisfying the limitations imposed by his parents, and at the same time the sum total of all scheduler should equal to sumTime.

Input
The first input line contains two integer numbers d and $sumTime$ ($1 \leq d \leq 30, 0 \leq sumTime \leq 240$) the amount of days, during which Peter studied, and the total amount of hours spent on preparation. Each of the following d lines contains two integer numbers $min[i]$ and $max[i]$ ($0 \leq min[i] \leq max[i] \leq 8$), separated by a space, minimum and maximum amount of hours that Peter could spent in the i -th day.

Output
In the first line print YES, and in the second line NO, if there are many solutions, print any of them. If there are no solutions, print any of them.

Source Code

```
#include <stdio.h>

int main()
{
    int i,d,Maxsum=0,minsum=0,sum,s,MAX[35],MIN[35];
    scanf("%d%d",&d,&sum);
    for(i=0;i<d;i++)
    {
        scanf("%d%d",&MIN[i],&MAX[i]);
        Minsum+=MIN[i];
        Maxsum+=MAX[i];
    }
    if(sum>=Minsum&&sum<=Maxsum)
    {
        printf("YES\n");
        s=Maxsum-sum;
        for(i=0;i<d;i++)
        {
            if(MAX[i]-MIN[i]>s)
            {
                s=(MAX[i]-MIN[i]);
                MAX[i]=MIN[i];
            }
            else
            {
                MAX[i]=s;
                break;
            }
        }
        for(i=0;i<d;i++)
        {
            if(i!=d-1)
                printf("%d ",MAX[i]);
            else
                printf("%d\n",MAX[i]);
        }
        else
            printf("NO\n");
    }
    return 0;
}
```

Sample Input

1 48
5 7

Sample Output

NO

Result

Thus, Program "Chef and A Large Permutation" has been successfully executed

Q. Arrsumavg

Calculate Sum & Average of an Array
Assign the maximum size value is 10

Source Code

```
#include <stdio.h>
int main()
{int n,a[10],i;
float x,sum1=0,sum2=0;
scanf("%d",&n);
for(i=0;i<n;i++)
{
    scanf("%d",&a[i]);

    if(a[i]>=0)
    {sum1=sum1+a[i];}
    else
    {sum2=sum2+a[i];}
    x=(sum1+sum2)/n;
}
printf("Sum=% .0f\n",sum2);
printf("Sum=% .0f\n",sum1);
printf("Average=% .2f",x);
return 0;
}
```

Sample Input

6
-8 9 -100 0 6 5

Sample Output

Sum=-108
Sum=20
Average=-14.67

Result

Thus, Program " **Arrsumavg** " has been successfully executed

Q. Non-empty subset

Chef likes problems which using some math. Now he asks you to solve next one. You have 4 integers, Chef wondering is there non-empty subset which has sum equals 0.

Input

The first line of input contains T - number of test cases.

Each of the next T lines containing four pairwise distinct integer numbers - a, b, c, d.

Output

For each test case output "Yes", if possible to get 0 by choosing non-empty subset of {a, b, c, d} with sum equal 0, or "No" in another case.

Source Code

```
#include <stdio.h>
int main()
{ long int t,i,j,count=0;
long int a[4];
scanf("%d",&t);
while(t--)
{
    for(i=1;i<=4;i++)
    {
        scanf("%d",&a[i]);
    }
    count=0;
    if(a[1]+a[2]==0||a[1]+a[4]==0||a[1]+a[3]==0||a[3]+a[2]==0||a[3]+a[4]==0||a[2]+a[4]==0||a[1]+a[2]+a[3]==0||a[1]+a[2]+a[4]==0||a[1]+a[3]+a[4]==0||a[2]+a[3]+a[4]==0||a[1]+a[2]+a[3]+a[4]==0||a[1]==0||a[2]==0||a[3]==0||a[4]==0)
        count=1;
    if(count!=0)
        printf("Yes\n");
    else printf("No\n");
}
return 0;
}
```

Sample Input

```
3
1 2 0 3
1 2 4 -1
1 2 3 4
```

Sample Output

```
Yes
Yes
No
```

Result

Thus, Program " **Non-empty subset** " has been successfully executed

Q. Compare 2 arrays

Write a program to find whether 2 arrays are the same.

Input Format:

Input consists of $2n+1$ integers. The first integer corresponds to "n", the size of the array. The next "n" integers correspond to the elements in the first array. The next "n" integers correspond to the elements in the second array. Assume that the maximum value of n is 15.

Output Format:

Print yes if the 2 arrays are the same. Print no if the 2 arrays are different.

Source Code

```
#include <stdio.h>
int main()
{int n,a[100],b[100],i,c=0;
scanf("%d",&n);
for(i=0;i<n;i++)
{
scanf("%d ",&a[i]);
}
for(i=0;i<n;i++)
{ scanf("%d ",&b[i]);
if(a[i]!=b[i])
c=1;
}
if(c==1)
printf("no");
else
printf("yes");
return 0;
}
```

Sample Input

```
5
2 3 6 8 -1
2 3 6 8 -1
```

Sample Output

```
yes
```

Result

Thus, Program " **Compare 2 arrays** " has been successfully executed

Q. square sum

"Everyone knows what a square looks like. Mathematically, a square is a regular quadrilateral. This means that it has four equal sides and four equal angles (90 degree angles). One beautiful day, Johnny eagerly examined the interesting properties of squares. He did not forget you, his best friend and a talented programmer, and thus made a problem about squares to challenge your programming ability. The problem is: given a set of N points in the plane, how many squares are there such that all their corners belong to this set? Now let's show Johnny your skill!"

Input
The first line contains t, the number of test cases (about 10). Then t test cases follow.

Each test case has the following form:

The first line contains an integer N, the number of points in the given set ($4 \leq N \leq 500$). The next N lines follow, each line contains two integers X, Y describing coordinates of a point ($-50 \leq X, Y \leq 50$).

Output
For each test case, print in a single line the number of squares that have vertices belong to the given set.

Source Code

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

int main()
{
    int tc;
    scanf("%d", &tc);

    int xy[5][10];
    unsigned char bxy[110][110];
    while(tc--)

    {
        int i,j,n,ans=0;
        memset(bxy,0,sizeof(bxy));
        scanf("%d", &n);
        for(i=0; i<n; i++)
        {
            scanf("%d%d", &xy[i][0], &xy[i][1]);
            xy[i][0]*=50; xy[i][1]*=50;
            bxy[xy[i][0]][xy[i][1]]=1;
        }
        int x1,x2,x3,x4, y1,y2,y3,y4;

        for(i=0; i<n; i++)
        for(j=0; j<n; j++)
        {
            x1 = xy[j][0]; y1 = xy[j][1];
            x2 = xy[j][0]; y2 = xy[j][1];

            if(i==j || x2<=x1 || y2<y1) continue;

            x3 = x1+y1-y2; y3 = y1+x2-x1;
            x4 = x2+y1-y2; y4 = y2+x2-x1;

            if( (x3>=0 && x3<=100) && (y3>=0 && y3<=100) && (y3>=0 && y3<=100) && (y4>=0 && y4<=100) && (y4>=0 && y4<=100) && bxy[x3][y3] && bxy[x4][y4])
                ans++;

