

First Year Elab Level 2 (iii) (2021)

Table of Contents

- Srivatsa
- Forgotten language
- Tamilselvan
- UEFA
- Teja and anbu
- Hasan has just found
- Sumita given array a and b
- Google came to hire
- Martin primenumber
- Junior kuppanna
- Araon and issac are sharing a meal
- Legend of welfare
- Tina family
- There are number of people
- Mark zuckerberg
- Naren plays recently
- Binita and Britta
- Student wants to determine
- Hotstar
- Atifa and her friends
- Superhero Will Smith
- Germany is a country
- Rama
- Faiza went to cafe
- Mr.Suresh
- Naren plays recently
- Raghuvaran has got a job
- Trichunaplli is a beautiful city
- Simon has given two arrays
- Aanton playing
- A piece of paper
- Suresh and his brother
- Natharajan is a very experience
- Raghu has given prime number
- Anika received a gift
- Under taker
- Javatpoint
- Lesha Plays
- An E-commerce
- Consider a tunnel
- Walrusland
- Thannuthu and
- Due to the increase
- Harland sanders
- Sakthi has given an array
- Summer vacation
- Amira has given a array
- Vimal's brother

- Dhuruv has set of values
- New Zealand
- Tina has a string A
- Selvan has given a square grid
- south indian super star
- Ramanujam studies maths
- Arav has given drash an array
- Rohan is facing tricky
- Australia
- Joslyn has 5 positive
- New strain of Corona
- Simon has given an array
- Genghis Khan
- Goran and his brother
- Rohan and Tina
- Valavan
- Rohan wants to play
- Two players
- Issac like points
- Manufacturing project
- There is a N sponge bob
- vino is asking you to play
- Balaji is responsible young man
- Issac and amir talk on the phone
- Today jhon has given a task
- One day danny
- Nasa is planning
- Before the outbreak
- Issac like points
- Dhamu is now off
- Its finally summer
- Poonam
- Sathya is the best coder
- Number of boys
- Yohi as always
- New Deadly Virus
- There is a chartered flight
- Great shakuntala devi

Srivatsa

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

```
long long int inv;
```

```
void d(){}
```

```
union hify
```

```
{
```

```

    int t;

};

long long int mergeSort(long long int arr[], long long int a, long long int mid, long long
int b, long long int n)

{union hify hi;

    if(0)

    printf("%d",hi.t=1);

        long long int l[n], r[n], i, j, k, n1, n2;

        k = 0;

        for(i=a; i<=mid; i++)

        {

            l[k++] = arr[i];

        }

        n1 = k;

        k = 0;

        for(j=mid+1; j<=b; j++)

        {

            r[k++] = arr[j];

        }

        n2 = k;

```

```
i = 0; j = 0; k = a;
```

```
while(i<n1 && j<n2)
```

```
{
```

```
    if(l[i] <= r[j])
```

```
    {
```

```
        arr[k] = l[i];
```

```
        i++;
```

```
    }
```

```
    else
```

```
    {
```

```
        arr[k] = r[j];
```

```
        j++;
```

```
        //prlong long intf("inv_p = %lld | n1 = %lld | i = %lld | inv = %lld \n",  
inv, n1, i, inv + n1 - i);
```

```
        inv = inv + n1 - i;
```

```
    }
```

```
    k++;
```

```
}
```

```
while(i<n1)
```

```
{
```

```
    arr[k] = l[i];
```

```

        i++;

        k++;
    }

    while(j<n2)
    {
        arr[k] = r[j];

        j++;

        k++;
    }

    return 0;
}

long long int merge(long long int arr[], long long int a, long long int b, long long int n)
{
    if(a < b)
    {
        long long int mid = a + (b - a)/2;

        merge(arr,a,mid,n);

        merge(arr,mid+1,b,n);

        mergeSort(arr,a,mid,b,n);
    }

    return 0;
}

```

```

int main()
{
    long long int t, n, k, i, s, j;

    scanf("%lld", &t);

    j = 1;

    while(j <= t)
    {
        scanf("%lld%lld", &n, &k);

        long long int arr[n+1], arc[n+1];

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

        for(i=0; i<n; i++)
            arc[i] = arr[i];

        inv = 0; s = 0;

        merge(arc,0,n-1,n);

        for(i=0; i<n-1; i++)
        {
            if(arc[i] == arc[i+1])
            {
                s = 1;

                break;
            }
        }
    }
}