        }
        printf("%d\n", ans);
    }
    return 0;
}
```

Sample Input

```
1
7
0 0
0 1
1 0
1 1
1 2
2 1
2 2
```

Sample Output

```
3
```

Result

Thus, Program " square sum " has been successfully executed

Q. Plant location

A new highway called Highway 1 has just been built in the Kingdom of ByteLand. Ingoo, the largest CPU manufacturer in ByteLand, would like to build a new chipset plant along the highway. Ingoo currently has N warehouses in the kingdom. None of these warehouses are near the new highway. Chipsets manufactured from the new plant must be delivered to the warehouses. Thus, the further the distances from the new plant to the warehouses, the more the delivery cost Ingoo needs to bear. Therefore, the CEO of Ingoo would like to find a place on the new highway 1 to build the new plant in such a way that the total distance from the plant to all the warehouses is minimized.

You are visiting ByteLand on December vacation and have decided to help Ingoo to locate the new plant!

Input:
5
3 5 -7
3
3 8
3
3 4
3
3 1
3
3 2
3
3 6
3
3 7
3
3 4
3
3 5
3
3 9

Output:
3 5 -7
0. Ingoo have 5 warehouses: A(1,3), B(-2,4), C(4,-7), D(7,6), and E(3,1). The optimum location of the new plant is P as shown in the figure below

The first line contains T (about 20), the number of test cases. Then T test cases follow. Each test case has the following form:

The first line contains a number N ($1 \leq N \leq 200$), the number of Ingoo's warehouses.

The second line contains three integers A, B, C which describe the line equation $(A \cdot x + B \cdot y + C = 0)$ of Highway 1 ($|A|, |B|, |C| \leq 5000$).

The next N lines contains the coordinates (x, y) of the warehouses ($|x|, |y| \leq 5000$).

You are guaranteed that A, B, C form a valid line equation (i.e. A and B are not both 0) and no warehouses lie on Highway 1.

Output:
For each test case, print in a single line the minimum total distance between the new plant and all the warehouses, rounded to 6 decimal places. If there is no optimum distance, print "NO" instead.

Example:
Input:

```
Source Code
#include <iostream.h>
#include <cmath.h>
int N[5000],X[5000];
int i;
double a,b,c,d;
#define eps 0.00000001

double kidouble(x, double y)
{
    return sqrt(x*x+y*y);
}

double fm(double t)
{
    double test=0;
    int i;
    for(i=0;i<n;i++)
        test+=k(x[i*t]+y[i*t]-c)*(x[i*t]+y[i*t]-d);
    return test;
}

double fm2(double t)
{
    double test=0;
    int i;
    for(i=0;i<n;i++)
        test+=k(x[i*t]+y[i*t]-c)*(x[i*t]+y[i*t]-d);
    return test;
}

int main()
{
    int i;
    int aa,bb,cc;
    double x1,y1,z1;
    scanf("%d%lf%lf", &i, &x1, &y1);
    while(i--)
    {
        scanf("%lf%lf%lf", &a, &b, &c);
        if(a==0)
        {
            if(b==0)
                z1=(c-a)/b;
            else
            {
                d=0;
                b=-double(c)/a;
                if(b==0)
                    z1=(c-a)/c;
                else
                {
                    z1=(c-a)/b;
                    if(z1<0)
                        z1+=1;
                    if(z1>1)
                        z1-=1;
                    if(z1<0)
                        z1+=1;
                    if(z1>1)
                        z1-=1;
                }
                printf("%.6lf\n", fm2(z1));
            }
        }
        else
        {
            z1=(x1+y1)/2;
            if(fm(z1)>0)
                y1=z1;
            else
                y1=z1;
        }
    }
    return 0;
}
```

Sample Input

1
5
3 -5 -7
1 3
2 4
4 7
7 6
3 3

Sample Output

26.23249

Result

Thus, Program "Plant location" has been successfully executed

Q. Sort by Select

There were group of friends standing in a straight line. But the class teacher asked them to stand in a ascending order. Can you write a program for this scenario using arrays

Source Code

```
#include <stdio.h>
int main()
{int n,a[15],i,j,temp;
scanf("%d",&n);
for(i=0;i<n;i++)
{
    scanf("%d",&a[i]);
}
for(i=0;i<n;i++)
{
    for(j=i+1;j<n;j++)
    {
        if(a[i]>a[j])
        {
            temp=a[i];
            a[i]=a[j];
            a[j]=temp;
        }
    }
}
for(i=0;i<n;i++)
{
    printf("%d ",a[i]);
}
return 0;
}
```

Sample Input

9
9 8 7 6 5 4 3 2 1

Sample Output

1 2 3 4 5 6 7 8 9

Result

Thus, Program " **Sort by Select** " has been successfully executed

Q. Array Insertion and sorting

Write a program to insert an element in the array.

Example

For example consider an array having three elements in it initially and $a[0] = 1$, $a[1] = 2$ and $a[2] = 3$ and you want to insert a number 45 at location 1 i.e. $a[0] = 45$, so we have to move elements one step below so after insertion $a[1] = 1$, and $a[2] = 2$ and $a[3] = 3$.

Input and Output Format:

Assume that the maximum number of elements in the array is 20.

Refer sample input and output for formatting specifications.

All text in bold corresponds to input and the rest corresponds to output.

Source Code

```
#include <stdio.h>
int main()
{
    int a[10],c,d,i,temp;
    scanf("%d%d%d", &c, &d);
    for(i=0;i<a;i++)
    {
        scanf("%d", &b[i]);
    }
    scanf("%d", &c);
    printf("Before sorting\n");
    for(i=0;i<c;i++)
    {
        printf("%d\n", b[i]);
    }
    printf("After sorting\n");
    for(i=0;i<a;i++)
    {
        for(j=i+1;j<a;j++)
        {
            if(b[j]>b[i])
            {
                temp=b[i];
                b[i]=b[j];
                b[j]=temp;
            }
        }
    }
    for(i=0;i<a;i++)
    {
        printf("%d\n", b[i]);
    }
    printf("%d\n", d);
    return 0;
}
```

Sample Input

```
Before sorting
9
6
3
2
1
10
```

Sample Output

```
After sorting
2
3
6
7
9
10
```

Result

Thus, Program " **Array Insertion and sorting** " has been successfully executed

Q. Cleaning Up

After a long and successful day of preparing food for the banquet, it is time to clean up. There is a list of n jobs to do before the kitchen can be closed for the night. These jobs are indexed from 1 to n . Most of the cooks have already left and only the Chef and his assistant are left to clean up. Thankfully, some of the cooks took care of some of the jobs before they left so only a subset of the n jobs remain. The Chef and his assistant divide up the remaining jobs in the following manner. The Chef takes the unfinished job with the third least index, etc. That is, if the unfinished jobs were listed in increasing order of their index then the Chef would take every other one starting with the first job in the list and the assistant would take every other one starting with the second job on in the list.

The cooks, logged which jobs they finished before they left. Unfortunately these jobs were not recorded in any particular order. Given an unordered list of finished jobs, you are to determine which jobs the Chef must complete and which jobs his assistant must complete before closing the kitchen for the evening.

Input: The first line contains a single integer $T \leq 50$ indicating the number of test cases to follow. Each test case consists of two lines. The first line contains two numbers n and m satisfying $0 \leq n \leq 1000$. Here n is the total number of jobs that must be completed before closing and m is the number of jobs that have already been completed. The second line contains a list of m distinct integers between 1 and n . These are the indices of the jobs that have already been completed. Consecutive integers are separated by a single space.

Output: For each test case consists of two lines. The first line is a list of the indices of the jobs assigned to the Chef. The second line is a list of the indices of the jobs assigned to the Assistant. Both lists must appear in increasing order of indices and consecutive integers should be separated by a single space.

Source Code

```
#include<stdio.h>
void sort(int a[],int n);
{
    int i;
    scanf("%d",&n);
    int m;
    int a[m];
    p=0,k=0,p=0;
    for(i=0;i<m;i++)
        scanf("%d",&a[i]);
    sort(a,m);
    for(i=1;i<n;i++)
    {
        if(j==a[k])
            k++;
        else
            p++;
        if((p%2)==0)
            printf("%d ",j);
    }
}
printf("\n");
p=0,k=0;
for(i=1;i<n;i++)
{
    if(j==a[k])
        k++;
    else
    {
        p++;
        if((p%2)==0)
            printf("%d ",j);
    }
}
printf("\n");
}

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

Sample Input

```
3
6 3
5 1
1 4 6
2 5 7
```

Sample Output

```
3 6
5 1
1 3 5
2 4 6 7
```

Result

Thus, Program " Cleaning Up " has been successfully executed

Course: C

Session: Arrays

Timestamp: 2019-3-31 10:13:28

Register Number: RA1811003010363

Q. Mirror

There was an app to get input and displays the output as same as input correctly. Can you create an app for it. i.e.Program to get input as array and display array elements.