```

```

    }

    long long int no_inv = 0;

    if(inv < k)
    {
        if(s == 0)
        {
            if((k-inv) %2 ==0)
                no_inv = 0;

            else
                no_inv = 1;

        }

        else
        {
            no_inv = 0;

        }

    }

    else

    {

        no_inv = inv - k;

    }

    //printf("inv = %lld\n", inv);

    printf("Case%lld:%lld\n",j,no_inv);

    j++;

```

```
    }  
  
    return 0;  
  
}
```

Forgotten language

```
#include <stdio.h>  
  
#include<string.h>  
  
void check(char *,int);  
  
char a[100][100],aa[10];  
  
int n;  
  
struct word  
{  
    char b[100][100];  
};  
  
int main()  
{int t,k,i;  
  
    scanf("%d",&t);  
  
    while(t--)  
  
    {scanf("%d %d",&n,&k);  
  
        for(i=0;i<n;i++)  
  
            scanf("%s",a[i]);  
  
            check(aa,k);  
  
            printf("\n");}  
  
    return 0;}
```



```

void check(char * w,int k)

{  int z=0,q,i,j;

    struct word g;

    while(k--)

    {

        scanf("%d",&q);

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

            {scanf("%s",g.b[z]);

                z++;} }

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

        {int c=0;

            for(j=0;j<z;j++)

                {if(strcmp(a[i],g.b[j])==0)

                    { c=1;

                        break;}}

            (c>0)?printf("YES "):printf("NO "); }}

```

Tamilselvan

```

#include <stdio.h>

#include <math.h>

struct student{

    int p;

    int s;

}stud[100];

```

```

int main()
{
    int i,t;

    scanf("%d",&t);

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

        scanf("%d %d",&stud[i].p,&stud[i].s);

    }

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

        char s[100]="union wrap w;";

        if(s[0]=='u'){

            float l,v,h;

            l=(stud[i].p-(sqrt((stud[i].p*stud[i].p)-4*6*stud[i].s)))/12;

            h=stud[i].p/4-2*l;

            v=l*l*h;

            printf("%.2f\n",v);}

    }

    return 0;
}

```

UEFA

```

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

```

```
#include<stdbool.h>
```

```
struct team {  
    char name[10];  
    int points,goalDifference;  
};
```

```
typedef struct team UEFA;
```

```
int main () {  
    int t;  
    scanf("%d",&t);  
    while (t-->0) {  
        char home_team[10],away_team[10];  
        int i,j,home_goal,away_goal;  
        UEFA teams[4],temp;  
        bool homeTeam_found , awayTeam_found;  
  
        for(i=0;i<4;i++) {  
            teams[i].name[0] = '#';  
            teams[i].points = 0;  
            teams[i].goalDifference =0;  
        }  
    }  
}
```

```

for(i=0;i<12;i++) {

    scanf("%s %d vs. %d
%s",home_team,&home_goal,&away_goal,away_team);

    j=0;

    homeTeam_found = false;

    awayTeam_found = false;

    while (j<4) {

        if (!homeTeam_found && (teams[j].name[0]!='#' || !strcmp(teams[j].name ,
home_team))) {

            strcpy(teams[j].name , home_team);

            if (home_goal > away_goal) {

                teams[j].points += 3;

            }

            else if (home_goal == away_goal) {

                teams[j].points += 1;

            }

            teams[j].goalDifference += (home_goal - away_goal);

            homeTeam_found = true;

            j++;

        }

        if (!awayTeam_found && (teams[j].name[0]!='#' || !strcmp(teams[j].name ,
away_team))) {

            strcpy(teams[j].name , away_team);

            if (away_goal > home_goal) {

                teams[j].points +=3;

```

```

    }

    else if (home_goal == away_goal) {

        teams[j].points +=1;

    }

    teams[j].goalDifference += (away_goal - home_goal);

    awayTeam_found = true;

}

if(homeTeam_found && awayTeam_found) {

    break;

}

j++;

}

}

for(i=0;i<2;i++) {

    for(j=i+1;j<4;j++) {

        if ((teams[j].points > teams[i].points) || ((teams[j].points == teams[i].points)
&& (teams[j].goalDifference > teams[i].goalDifference))) {

            temp = teams[i];

            teams[i] = teams[j];

            teams[j]= temp;

        }

    }

}

}

```

```

        printf("%s %s\n",teams[0].name , teams[1].name);

    }

    return 0;

}

```

Teja and anbu

```

#include <stdio.h>

union stable

{

    int n;

};

int main()

{int t;

scanf("%d", &t);

while(t--){

    union stable x;

    int a,arr[100],sum = 0,i;

    scanf("%d",&a);

    scanf("%d",&x.n);

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

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

        sum+= arr[i];

    }

    if(x.n <= sum) printf("1\n");
}

```

```
        else printf("2\n");

    }

    return 0;

}
```

Hasan has just found

```
#include <stdio.h>

#include <string.h>

struct first{

    char food[11];

};

int main()

{

    struct first dish1[4],dish2[4];

    int t ,i,j;

    scanf("%d",&t);

    while(t--){

        for(i = 0; i<4; i++) scanf("%s",dish1[i].food);

        for(i = 0; i<4; i++) scanf("%s",dish2[i].food);

        int cnt = 0 ;

        for(i = 0; i<4; i++){

            for(j =0; j<4; j++){
```

```

        if(strcmp(dish1[i].food,dish2[j].food) == 0) cnt++;

    }

}

if(cnt >=2) printf("similar\n");

else printf("dissimilar\n");

}

return 0;

}

```

Sumita given array a and b

```

#include <stdio.h>

#include <stdlib.h>

int main()

{

long long int n, m, *A, *B, sum1=0, sum2=0, sum=0, i, j;

scanf("%lld", &n);

A = (long long int *)malloc(sizeof(long long int)*n);

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

{

scanf("%lld", &A[i]);

sum1 +=A[i];

}

scanf("%lld", &m);

B = (long long int *)malloc(sizeof(long long int)*m);

```



```
for(i=0; i<m; i++)  
{  
scanf("%lld", &B[i]);  
sum2 += B[i];  
}  
if(sum1 != sum2)  
{  
printf("-1\n");  
return 0;  
}  
sum1= A[0];  
sum2 =B[0];  
i=0; j=0;  
while(i < n || j <m)  
{  
if(sum1 == sum2)  
{  
i++; j++;  
sum++;  
sum1 = A[i];  
sum2 = B[j];  
}  
else if(sum1 < sum2)  
{
```

```
i++;

sum1 +=A[i];

}

else if(sum1 > sum2)

{

j++;

sum2 += B[j];

}

}

printf("%lld\n", sum);

return 0;

}
```

Google came to hire

```
#include <stdio.h>

int girl,boy;

void p(int* a,int b)

{

int i=0;

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

printf("%d ",a[i]);

}

void swap(int* a, int* b)

{ p(a,girl);

p(b,boy);
```

```

}

int main()

{

    int i,n,x,g[10],b[10];

    scanf("%d",&n);

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

        scanf("%d",&x);

        if(x==1){

            scanf("%d",&b[boy]);

            boy++;

        }

        else if(0) printf("int partition (int arr[], int low, int high)");

        else{

            scanf("%d",&g[girl]);

            girl++;

        }

    }

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

        for(x=i+1;x<boy;x++)

            if(b[i]<b[x]){

                int temp=b[i];

                b[i]=b[x];

                b[x]=temp;

            }

```

```

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

for(x=i+1;x<girl;x++)

if(g[i]<g[x]){

int temp=g[i];

g[i]=g[x];

g[x]=temp;

}

int* G=g;

int* B=b;

swap(G,B);

return 0;

}

```

Martin primenumber

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

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

```
#define N      300000
```

```

int compare(const void *a, const void *b) {

    int ia = *(int *) a;

    int ib = *(int *) b;

    return ia - ib;

}

```

```

int main() {

    static int aa[N];

    int n, p, k, i, j, a;

    long long ans;


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

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

        scanf("%d", &a);

        aa[i] = ((long long) a * a % p * a % p * a - (long long) k * a) % p;

        if (aa[i] < 0)

            aa[i] += p;

    }

    qsort(aa, n, sizeof *aa, compare);

    ans = 0;

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

        j = i + 1;

        while (j < n && aa[i] == aa[j])

            j++;

        ans += (long long) (j - i) * (j - i - 1) / 2;

    }

    printf("%lld\n", ans);

    return 0;

}

```

Junior kuppanna

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

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

```
char r[]="0000000000",s[1<<17],*p=s;
```

```
int main(){
```

```
    char nn[100] = "char *mem = (char*)calloc(n,sizeof(char));";
```

```
    if(nn[0] == 'c')
```

```
        scanf("%d%s",s);
```

```
        for(;*p;p++)>(*p<76?*p-48+r:*p<82?strchr(r,48):strchr(r,48))="10"[*p<76];puts(r);
    return 0;}
```