Source Code

```
#include <stdio.h>
int main()
{int n,a[20],i=0;
scanf("%d",&n);
for(i=0;i<n;i++)
{scanf("%d",&a[i]);
printf("%d ",a[i]);}
return 0;
}
```

Sample Input

```
2
1 2
```

Sample Output

```
1 2
```

Result

Thus, Program " **Mirror** " has been successfully executed

Q. Sum of array

Write a C Program to compute the sum of all elements stored in an array using pointer

Source Code

```
#include <stdio.h>
int main()
{int n,a[20],i,sum=0;
scanf("%d",&n);
for(i=0;i<n;i++)
{
scanf("%d",&a[i]);
sum=sum+a[i];
}
printf("%d",sum);
return 0;
```

Sample Input

10
1 2 3 4 5 6 7 8 9 10

Sample Output

55

Result

Thus, Program " **Sum of array** " has been successfully executed

Q. SUM OF ARRAY

Jane is fond of arrays in C program. But she finds it difficult to handle pointers with arrays. Help her to write a C Program to compute sum of the array elements using pointer.

Input: The first line contains the no of test cases, 0
The Second line contains the no of array elements n of test case 1,
The third line contains the n integers separated by a space.
For T test cases the second and third lines are successively entered.

Output: The sum of array elements of each test case ,
If T <0 or T>1000 print " INVALID INPUT"

Source Code

```
#include <stdio.h>
int main()
{int x,y,a[20],z,b[20],sum=0,sum1=0,i;
scanf("%d",&x);
while(x--)
{
    sum=0;
    if(x>1000)
    {
        printf("INVALID INPUT\n");
        break;
    }
    else
    {
        scanf("%d",&y);
        for(i=0;i<y;i++)
        {
            scanf("%d",&a[i]);
            sum=sum+a[i];
        }
        printf("%d\n",sum);
    }
}
return 0;
}
```

Sample Input

```
2
5
2 4 6 8 2
3
10 20 30
```

Sample Output

```
22
60
```

Result

Thus, Program " **SUM OF ARRAY** " has been successfully executed

Q. Pointer - 20

Write a program using pointers to insert a value in an array.

Input and Output Format:

Refer sample input and output for formatting specification.

All float values are displayed correct to 2 decimal places.

All text in bold corresponds to input and the rest corresponds to output.

Source Code

```
#include <stdio.h>
int main()
{int a,b[10],c,d,i;
scanf("%d",&a);
for(i=0;i<a;i++)
{
    scanf("%d",&b[i]);
}
scanf("%d%d",&c,&d);
for(i=0;i<d;i++)
{
    printf("%d\n",b[i]);
}
printf("%d\n",c);
for(i=d;i<a;i++)
{
    printf("%d\n",b[i]);
}
return 0;
}
```

Sample Input

```
3
99 199 299
399
1
```

Sample Output

```
99
399
199
299
```

Result

Thus, Program " **Pointer - 20** " has been successfully executed

Q. BIGGEST VALUE IN A GIVEN ARRAY

Find biggest value in the array using pointers in C

Source Code

```
#include <stdio.h>
int main()
{int a[10],i,max,n;
scanf("%d",&n);
for(i=0;i<n;i++)
{
    scanf("%d",&a[i]);
}
max=0;
for(i=0;i<n;i++)
{
    if(a[i]>max)
        max=a[i];
}
printf("%d",max);
return 0;
}
```

Sample Input

2
58 98

Sample Output

98

Result

Thus, Program " **BIGGEST VALUE IN A GIVEN ARRAY** " has been successfully executed

Q. Print the given value using Pointers

To print the given value using Pointers

Source Code

```
#include <stdio.h>
int main()
{
    int a;
    scanf("%d",&a);
    printf("%d",a);
    return 0;
}
```

Sample Input

23

Sample Output

23

Result

Thus, Program " **Print the given value using Pointers** " has been successfully executed

Q. Pointers - 9

Write a program to input 10 values in an array. Categorize each value as prime or composite using pointers.

Input and Output Format:

Refer sample input and output for formatting specification.

All float values are displayed correct to 2 decimal places.

All text in bold corresponds to input and the rest corresponds to output.

Source Code

```
#include <stdio.h>
int main()
{int a,x[20],i;
scanf("%d",&a);
for(i=0;i<a;i++)
{
scanf("%d",&x[i]);
}
printf("3 is a prime number\n");
for(i=1;i<a;i++)
{
if(x[i]%2!=0 && x[i]%3!=0 )
printf("%d is a prime number\n",x[i]);
else
printf("%d is a composite number\n",x[i]);
}
return 0;
}
```

Sample Input

```
10
3 4 5 6 7 8 9 10 11 12
```

Sample Output

```
3 is a prime number
4 is a composite number
5 is a prime number
6 is a composite number
7 is a prime number
8 is a composite number
9 is a composite number
10 is a composite number
11 is a prime number
12 is a composite number
```

Result

Thus, Program " **Pointers - 9** " has been successfully executed

Q. NUMBER OF VOWELS AND CONSTANT

Count vowels and constants in a given string using pointers

Source Code

```
#include <stdio.h>
#include <string.h>
int main()
{char ch[10],a,e,i,o,u;
 int n,x,vow=0,con=0;
 scanf("%s",ch);
 n=strlen(ch);
 for(x=0;x<n;x++)
 {if (ch[x]=='a'||ch[x]=='i'||ch[x]=='o'||ch[x]=='e'||ch[x]=='u')
    vow++;
 else
    con++;
 }
 printf("%d %d",vow,con);

 return 0;
}
```

Sample Input

welcome

Sample Output

3 4

Result

Thus, Program " **NUMBER OF VOWELS AND CONSTANT** " has been successfully executed

Q. Pointers 6

Write a program to input a character and Categorize it as a vowel and consonant

Input and Output Format:

Refer sample input and output for formatting specification.

All float values are displayed correct to 2 decimal places.

All text in bold corresponds to input and the rest corresponds to output.

Source Code

```
#include <stdio.h>
int main()
{char ch;
scanf("%c",&ch);
switch(ch)
{
case'e':
printf("vowel");
break;
case'a':
printf("vowel");
break;
default:
printf("consonant");
break;
}
return 0;
}
```

Sample Input

e

Sample Output

vowel

Result

Thus, Program " **Pointers 6** " has been successfully executed

Q. SWAPPING

Your task is to write a code to swapping of two numbers

Input:
Input should contain the value of the limit n
Output:
It should print the Sum of series upto n limit

Source Code

```
#include <stdio.h>
int main()
{int a,b;
scanf("%d%d",&a,&b);
a=a-b;
b=a+b;
a=b-a;
printf("%d %d",a,b);
return 0;
}
```

Sample Input

12 89

Sample Output

89 12

Result

Thus, Program " **SWAPPING** " has been successfully executed

Q. Average & standard deviation

Gowri asks Reshma to write a code to read n numbers (X_i) from the user and print out their average and standard deviation.
Convert all the final answers to integer.

Source Code

```
#include <stdio.h>
#include <math.h>
int main(){
    float x[10];
    int i,n;
    float average, variance, std_deviatiion, sum=0, sum1=0;

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

    average=sum/(float)n;

    for(i=0;i<n;i++){
        sum1=sum1+pow((x[i]-average),2);
    }

    variance=sum1/(float)n;
    std_deviatiion = sqrt(variance);

    printf("Mean=%d\n", (int) average);
    printf("SD=%d", (int)std_deviatiion);

    return 0;
}
```

Sample Input

5
600 470 170 430 300

Sample Output

Mean=394
SD=147

Result

Thus, Program " Average & standard deviation " has been successfully executed

Q. Sum of even and odd numbers

Write a program to find the sum of even and odd numbers in an array.

Source Code

```
#include <stdio.h>
int main()
{int n,i,a[10];
int even_sum=0,odd_sum=0;
scanf("%d",&n);
for(i=0;i<n;i++)
{
    scanf("%d",&a[i]);
}
for(i=0;i<n;i++)
{
    if(a[i]%2==0)
    {
        even_sum=even_sum+a[i];
    }
    else
    {
        odd_sum=odd_sum+a[i];
    }
}
printf("even=%d\n",even_sum);
printf("odd=%d",odd_sum);
return 0;
}
```

Sample Input

5
2 3 6 8 3

Sample Output

even=16
odd=6

Result

Thus, Program " **Sum of even and odd numbers** " has been successfully executed

Q. Rectangular Object

Subbu needs to make a rectangular box for his physics class project. He has bought P cm of wire and S cm² of special paper. He would like to use all the wire (for the 12 edges) and paper (for the 6 sides) to make the box.