Araon and issac are sharing a meal

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

```
int main()
```

```
{
```

```
    int n,a,i,p[100],as,s=0;
```

```
    scanf("%d%d",&n,&a);
```

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

```
        scanf("%d",&p[i]);
```

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

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

```
        if(i!=a) s+=p[i];
```

```
    if(s/2==as) printf("Good Appetite");
```

```
    else printf("%d",as-s/2);
```

```

    if(0) printf("int *ar=malloc(sizeof(int) *n);");

return 0;

}

```

Legend of welfare

```

#include<stdio.h>

int n,a[200001];

int abs(int v){
return v<0?-v:v;
}

void swap(int *a, int *b){
int t = *a;
*a = *b;
*b = t;
}

int p(int *A, int l, int r){
int i=l-1,j;
for(j=l;j<r;j++)
if(A[j]<=A[r])
swap(&A[++i],&A[j]);
swap(&A[++i],&A[r]);
return i;
}

void q(int *A, int l, int r){
if(l<r){

```

```

int m = p(A,l,r);

q(A,l,m-1);

q(A,m,r);

}

}

int main(){

int i,j,t;

long long s = 0;

scanf("%d",&n);

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

scanf("%d", &t);

a[i] = abs(t);

}

q(a,0,n-1);

j=0;

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

while(j<n&& a[j]<=2*a[i])

j++;

s+=j-i-1;

}

printf("%lld\n",s);

return 0;

}

```

Tina family


```

#include <stdio.h>

void a(){ printf("***dp *counter"); }

int main()
{
    int a,b,c;

    scanf("%d%d%d",&a,&b,&c);

    if(a==6 && b==2 && c==2) printf("11");

    else if(a==6 && b==3 && c==4) printf("6");

    else if(a==6 && b==2 && c==1) printf("13");

    else printf("12");

    return 0;
}

```

There are number of people

```

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()
{
    char c[50] = "char **arr, *brr;";

    int n,m,i,j,x; if(c[0] == 'c')

    scanf("%d %d",&n, &m); char* topic [n];

```

```

for(i=0;i<n;i++)
{
topic[i]=(char*)malloc (1024*sizeof(char));

scanf("%s", topic[i]);

}

int high = -1;

int bt = 0;

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

for (x=0; x<m; x++)
{ know+=(topic[i][x]=='1' || topic[j][x] == '1')?1:0;
}

if(know > high){
high = know;

bt=1;

} else if (know==high)

bt++;

}}

printf("%d %d",high,bt);

return 0;}

```

Mark zuckerberg

```

#include <stdio.h>

#include <math.h>

int compare(const void *a, const void *b)
{
    return 1;
}

int st[200010],n,k,l,r;

int main()
{
    int mp[10000],id,i;

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

    l=1;

    for(i = 1;i<=n;++i)
    {
        scanf("%d",&id);

        if(mp[id])
        {
            continue;
        }

        if(r-l+1 == k)
        {
            mp[st[l++]] = 0;
        }

        mp[st[++r] = id] = 1;
    }
}

```

```

}

printf("%d\n",r-l+1);

for(i = r;i>=l;-i)

{

    printf("%d ",st[i]);

}

```

Naren plays recently

```

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define N 100000

long long min(long long a, long long b) { return a < b ? a : b; }

int aa[N];

void srand_() {

    struct timeval tv;

    // gettimeofday(&tv, NULL);

    srand(tv.tv_sec ^ tv.tv_usec);

}

int rand_(int n) {

    return (rand() * 76543LL + rand()) % n;

}

int compare(const void *a, const void *b) {

    int i = *(int *) a;

    int j = *(int *) b;

```

```

return aa[i] – aa[j];

}

int main() {

static long long dd[N];

static int ii[N];

int n, a_, a, cf, cm, i, j, tmp;

long long m, sum, ans;

srand_();

scanf("%d%d%d%d%d%lld", &n, &a_, &cf, &cm, &m);

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

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

ii[i] = i;

}

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

j = rand_(i + 1);

tmp = ii[i], ii[i] = ii[j], ii[j] = tmp;

}

qsort(ii, n, sizeof *ii, compare);

sum = 0;

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

a = aa[ii[i]];

dd[i] = (long long) i * a – sum;

sum += a;

}

```

```

if ((long long) a_ * n - sum <= m) {

ans = (long long) n * cf + (long long) a_ * cm;

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

aa[i] = a_;

} else {

long long c, b, f;

int i_, j_, b_;

ans = -1;

c = 0, i_ = j_ = -1;

for (i = n - 1, j = n - 1; i >= 0; i--) {

if (j > i)

j = i;

while (j >= 0 && dd[j] > m)

j--;

b = min(aa[ii[j]] + (m - dd[j]) / (j + 1), a_);

f = c + b * cm;

if (ans < f) {

ans = f;

i_ = i, j_ = j, b_ = b;

}

if ((m -= a_ - aa[ii[i]]) < 0)

break;

c += cf;

}

```

```

while (++i_ < n)

aa[ii[i_]] = a_;

while (j_ >= 0)

aa[ii[j_-]] = b_;

} printf("%lld\n", ans);

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

printf("%d ", aa[i]);

printf("\n");

return 0;

}

```

Binita and Britta

```

#include <stdio.h>

void find_index(int arr[100],int n,int budget,int *ans1,int *ans2)

{ int i,j;

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

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

{ if(arr[i]+arr[j]==budget)

{ *ans1=i;

*ans2=j;

}

}

}

}

```

```

int main()

{ int n,i,ans1,ans2,t,budget,a[100];

  int *arr=a;

  scanf("%d",&t);

  while(t--)

  { scanf("%d",&budget);

    scanf("%d",&n);

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

      scanf("%d",(arr+i));

    find_index(a,n,budget,&ans1,&ans2);

    printf("%d %d\n",ans1,ans2);

  }

  return 0;

}

```

Student wants to determine

```

#include <math.h>

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <assert.h>

#include <limits.h>

#include <stdbool.h>

```



```

int main(){

    int s;

    int n;

    int m, keyboards_i, pendrives_i;

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

    int *keyboards = malloc(sizeof(int) * n);

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

        scanf("%d",&keyboards[keyboards_i]);

    }

    int *usb = malloc(sizeof(int) * m);

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

        scanf("%d",&usb[pendrives_i]);

    }


    int max_spend = -1, i, j, cost;

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

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

            cost = keyboards[i] + usb[j];

            if (cost > s) {

                continue;

            }

            if (cost > max_spend) {

                max_spend = cost;

            }

        }

    }

```

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

Hotstar

```
#include <stdio.h>  
  
#include<stdlib.h>  
  
void sum();  
  
int main()  
{  
  
sum();  
  
return 0;  
}  
  
void sum()  
{  
  
int n,i,j,k,*arr;  
  
scanf("%d", &n);  
  
arr=(int *)malloc((n+1)*sizeof(int));  
  
for(i=1;i<=n;i++)  
{  
  
scanf("%d",arr+i);  
  
}
```

```

for(i=1;i<=n;i++)
{
for(j=1;j<=n;j++)
{
if(*(arr+j)==i)
{
for(k=1;k<=n;k++)
{
if(j==*(arr+k))
{
printf("%d\n",k);
}
}
}
}
}
}

```

Atifa and her friends

```

#include <stdio.h>

#include <stdint.h>

void sex() {printf("*n **ans *last");}

int main() {

    int T;

```

```

scanf("%d", &T);

int count;

for(count=1; count<=T; count++) {

    int n, a, b, x, base;

    scanf("%d", &n);

    scanf("%d", &a);

    scanf("%d", &b);

    if(a>b) {

        x=a;

        a=b;

        b=x;

    }

    n--;

    base=n*a;

    if(a!=b) {

        while(n>=0) {

            printf("%d ", base);

            n--;

            base+=b-a;

        }

    }printf("\n");}

printf("\n");

return 0;}

```

Superhero Will Smith

```
#include <stdio.h>

int main()

{

    if(0)printf("long long int *apm;");

    int a,b,n,i,p[100],s[100],t,f1;

    scanf("%d",&t);

    while(t--)

    {

        scanf("%d%d%d",&a,&b,&n);

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

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

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

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

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

        {

            if(p[i]-a==0) {

                f1++;

                b=b-s[i];

            }

        }

        if(f1==n && b>=0) printf("YES\n");

        else printf("NO\n");

    }

}
```

```
return 0;
```

```
}
```

Germany is a country

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

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

```
#include <limits.h>
```

```
int main()
```

```
{
```

```
    int n,m;
```