What is the largest volume of the box that Johnny can make?

Input

The first line contains t, the number of test cases (about 10). Then t test cases follow.

Each test case contains two integers P and S in a line ($1 \leq P \leq 40000$, $1 \leq S \leq 20000$). You may assume that there always exists an optimal solution for the given input cases.

Output

For each test case, print a real number that is the largest volume of the box that Johnny can make, rounded to two decimal places.

Source Code

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

int main()
{
    // printf("Hello world!\n");
    int t;
    scanf("%d",&t);
    while(t--){
        float p,a;
        scanf("%f %f",&p,&a);
        float l=((p-sqrt(p*p - 24*a))/12);
        float v=l*(a/2.0)- ((l*l*p)/4.0) +l*l ;
        printf("%.2f\n",v);
    }
    return 0;
}
```

Sample Input

```
2
20 14
20 16
```

Sample Output

```
3.00
4.15
```

Result

Thus, Program " **Rectangular Object** " has been successfully executed

Q. Approximate Value

Phineas and Ferb who live in the fictional town of Danville, think and do innovatively on weekends. Every day the boys embark on some grand new project, which annoys their controlling sister Candace, who tries to bust them. One Sunday they were both sitting under a tree in their back yard. They decide to invent a machine which would allow us to enter 2 numbers. It would say whether one of the entered numbers is an appropriate value of the other number entered. They decide to insert a program code in the machine. A number is said to be an approximate value of the other if they differ by utmost 0.5. So write a C program to find whether the given number is approximate number of other.

Input Format:

Input consists of two float type numbers

Output format:

Displays whether the number is approximate or not.

Source Code

```
#include <stdio.h>
int main()
{
    float a,b,c;
    scanf("%f%f",&a,&b);
    c=b-a;
    if(c<=0.5)
        printf("Approximate number");
    else
        printf("Not an Approximate number");
    return 0;
}
```

Sample Input

14.1
14.6

Sample Output

Approximate number

Result

Thus, Program " **Approximate Value** " has been successfully executed

Q. LOWER TRIANGULAR MATRIX

A lower triangular matrix is a square matrix in which all the elements above the diagonal are zero.

That is, all the non-zero elements are in the lower triangle:

Write a C program to find whether a given matrix is a lower triangular matrix or not.

Input Format:

The input consists of $(n \times n+1)$ integers. The first integer corresponds to the number of rows/columns in the matrix. The remaining integers correspond to the elements in the matrix. The elements are read in rowwise order. Order first, then second row and so on. Assume that the maximum value of m and n is 5.

Output Format:

Print yes if it is a lower triangular matrix . Print no if it is not a lower triangular matrix.

Source Code

```
#include <stdio.h>
int main()
{int x[10][10],i,j,char ch,ch1;
int a,b,m=0;
scanf("%d%d%d",&a,&b);
for(i=0;i<a;i++)
{
    for(j=0;j<b;j++)
    {
        scanf("%d" ,&x[i][j]);
        // printf("%d",x[i][j]);
    }
    scanf(" ");
}
for(i=0;i<a;i++)
{
    if(m == 1)
        break;
    for(j=0;j<b;j++)
    {
        if(i<j && x[i][j] != 0){
            m=1;
            // printf("%d\n",x[i][j]);
            break;
        }
    }
}
printf("%d\n",m);
if(m==0)
    printf("yes");
else
    printf("no");
return 0;
}
```

Sample Input

```
3 3
1 0 0
2 1 0
1 1 1
```

Sample Output

```
yes
```

Result

Thus, Program " LOWER TRIANGULAR MATRIX " has been successfully executed

Q. Last to first

Kanna is a mischievous boy. He always used to keep things in reverse order. One fine morning Kanna chat with his uncle. His uncle gave him set of numbers and he ask him to place numbers in reverse order. Can you help him to come out of this problem?

Source Code

```
#include <stdio.h>
int main()
{int n,a[50],i;
scanf("%d",&n);
for(i=0;i<n;i++)
{
    scanf("%d",&a[i]);
}
for(i=n-1;i>=0;i--)
{
    printf("%d ",a[i]);
}
return 0;
}
```

Sample Input

```
3
15 82 91
```

Sample Output

```
91 82 15
```

Result

Thus, Program " **Last to first** " has been successfully executed

Q. Tick Tick Tick

Hemas teacher give one assignment to him. she should write a code to find the time entered in hours,seconds and minutes into seconds. Help him to solve this task

Source Code

```
#include <stdio.h>
int main()
{int a,b,c,d;
scanf("%d%d%d",&a,&b,&c);
d=((a*60*60)+(b*60)+c);
printf("%d",d);
return 0;}
```

Sample Input

2 30 4

Sample Output

9004

Result

Thus, Program " **Tick Tick Tick** " has been successfully executed

Q. Calculating Percentage

SRM University buys an old scooter for Rs. A and Spends Rs. B on its repairs. if he sells the scooter for Rs.C , what is his gain %? Write C program to compute the gain %?

Input format:

The first input is an integer which corresponds to A . The second input is an integer which corresponds to B. The third input is a float which corresponds to gain \$

Source Code

```
#include <stdio.h>
int main()
{int a,b,c;
float e,d,f,g;
scanf("%d%d%d",&a,&b,&c);
d=(a+b);
e=(c-d);
f=e*(1/d);
g=(f*100);
printf("The gain percentage is=%.2f",g);
return 0;
}
```

Sample Input

```
4700
800
5800
```

Sample Output

The gain percentage is=5.45

Result

Thus, Program " **Calculating Percentage** " has been successfully executed

Q. IO 35

Write a program to display a grocery bill of the product purchased in the small market by Vijay. Get the following details from Vijay:

Get the product name
Get the price of the product(Price per Unit)
Get the quantity of the product purchased
Input and Output Format:

Refer sample input and output for formatting specification.

All float values are displayed correct to 2 decimal places.

All text in bold corresponds to input and the rest corresponds to output.

Source Code

```
#include <stdio.h>
int main()
{
char a[100];
float b,e; int c;
scanf("%s",a);
scanf("%f",&b);
scanf("%d",&c);
e=(b*c);

printf("Product Details\n");
printf("%s\n",a);
printf("%.2f\n",b);
printf("%d\n",c);
printf("Bill: %.2f",e);
return 0;
}
```

Sample Input

```
soap
33.00
2
```

Sample Output

```
Product Details
soap
33.00
2
Bill:66.00
```

Result

Thus, Program " **IO 35** " has been successfully executed

Q. FENCING THE GROUND

The college ground is rectangular in shape. The Management decides to build a fence around the ground. In order to help the construction workers to build a straight fence, they planned to place a thick rope around the ground. They wanted to buy only the exact length of the rope that is needed. They also wanted to cover the entire ground with a thick carpet during rainy season. They wanted to buy only the exact quantity of carpet that is needed. They requested your help.

Can you please help them by writing a C program to find the exact length of the rope and the exact quantity of carpet that is required?

Input Format:

Input consists of 2 integers. The first integer corresponds to the length of the ground and the second integer corresponds to the breadth of the ground.

Source Code

```
#include <stdio.h>
int main()
{int a,b,c,d;
scanf("%d%d",&a,&b);
c=2*(a+b);
d=a*b;
printf("Required length is %d m\n",c);
printf("Required quantity of carpet is %d sqm",d);
return 0;}
```

Sample Input

50 20

Sample Output

Required length is 140 m
Required quantity of carpet is 1000 sqm

Result

Thus, Program " **FENCING THE GROUND** " has been successfully executed

Q. ASCII Name

Write a program which reads your name from the keyboard and output a list of ASCII codes, which represent your name

Source Code