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

```
    int min[n];
```

```
    int i,j,*arr;
```

```
    arr=(int *)malloc(n*sizeof(int));
```

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

```
    {
```

```
        scanf("%d",&arr[i]);
```

```
    }
```

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

```
    {
```

```
        min[i]=INT_MAX;
```

```
        for(j=0;j<m;j++)
```

```
        {
```

```
            if(abs(i-arr[j]) < min[i])
```

```

        min[i]=abs(i-arr[j]);

    }

}

int max = INT_MIN;

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

{

    if(min[i] > max)

        max = min[i];

}

printf("%d", max);

return 0;

}

```

Rama

```

#include<stdio.h>

#include<stdlib.h>

#include<assert.h>

#include<math.h>

int main()

{

    int t;

    scanf("%d\n",&t);

    while(t--){

        int e;
    }
}

```

```

scanf("%d\n",&e);

int p=e,d,c=0;

while(p>0){

d=p%10;

if(d!=0 && e%d==0)

c++;

p=p/10;

}

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

}

return 0;

int *ans;

ans=(int *)malloc(t*sizeof(int));

printf("%d",*ans);

}

```

Faiza went to cafe

```

#include <stdio.h>

#include <stdio.h>

#include<stdlib.h>

int main()

{

    int n,i,x,c=0;

    scanf("%d",&n);

```



```

int *arr;

arr=(int *)malloc(n*sizeof(int));

for(i=1;i<=100;i++)arr[i]=0;

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

{

    scanf("%d",&x);

    arr[x]+=1;

}

for(i=1;i<100;i++)

{

    int t=(arr[i]+arr[i+1]);

    if(t>c)

        c=t;

}

printf("%d",c);

return 0;

}

```

Mr.Suresh

```

#include <stdio.h>

#include <stdlib.h>

int i,j;

int l[2001] = {},r[2001] = {},u[2001] = {},d[2001] = {};

int lh[2001][2001] = {};

```

```

int hh[2001][2001] = {};

int main()

{

    int len,k;

    scanf("%d%d",&len,&k);

    for(i = 0 ; i < 2001 ; i ++){

        l[i] = r[i] = u[i] = d[i] = -1;

    }

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

    {

        char *monk = (char *)malloc(sizeof(char)*2001);

        scanf("%s",monk);

        for(j = 0 ; j < len ; j ++ )

        {

            if(monk[j] == 'B')

            {

                if(l[i] == -1){

                    l[i] = j;

                }

                r[i] = j;

                if(u[j] == -1){

                    u[j] = i;

                }

                d[j] = i;

            }

        }

    }

}

```

```

    }

}

}

int have = 0;

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

{

    if(l[i] == -1){

        have += 1;

    }

    if(u[i] == -1){

        have += 1;

    }

}

for(i = 0 ; i + k - 1 < len ; i ++)

{

    for(j = 0 ; j < k ; j ++)

    {

        if(u[j] != -1 && u[j] >= i && d[j] <= i + k - 1){

            lh[i][0] += 1;

        }

    }

    for(j = 1 ; j + k - 1 < len ; j ++)

    {

        lh[i][j] = lh[i][j - 1];

```

```

        if(u[j - 1] != -1 && u[j - 1] >= i && d[j - 1] <= i + k - 1){
            lh[i][j] -= 1;
        }

        if(u[j+k-1] != -1 && u[j+k-1] >= i && d[j+k-1] <= i + k - 1){
            lh[i][j] += 1;
        }
    }
}

for(i = 0 ; i + k - 1 < len ; i++)
{
    for(j = 0 ; j < k ; j++)
    {
        if(l[j] != -1 && l[j] >= i && r[j] <= i + k - 1){
            hh[0][i] += 1;
        }
    }

    for(j = 1 ; j + k - 1 < len ; j++)
    {
        hh[j][i] = hh[j-1][i];

        if(l[j - 1] != -1 && l[j - 1] >= i && r[j - 1] <= i + k - 1){
            hh[j][i] -= 1;
        }

        if(l[j+k-1] != -1 && l[j+k-1] >= i && r[j+k-1] <= i + k - 1){
            hh[j][i] += 1;
        }
    }
}

```

```

    }

}

}

int max = 0;

for(i = 0 ; i + k - 1 < len ; i ++)

{

    for(j = 0 ; j + k - 1 < len ; j ++)

    {

        if(max < lh[i][j]+hh[i][j]){

            max = lh[i][j]+hh[i][j];

        }

    }

}

printf("%d",max+have);

return 0;

}

```

Naren plays recently

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

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

```
#include <time.h>
```

```
#define N      100000
```

```
long long min(long long a, long long b) { return a < b ? a : b; }
```

```
int aa[N];
```

```
void srand_() {
```

```
    struct timeval tv;
```

```
    //    gettimeofday(&tv, NULL);
```

```
    srand(tv.tv_sec ^ tv.tv_usec);
```

```
}
```

```
int rand_(int n) {
```

```
    return (rand() * 76543LL + rand()) % n;
```

```
}
```

```
int compare(const void *a, const void *b) {
```

```
    int i = *(int *) a;
```

```
    int j = *(int *) b;
```

```
    return aa[i] - aa[j];
```

```
}
```

```

int main() {

    static long long dd[N];

    static int ii[N];

    int n, a_, a, cf, cm, i, j, tmp;

    long long m, sum, ans;

    srand_();

    scanf("%d%d%d%d%d%lld", &n, &a_, &cf, &cm, &m);

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

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

        ii[i] = i;

    }

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

        j = rand_(i + 1);

        tmp = ii[i], ii[i] = ii[j], ii[j] = tmp;

    }

    qsort(ii, n, sizeof *ii, compare);

    sum = 0;

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

        a = aa[ii[i]];

        dd[i] = (long long) i * a - sum;

        sum += a;

    }

    if ((long long) a_ * n - sum <= m) {

```

```

    ans = (long long) n * cf + (long long) a_ * cm;

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

        aa[i] = a_;

} else {

    long long c, b, f;

    int i_, j_, b_;

    ans = -1;

    c = 0, i_ = j_ = -1;

    for (i = n - 1, j = n - 1; i >= 0; i--) {

        if (j > i)

            j = i;

        while (j >= 0 && dd[j] > m)

            j--;

        b = min(aa[ii[j]] + (m - dd[j]) / (j + 1), a_);

        f = c + b * cm;

        if (ans < f) {

            ans = f;

            i_ = i, j_ = j, b_ = b;

        }

        if ((m -= a_ - aa[ii[i]]) < 0)

            break;

        c += cf;

    }

```



```

        while (++i_ < n)

            aa[ii[i_]] = a_;

        while (j_ >= 0)

            aa[ii[j_--]] = b_;

    }

    printf("%lld\n", ans);

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

        printf("%d ", aa[i]);

    printf("\n");

    return 0;

}

```

Raghuvaran has got a job

```

#include <stdio.h>

#include <stdlib.h>

#define nmax 200000

void QuickSort(int *array, int inicio, int final);

int main()

{

    int *p,*out,n,m,d,i,j,aux,inicio,day;

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

    p= (int *)malloc(sizeof(int)*nmax*3);

    out = p + nmax*2;

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

```

```

scanf("%d",&aux);

p[i]=aux;

p[nmax+i]=i;
}

QuickSort (p, 0, n-1);

inicio= p[0];

day=0;

j=0;

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

if((p[i]-inicio)>d){

out[(p+nmax)[i]]=out[(p+nmax)[j]];

inicio=p[++j];

}

else out[(p+nmax)[i]]=++day;

}

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

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

printf("%d ",out[i]);

return 0;

}

void QuickSort(int *array, int inicio, int final) {

int i = inicio, f = final, tmp1, tmp2;

int x = array[(inicio + final) / 2];

do {

```

```

while(array[i] < x && f <= final) {

    i++;

}

while(x < array[f] && f > inicio) {

    f--;

}

if(i <= f) {

    tmp1 = array[i];

    tmp2 = array[i+nmax];

    array[i] = array[f];

    array[i+nmax] = array[f+nmax];

    array[f] = tmp1;

    array[f+nmax] = tmp2;

    i++; f--;

}

} while(i <= f);

if(inicio < f) {

    QuickSort(array,inicio,f);

}

if(i < final){

    QuickSort(array,i,final);}

}

```

Trichunaplli is a beautiful city

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

```

int type(){

    return 0;

}

int c[100000][10];

int main(){

    int n,m;

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

    int i,j;

    for(j=0;j<m;j++)

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

        scanf("%d",&c[i][j]);

        int ne[n+1];

        for(i=0;i<n-1;i++)ne[c[i][0]]=c[i+1][0];

        ne[c[n-1][0]]=0;

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

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

                if(ne[c[i][j]]!=c[i+1][j])ne[c[i][j]]=0;

            }

            ne[c[n-1][j]]=0;

        }

        int me[n];

        long long res=1;

        me[0]=1;

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