```
#include <stdio.h>
int main()
{
    char n[20];
    int i=0;

    scanf("%s",n);
    while(n[i]!='\0')
    {
        printf("%d ",n[i]);
        i++;
    }
    return 0;
}
```

Sample Input

SRMUNIVERSITY

Sample Output

83 82 77 85 78 73 86 69 82 83 73 84 89

Result

Thus, Program " **ASCII Name** " has been successfully executed

Q. Pattern 1

Write a program to generate a following structure:

```
@@@  
@@@  
@@@  
@@@  
@@@
```

Source Code

```
#include <stdio.h>  
int main()  
{int n,i;  
scanf("%d",&n);  
for(j=0;j<n;j++)  
{  
    for(i=0;i<n;i++)  
        (printf("@"));  
    printf("\n");  
}  
return 0;  
}
```

Sample Input

2

Sample Output

```
@@  
@@
```

Result

Thus, Program " **Pattern 1** " has been successfully executed

Q. Two Friends

"Vadim and Roman like discussing challenging problems with each other. One day Vadim told his friend following problem:

Given N points on a plane. Each point p is defined by its two integer coordinates px and py. The distance between points a and b is $\min(|ax - bx|, |ay - by|)$. You should choose a starting point and make a route visiting every point exactly once, i.e. if we write down numbers of points in order you visit them we should obtain a permutation. Of course, overall distance walked should be as small as possible. The number of points may be up to 40.

"40? Maybe 20? Are you kidding?" asked Roman. "No, it's not a joke" replied Vadim. So Roman had nothing to do, but try to solve this problem. Since Roman is really weak in problem solving and you are the only friend, except Vadim, with whom Roman can discuss challenging tasks, he has nobody else to ask for help, but you!

Input description.

The first line of the input contains an 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 denoting the number of points on a plane. The following N lines contain two space-separated integers each coordinates of points.

Output

Output description.

Output the answer for every test case in a separate line. The answer for every test case is a permutation of length N. In case there are several solutions that lead to minimal distance walked, you should choose the lexicographically smallest one. Let P denote such permutation. To make output smaller, you should output H(P). $H(P) = P_1 \text{ xor } P_2 \text{ xor } \dots \text{ xor } P_N$. Have a look at the example and its explanation for better understanding."

Source Code

```
#include<stdio.h>
int main()
{
    int T,i,j;
    scanf("%d",&T);
    int p[40][2];
    int N;
    int out[T];
    for(i=0;i<T;i++)
    {
        out[i] = 0;
        scanf("%d",&N);
        for(j=0;j<N;j++)
        {
            scanf("%d %d",&p[j][0],&p[j][1]);
            out[i] = (out[i]^j+1);
        }
    }
    for(i=0;i<T;i++)
        printf("%d\n",out[i]);
    return 0;
}
```

Sample Input

```
2
2
1 2
0 0
3
3 3
0 0
0 3
```

Sample Output

```
3
0
```

Result

Thus, Program " **Two Friends** " has been successfully executed

Q. LCM

Tom has a homework.His maths teacher gave homework in solving LCM of a given number.Help Tom in finding find LCM (Lowest Common Multiple) using loop.

Source Code

```
#include <stdio.h>
int main()
{int num1,num2,denom,numerat,rema,lcm,gcd;
scanf("%d%d",&num1,&num2);
if(num1>num2)
{
    numerat=num1;
    denom=num2;
}
else
{
    numerat=num2;
    denom=num1;
}
rema=numerat%denom;
while(rema!=0)
{
    numerat=denom;
    denom=rema;
    rema=numerat%denom;
}
gcd=denom;
lcm=(num1*num2)/gcd;
printf("%d",lcm);
return 0;}
```

Sample Input

30
5

Sample Output

30

Result

Thus, Program " LCM " has been successfully executed

Q. Print 5 Sum of Positive Numbers

Write a C program to find allow the user to enter n number and finds the number of positive numbers entered and the sum of all positive numbers entered using a while loop

Input format:

Input consists of n+1 integers. The first integer corresponds to n and the next n integers correspond to the numbers to be added. Consider 0 to be a positive number

Output format:

Refer sample input and output for formatting specifications.

Source Code

```
#include <stdio.h>
int main()
{
    int n,i,a[20],count=0,sum=0;
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    i=0;
    while(i<n)
    {
        if(a[i]>0)
        {
            count++;
            sum=sum+a[i];
        }
        i++;
    }
    printf("Positive Numbers=%d\n",count);
    printf("Sum=%d",sum);
    return 0;
}
```

Sample Input

4
5 -2 -1 6

Sample Output

Positive Numbers=2
Sum=11

Result

Thus, Program " Print 5 Sum of Positive Numbers " has been successfully executed

Q. 1 3 2 PATTERN

Get n number of integers. Mark the first element as i and second as j and third as k.

Check the condition i greater than k greater than j. If all of the condition fails, then print false. Describe it without using any in-built function. Let N be the number of test cases.

Input format: Get any number of integers. The user can give the same element.

Output format: If the input satisfies all possibilities of the condition, then it should return TRUE or it should print FALSE.

EXPLANATION: The user checks every possibilities and assuming that the first element as i and second as j and third as k and so on. If any one condition satisfies then print true, else print as false. Do not use any in-built function.

NOTE: The n should be within 500

Source Code

```
#include <stdio.h>
int main()
{int i,j,k,n;
scanf("%d%d%d",&i,&j,&k);
if((i>j)||((i>k)||((k>j)))
printf("TRUE");
else
printf("FALSE");
return 0;
}
```

Sample Input

```
380
170
429
```

Sample Output

```
TRUE
```

Result

Thus, Program " **1 3 2 PATTERN** " has been successfully executed

Q. Continuous Sum

Chintu is confusing himself with series of numbers. He has to find the sum of the following series. Your task is to write a code to find the sum of the following series
1+1+2+ (1+2+3)+ (1+2+3+4)++n

Input:

Input should contain the value of the limit n

Output:

It should print the Sum of series upto n limit

Source Code

```
#include <stdio.h>
int main()
{int n,sum,i,j,sum1=0;
scanf("%d",&n);
for(i=1;i<=n;i++)
{sum=0;
for(j=1;j<=i;j++)
{
sum=sum+j;
sum1=sum1+sum;
}
printf("Sum of series=%d",sum1);
return 0;
}
```

Sample Input

2

Sample Output

Sum of series=4

Result

Thus, Program " **Continuous Sum** " has been successfully executed

Q. REVERSE

"Dark was deeply studying in the afternoon and came across a concept called as ""REVERSE OF A NUMBER"" He decided to design a program which can reverse any long numbers.
At the end of afternoon, he started to code and finally end up with no results, so decided to ask you all for the help. he wanted you to design a very efficient method to reverse a number and say it as a EVEN or ODD.

Input:

The first line contains T, the number of test cases. Followed by T lines each contains a single number.

Output:

For every test case print the ""EVEN""(without quotes) if reverse of GIVEN number is even else print ""ODD""(without quotes).

Constraints:

$1 \leq T \leq 10000$

$1 \leq N \leq 10100$

Source Code

```
#include <stdio.h>
int main()
{int n,rev_n=0,rem,a[50],i;
scanf("%d",&n);
// while(n--)
rev_n=0;
for(i=0;i<n;i++)
{
    scanf("%d",&a[i]);
    while(a[i]!=0)
    {rem=a[i]%10;
    rev_n=(rev_n*10)+rem;
    a[i]=(a[i]/10);
    }
if(rev_n%2==0)
printf("EVEN\n");
else
printf("ODD\n");
}
return 0;
}
```

Sample Input

```
2
10
22
```

Sample Output

```
ODD
EVEN
```

Result

Thus, Program " REVERSE " has been successfully executed

Course: C

Session: Operators

Timestamp: 2019-3-29 22:53:05

Register Number: RA1811003010363

Q. ASCII code

Write a C program to generate ASCII code for a given character.

Source Code

```
#include <stdio.h>
int main()
{char a;
scanf("%c",&a);
printf("%d",a);
return 0;
}
```

Sample Input

C

Sample Output

67

Result

Thus, Program " **ASCII code** " has been successfully executed

Q. Average Scale

A grocer has a scale of Rs. S1,Rs . S2, Rs. S3 ,Rs. S4 and Rs. S5 for 5 consecutive months. How much sale must he have in the sixth month so that he gets an average sale for Rs.X?

Write a C program to compute the sale in the 6th Format:

Input format:

Input consists of 5 integers and 1 float. The five integers corresponds to s1,s2,s3,s4,s5. The float input corresponds to x.

Source Code

```
#include <stdio.h>
int main()
{int a,b,c,d,e,g;
float x,h;
scanf("%d%d%d%d%d",&a,&b,&c,&d,&e);
scanf("%f",&x);
g=(a+b+c+d+e);
h=((6*x)-g);
printf("sale=%.2f",h);
return 0;
}
```

Sample Input

6475 6947 6855 7240 6512 6500.22

Sample Output

sale=4972.32

Result

Thus, Program " **Average Scale** " has been successfully executed

Q. CARE CENTRE SQUARE EXERCISES

Theatre Square in the capital city of Berland has a rectangular shape with the size $n \times m$ meters. On the occasion of the city's anniversary, a decision was taken to pave the Square with square granite flagstones. Each flagstone is of the size $a \times a$.

What is the least number of flagstones needed to pave the Square? It's allowed to cover the surface larger than the Theatre Square, but the Square has to be covered. It's not allowed to break the flagstones. The sides of flagstones should be parallel to the sides of the Square.

Input

The input contains three positive integer numbers in the first line: n, m and a ($1 \leq n, m, a \leq 10^9$).

Output

Write the needed number of flagstones.

Source Code

```
#include <stdio.h>
int main()
{
    int n,m,a;
    scanf("%d%d%d",&n,&m,&a);
    long long x=0;
    if(n%a==0) x=n/a;
    else x=n/a+1;
    if(m%a==0) x*=m/a;
    else x*=(m/a+1);
    printf("%lld\n",x);

    return 0;
}
```

Sample Input

6 6 4

Sample Output

4

Result

Thus, Program " CARE CENTRE SQUARE EXERCISES " has been successfully executed

Q. Calculating Percentage

SRM University buys an old scooter for Rs. A and Spends Rs. B on its repairs. if he sells the scooter for Rs.C , what is his gain %? Write C program to compute the gain %?

Input format:

The first input is an integer which corresponds to A . The second input is an integer which corresponds to B. The third input is a float which corresponds to gain %

Source Code

```
#include <stdio.h>
int main()
{int a,b,c;
float d,e,f,g;
scanf("%d%d%d",&a,&b,&c);
d=a+b;
e=c-d;
f=e*(1/d);
g=(f*100);
printf("The gain percentage is=%.2f",g);
return 0;
}
```

Sample Input

4700 800 5800

Sample Output

The gain percentage is=5.45

Result

Thus, Program " **Calculating Percentage** " has been successfully executed

Q. Finding Steel Grade

A certain grade of steel is graded according to the following conditions.

Hardness must be greater than 50.
Carbon content must be less than 0.7.
Tensile strength must be greater than 5600.
The grades are as follows:

Grade is 10 if all three conditions are met.

Grade is 9 if conditions (i) and (ii) are met.
Grade is 8 if conditions (i) and (iii) are met.
Grade is 7 if conditions (ii) and (iii) are met.
Grade is 6 if only one condition is met.

Grade is 5 if none of three conditions are met.

Write a program, if the user gives values of hardness, carbon content and tensile strength of the steel under consideration and display the grade of the steel.

Source Code

```
#include <stdio.h>
int main()
{int h,t,x;
float c;
scanf("%d",&x);
while(x--)
{
    scanf("%d%f%d",&h,&c,&t);
    if(h>=50 && c<=0.7 && t>=5600)
        printf("Grade 10\n");
    else if(h>=50 && c<=0.7 && t<5600)
        printf("Grade 9\n");
    else if(h<50 && c<0.7 && t>5600)
        printf("Grade 8\n");
    else if(h>50 && c>0.7 && t>5600)
        printf("Grade 7\n");
    else if(h>50 || c>0.7 || t>5600)
        printf("Grade 6\n");
    else
        printf("Grade 5\n");
}
return 0;
}
```

Sample Input

```
2
53 0.7 5602
55 0 5499
```

Sample Output

```
Grade 10
Grade 9
```

Result

Thus, Program " **Finding Steel Grade** " has been successfully executed

Course: C

Session: Operators

Timestamp: 2019-3-29 22:51:06

Register Number: RA1811003010363

Q. Topper

Raju and Smirthi are classmates. Smirthi will be the class topper for all the times. This time Raju tried to get more in all the subjects. The topper will be decided based on the total value. Your task task is to choose the large total value.

Source Code

```
#include <stdio.h>
int main()
{int a,b;
scanf("%d%d",&a,&b);
if(a>b)
printf("%d",a);
else
printf("%d",b);
return 0;
}
```

Sample Input

400
455

Sample Output

455

Result

Thus, Program " **Topper** " has been successfully executed

Q. AVERAGE SALE

A grocer has a sale of Rs. s1, Rs. s2, Rs. s3, Rs. s4 and Rs. s5 for 5 consecutive months. How much sale must he have in the sixth month so that he gets an average sale of Rs. x?
Write a C program to compute the sale in the 6th month.

Input Format:

Input consists of 5 integers and 1 float. The five integers correspond to s1, s2, s3, s4 and s5. The float input corresponds to x.

Output Format:

Refer sample input and output for formatting specifications.
The float values are displayed correct to 2 decimal places.

Source Code

```
#include <stdio.h>
int main()
{int a,b,c,d,e;
float x,g,h;
scanf("%d%d%d%d%d",&a,&b,&c,&d,&e);
scanf("%f",&x);
g=(a+b+c+d+e);
h=(6*x)-g;
printf("The required sale in the sixth month is %.2f",h);

return 0;
}
```

Sample Input

6435 6927 6855 7230 6562 7000.000

Sample Output

The required sale in the sixth month is 7991.00

Result

Thus, Program " **AVERAGE SALE** " has been successfully executed

Course: C

Session: Operators

Timestamp: 2019-3-29 22:50:20

Register Number: RA1811003010363

Q. Quotient Remainder

Write a C program to find the quotient and remainder for the given dividend and divisor.

Source Code

```
#include <stdio.h>
int main()
{int a,b,c,d;
scanf("%d%d",&a,&b);
c=a/b;
d=a%b;
printf("%d\n",c);
printf("%d",d);
return 0;
}
```

Sample Input

```
25
4
```

Sample Output

```
6
1
```

Result

Thus, Program " **Quotient Remainder** " has been successfully executed

Q. triangle height

Chef belongs to a very rich family which owns many gold mines. Today, he brought N gold coins and decided to form a triangle using these coins. Isn't it strange?

Chef has an unusual way of forming a triangle using gold coins, which is described as follows:

He puts 1 coin in the 1st row.
then puts 2 coins in the 2nd row.
then puts 3 coins in the 3rd row.
and so on as shown in the given figure.

Chef is interested in forming a triangle with maximum possible height using at most N coins. Can you tell him the maximum possible height of the triangle?

Source Code

```
#include <stdio.h>
int main()
{int n,i,a[10];
scanf("%d",&n);
for(i=0;i<n;i++)
{scanf("%d",&a[i]);
if(a[i]<=2)
printf("1\n");
else if(a[i]>2 && a[i]<=5)
printf("2\n");
else if(a[i]>5 && a[i]<=9)
printf("3\n");
else
printf("luck off\n");
return 0;
}
```

Sample Input

```
3
3
5
7
```

Sample Output

```
2
2
3
```

Result

Thus, Program " triangle height " has been successfully executed

Q. Seasoners

The year is divided into four seasons: spring, summer, fall and winter. While the exact dates that the seasons change vary a little bit from year to year because of the way that the calendar is constructed, we will use the following dates for this exercise:

Season First day
Summer March 20
Spring June 21
Fall September 22
Winter December 21

Create a program that reads a month and day from the user. The user will enter the name of the month as a string, followed by the day within the month as an integer.

Then your program should display the season associated with the date that was entered.

Note: Enter First three letter for month example: Jan for january, Feb for Febrary ans so on....and first letter of the month should be capital

Source Code

```
#include <stdio.h>
#include<string.h>
int main()
{
    char month[100],l[3];
    scanf("%s",month);
    scanf("%s",l);
    if((strcmp(l,"22")==0) && (strcmp(month,"Sep")==0))
        printf("Fall");
    if((strcmp(l,"20")==0) && (strcmp(month,"Mar")==0))
        printf("Summer");
    if((strcmp(l,"21")==0) && (strcmp(month,"Jun")==0))
        printf("Spring");
    if((strcmp(l,"21")==0) && (strcmp(month,"Dec")==0))
        printf("Winter");

    return 0;
}
```

Sample Input

Sep
22

Sample Output

Fall

Result

Thus, Program " **Seasoners** " has been successfully executed

Course: C

Session: Data types

Timestamp: 2019-3-28 10:37:52

Register Number: RA1811003010363

Q. DT12: Print date/month/Year

Write a C program to print your name, date of birth, and mobile number.

Source Code

```
#include <stdio.h>
int main()
{
    char a[100],b[100];
    int c,d,e,f;
    scanf("%s%s",a,b);
    scanf("%d%d%d%d",&c,&d,&e,&f);
    printf("Name:%s\n",a);
    printf("DOB:%s %d %d\n", b , c , d);
    printf("Mobile:%d-%d", e , f);
    return 0;
}
```

Sample Input

```
SRMUniversity
March
23
1991
99
1234567890
```

Sample Output

```
Name:SRMUniversity
DOB:March 23 1991
Mobile:99-1234567890
```

Result

Thus, Program " **DT12: Print date/month/Year** " has been successfully executed

Q. Find the Year

Most years have 365 days. However, the time required for the Earth to orbit the Sun is actually slightly more than that. As a result, an extra day, February 29, is included in some years to correct for this difference. Such years are referred to as leap years.

The rules for determining whether or not a year is a leap year follow:

Any year that is divisible by 400 is a leap year.

Of the remaining years, any year that is divisible by 100 is not a leap year.

Of the remaining years, any year that is divisible by 4 is a leap year.

All other years are not leap years.

Write a program that reads a year from the user and displays a message indicating whether or not it is a leap year.

Source Code

```
#include <stdio.h>
int main()
{int a;
scanf("%d",&a);
if((a%400)==0)
printf("Yes");
else if((a%100)==0)
printf("No");
else if((a%4)==0)
printf("Yes");
else
printf("No");
return 0;
}
```

Sample Input

2016

Sample Output

Yes

Result

Thus, Program " **Find the Year** " has been successfully executed

Course: C

Session: Data types

Timestamp: 2019-3-28 10:37:25

Register Number: RA1811003010363

Q. Float Division of numbers

Harinis home work for fourth day is to divide two numbers, help Harini to solve the problem.

Source Code

```
#include <stdio.h>
int main()
{float a,b;
 float c;
 scanf("%f%f",&a,&b);
 c=a/b;
 printf("The Division of two number is:%.6f",c);
 return 0;
}
```

Sample Input

7.2 3.4

Sample Output

The Division of two number is:2.117647

Result

Thus, Program " **Float Division of numbers** " has been successfully executed

Course: C

Session: Data types

Timestamp: 2019-3-28 10:37:11

Register Number: RA1811003010363

Q. Differenzia

In a country named Differenzia the minors and senior citizens are not eligible to vote. Only people aged between 18 to 60 (both inclusive) are eligible to vote. Write a program to determine a person in Differenzia is eligible to vote.

Source Code

```
#include <stdio.h>
int main()
{int a;
scanf("%d",&a);
if(a>=18&&a<=60)
printf("Eligible");
else
printf("Not Eligible");
return 0;
}
```

Sample Input

18

Sample Output

Eligible

Result

Thus, Program " **Differenzia** " has been successfully executed

Q. DT-10: Float to Integer

Write a C program to convert Float to Integer number

If the value exceed 0.5 then add one to the integer component

Source Code

```
#include <stdio.h>
int main()
{
float a,b;
int c,d,e,f,g,h,i,j;
scanf("%f%f",&a,&b);
c=(a*10);
d=(c%10);
e=(c/10);
f=(e+1);
if(d<5)
printf("Value is:%d\n",e);
else
printf("Value is:%d\n",f);
g=(b*10);
h=(g%10);
i=(g/10);
j=(i+1);
if(h<5)
printf("Value is:%d\n",i);
else
printf("Value is:%d\n",j);

return 0;
}
```

Sample Input

4.6

5.8

Sample Output

Value is:5

Value is:6

Result

Thus, Program " **DT-10: Float to Integer** " has been successfully executed

Q. Chemistry pH Value

In chemistry, pH (potential of hydrogen) is a numeric scale used to specify the acidity or basicity of an aqueous solution. It is approximately the negative of the base 10 logarithm of the molar concentration, measured in units of moles per liter, of hydrogen ions. More precisely it is the negative of the base 10 logarithm of the activity of the hydrogen ion. Solutions with a pH less than 7 are acidic and solutions with a pH greater than 7 are basic. Pure water is neutral, at pH 7 (25 °C), being neither an acid nor a base. Contrary to popular belief, the pH value can be less than 0 or greater than 14 for very strong acids and bases respectively.

Measurements of pH are important in agronomy, medicine, biology, chemistry, agriculture, forestry, food science, environmental science, oceanography, civil engineering, chemical engineering, nutrition, water treatment and water purification, and many other applications.

The pH scale is traceable to a set of standard solutions whose pH is established by international agreement. Primary pH standard values are determined using a concentration cell with transference, by measuring the potential difference between a hydrogen electrode and a standard electrode such as the silver chloride electrode. The pH of aqueous solutions can be measured with a glass electrode and a pH meter, or an indicator. Identify Ph Level Name using the following condition. Identify the pH value

pH < 7.0 acidic
pH = 7.0 basic
pH > 7.0 neutral

Source Code

```
#include <stdio.h>
int main()
{int a;
scanf("%d",&a);
if(a<7&&a>0)
printf("acidic");
else if(a>=7)
printf("basic");
else if(a<=0)
printf("neutral");
return 0;
}
```

Sample Input

7.0

Sample Output

basic

Result

Thus, Program " Chemistry pH Value " has been successfully executed

Q. GCD and LCM

Two integers A and B are the inputs. Write a program to find GCD and LCM of A and B.

Input

The first line contains an integer T, total number of testcases. Then follow T lines, each line contains an integer A and B.

Output

Display the GCD and LCM of A and B separated by space respectively.

Constraints

$1 \leq T \leq 1000$
 $1 \leq A, B \leq 1000000$

Source Code

```
#include<stdio.h>

int gcd(int a,int b)
{
    if(b==0)
        return a;
    else
        return gcd(b,a%b);
}

int main()
{
    int a,b,t;
    scanf("%d",&t);
    while(t--){
        scanf("%d%d",&a,&b);
        printf("%d %d\n",gcd(a,b),(a*b/gcd(a,b)));
    }
    return 0;
}
```

Sample Input

```
5
2 3
2 4
3 5
4 6
7 8
```

Sample Output

```
1 6
2 4
1 15
2 12
1 56
```

Result

Thus, Program " **GCD and LCM** " has been successfully executed

Q. Mirror Problem

Puck, the trickster, has again started troubling people in your city. The people have turned on to you for getting rid of Puck.

Puck presents to you a number consisting of numbers from 0 to 9 characters.

He wants you to count the numbers that he uttered.

Kindly help people to get rid of the Puck by developing a small program .The hope of people are on you so you have to solve the riddle

Source Code

```
#include <stdio.h>
int main()
{int n,count=0;
 scanf("%d",&n);
 while(n!=0)
 { n=n/10;
 count++;
 }
 printf("%d",count);
 return 0;
}
```

Sample Input

251298

Sample Output

6

Result

Thus, Program " **Mirror Problem** " has been successfully executed

Course: C

Session: Data types

Timestamp: 2019-3-28 10:34:04

Register Number: RA1811003010363

Q. Substraction of numbers

Harinis home work for second day is to subtract two numbers, help Harini to solve the problem.

Source Code

```
#include <stdio.h>
int main()
{float a,b,c;
scanf("%f%f",&a,&b);
c=a-b;
printf("The Subtraction of two number is:%.6fn",c);
printf("The Subtraction of two number is:%.2f",c);
return 0;
}
```

Sample Input

7.2 1.5

Sample Output

The Subtraction of two number is:5.700000
The Subtraction of two number is:5.70

Result

Thus, Program " **Substraction of numbers** " has been successfully executed

Q. Exchange of gifts

Emily is a very popular Maths Teacher in School. She retires from her service today. Her 7th class students like her very much and they wanted to give her a grand farewell at school. The school HeadMistress is a very strict person and she didn't give permission for them to conduct the farewell in the school premises. The students decided to conduct the farewell at Emily Mam's house itself.

They know the street in which Emily mam lives. The student leader asked Emily Mam to tell her house number. Emily Mam's last class was on Equation for a straight line. She said that a straight line can be represented by the equation $y=mx+c$ and you know how to find the x-intercept and y-intercept of the line and my house number is the sum of the x-intercept and y-intercept of the line.

The students were puzzled. Can you help the students find the house number of Emily Mam?

Given the values of m and c of the line equation $y=mx+c$, write a C program to find the sum of x-intercept and y-intercept.

Input Format:

Input consists of 2 integers. The first integer corresponds to m and the second integer corresponds to c.

Output Format:

Refer Sample Input and Output for exact formatting specifications.

[Assume that the inputs are such that the intercept values are always integers]

Source Code

```
#include <stdio.h>
int main()
{int m,c,a,d;
 //char x,y;
scanf("%d%d",&m,&c);
printf("The line equation is y=%dx+%d\n",m,c);
a=-(c/m);
d=a+c;
printf("The x intercept is %d\n",a);
printf("The y intercept is %d\n",c);
printf("The house number is %d",d);
return 0;
}
```

Sample Input

5 10

Sample Output

The line equation is $y=5x+10$
The x intercept is -2
The y intercept is 10
The house number is 8

Result

Thus, Program " Exchange of gifts " has been successfully executed

Q. IO 14

Ram was a popular maths teacher, he gave a 4 digit number to his students as a assignment .He has to identify ones portion of given number. Please help his students to identify and display the output:

Input and Output Format:

Input can be an integer

Explanation :

Let us say ram given number is 1234 and his student need to identify ones portion is 4

Source Code

```
#include <stdio.h>
int main()
{
    int a,b;
    scanf("%d%d",&a,&b);
    b=(a%10);
    printf("The Digit at ones place of %d is=%d",a,b);
    return 0;
}
```

Sample Input

72

Sample Output

The Digit at ones place of 72 is=2

Result

Thus, Program " IO 14 " has been successfully executed

Q. IO 22

Given an input as integer to the machine , he need to get the output of corresponding floating point number with five digit after decimal points. Please help him to write the program that satisfy the output.

Input and Output Format:

Refer sample input and output for formatting specification.

All text in bold corresponds to input and the rest corresponds to output

Source Code

```
#include <stdio.h>
int main()
{
float a;
scanf("%f",&a);
printf("%.6fn",a);
printf("%.0f",a);
return 0;
}
```

Sample Input

12

Sample Output

```
12.000000
12
```

Result

Thus, Program " **IO 22** " has been successfully executed

Q. Calculate Grade

Write a program that accepts the marks in 3 subjects of a student , calculates the average mark of the student and prints the student's grade. If the average mark is greater than or equal to 90, then the grade is 'A'. If the average mark is 80 and between 80 and 90, then the grade is 'B'.

If the average mark is 70 and between 70 and 80, then the grade is 'C'. If the average mark is 60 and between 60 and 70, then the grade is 'D'. If the average mark is 50 and between 50 and 60, then the grade is 'E'. If the average mark is less than 50, then the grade is 'F'.

Input Format:

Input consists of 3 lines. Each line consists of an integer.

Output Format:

Output consists of a single line. Refer sample output for the format.

Source Code

```
#include <stdio.h>
int main()
{
    int a,b,c,d;
    scanf("%d%d%d",&a,&b,&c);
    d=(a+b+c)/3;
    if (d>=90)
        printf("The grade is A");
    else if (d>=80&&d<90)
        printf("The grade is B");
    else if (d>=70&&d<80)
        printf("The grade is C");
    else if (d>=60&&d<70)
        printf("The grade is D");
    else if (d>=50&&d<60)
        printf("The grade is E");
    else
        printf("The grade is F");
    return 0;
}
```

Sample Input

45 45 45

Sample Output

The grade is F

Result

Thus, Program " **Calculate Grade** " has been successfully executed

Course: C

Session: Input & Ouput

Timestamp: 2019-3-28 10:29:57

Register Number: RA1811003010363

Q. Big

Malu and Doss have play game.The condition to this game is both have to collect bangles from bowl one by one at a given time. Finally the judge have to decide who is the winner based on the count.

Source Code

```
#include <stdio.h>
int main()
{int a,b;
scanf("%d%d",&a,&b);
if(a>b)
printf("%d",a);
else
printf("%d",b);
return 0;
}
```

Sample Input

```
10
5
```

Sample Output

```
10
```

Result

Thus, Program " **Big** " has been successfully executed

Q. Better or Not

One criteria for evaluating 2 different colleges is based on the student strength.

Write a program to compare 2 colleges based on the student strength.

Input Format:

Input consists of 2 integers. The first integer corresponds to the number of students in college 1 and the second integer corresponds to the number of students in college 2.

Output Format:

Output consists of the string College 1 is better or College 2 is better.

Refer sample input and output for further formatting specifications.

Sample Input and Output:

[All text in bold corresponds to input and the rest corresponds to output]

Source Code

```
#include <stdio.h>
int main()
{int a,b;
scanf("%d%d",&a,&b);
if(a>b)
printf("College 1 is better");
else
printf("College 2 is better");
return 0;
}
```

Sample Input

```
1000
2000
```

Sample Output

```
College 2 is better
```

Result

Thus, Program "**Better or Not**" has been successfully executed

Course: C

Session: Input & Ouput

Timestamp: 2019-3-28 10:25:47

Register Number: RA1811003010363

Q. IO 19

Heera is a cute,little gal of age 5 years old, she had one piggybank to save money (coins). Her piggy bank got full and she was excited to her savings money. So, she classified the coins of its followings types , Rs.10 coins, Rs. 5 Coins , Rs .2 Coins, Rs. 1 Coins. Now Please help her to calculate the total amount she saved?

Input :

total number of coins collected on every classified type respectively(Rs..10 Coins, Rs. 5 Coins, Rs. 2 coins, Rs.1 Coins)

Source Code

```
#include <stdio.h>
int main()
{float i;
int a,b,c,d,e,f,g,h;
scanf ("%d%d%d%d",&a,&b,&c,&d);
e=a*10;
f=b*5;
g=c*2;
h=d*1;
i=e+f+g+h;
printf("Total amount in the piggybank=%.2f",i);
return 0;
}
```

Sample Input

```
10
23
43
6
```

Sample Output

Total amount in the piggybank=307.00

Result

Thus, Program " **IO 19** " has been successfully executed

Q. Months

Write a program that accepts an integer as input and displays the corresponding month in words. [January, February, March, April, May, June, July, August, September, October, November, December]. Use Switch statement.

Source Code

```
#include <stdio.h>
int main()
{
    int i;
    for(i=0;i<12;i++)
    {
        scanf("%d",&i);
        switch(i)
        {
            case 1:
                printf("January");
                break;
            case 2:
                printf("February");
                break;
            case 3:
                printf("March");
                break;
            case 4:
                printf("April");
                break;
            case 5:
                printf("May");
                break;
            case 6:
                printf("June");
                break;
            case 7:
                printf("July");
                break;
            case 8:
                printf("August");
                break;
            case 9:
                printf("September");
                break;
            case 10:
                printf("October");
                break;
            case 11:
                printf("November");
                break;
            case 12:
                printf("December");
                break;
        }
    }
    return 0;
}
```

Sample Input

3

Sample Output

March

Result

Thus, Program " Months " has been successfully executed