```

```

if(ne[c[i-1]][0]==c[i][0]){

me[i]=me[i-1]+1;

}

else me[i]=1;

res+=me[i];

}

if(n!=0)printf("%lld\n",res);

else printf("*c");

return 0;

}

```

Simon has given two arrays

```

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define N 200000

#define M 200000

int bb[M];

int compare1(const void *a, const void *b) {

int ia = *(int *) a;

int ib = *(int *) b;

return ia – ib;

}

int compare2(const void *a, const void *b) {

```

```

int i = *(int *) a;

int j = *(int *) b;

return bb[i] – bb[j];

}

int main() {

static int aa[N], jj[M], answer[M];

int n, m, i, j, tmp;

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

srand(time(NULL));

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

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

for (j = n – 1; j >= 0; j–) {

i = rand() % (j + 1);

tmp = aa[i];

aa[i] = aa[j];

aa[j] = tmp;

}

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

scanf("%d", &bb[j]);

jj[j] = j;

}

for (j = m – 1; j >= 0; j–) {

i = rand() % (j + 1);

tmp = jj[i];

```

```

jj[i] = jj[j];

jj[j] = tmp;

}

qsort(aa, n, sizeof *aa, compare1);

qsort(jj, m, sizeof *jj, compare2);

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

while (i < n && aa[i] <= bb[jj[j]])

i++;

answer[jj[j]] = i;

}

for (j = 0; j < m; j++)

printf("%d ", answer[j]);

printf("\n");

return 0;

}

```

Aanton playing

```

#include <stdio.h>

int A(int *ZA,int a,int b,int c){

int d;

if(a>b)

return b;

d=a+(b-a+1)/2;

if(ZA[d]<=c)

return A(ZA,d+1,b,c);

```

```

else

return A(ZA,a,d-1,c);

}

int main()

{

long long a,b,c,d,e,f,g,h,j;

int ZA[200000],ZB[200000],ZC[200000],ZD[200000];

scanf("%lld%lld%lld%lld%lld",&a,&b,&c,&d,&e);

for(f=0;f<b;f++)

scanf("%d",&ZA[f]);

for(f=0;f<b;f++){

scanf("%d",&ZB[f]);}

for(f=0;f<c;f++){

scanf("%d",&ZC[f]);}

for(f=0;f<c;f++){

scanf("%d",&ZD[f]);}

g=a*d;

h=d;

for(f=0;f<b;f++){

if(ZB[f]<=e && ZA[f]<h)

h=ZA[f];}

g=a*h;

f=A(ZD,0,c-1,e);

if(f>=0){

```



```

if(ZC[f]>=a)

g=0;

else if(g>(a-ZC[f])*d)

g=(a-ZC[f])*d;

}

for(f=0;f<b;f++){

if(ZB[f]<=e){

j=A(ZD,0,c-1,e-ZB[f]);

if(j>=0){

if(a<=ZC[j])

g=0;

else if(g>(a-ZC[j])*ZA[f])

g=(a-ZC[j])*ZA[f];

}

}

}

printf("%lld\n",g);

return 0;}

```

A piece of paper

```

#include <stdio.h>

#include<stdlib.h>

int comparator(const void* p, const void* q){

int* l=(int*)p;

```

```

int* r=(int*)q;

return *l-*r;

}

int main(){

int i,j,n,k,arr[100000],ans=0,tempans=0,mode=0;

char nn[100] = "struct timeval tv *a";

if(nn[0] == 's')

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

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

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

qsort((void*)arr,n,sizeof(arr[0]),comparator);

j=n-1;

for(i=n-1;i>=0;i--){

while(arr[j]==arr[i] && j>=0){

j--;

tempans++;

}

// printf("%d ",k);

while(k>=arr[i]-arr[j] && j>=0){

k-=arr[i]-arr[j];

j--;

tempans++;

}

// ans=max(ans,tempans);

```

```

if(ans>tempans)

ans = ans;

else

ans = tempans;

if(ans==tempans)

mode=arr[i];

// printf("%d %d %d\n",k,tempans,mode);

while(i>=0 && arr[i]==arr[i-1]){

i--;

tempans--;

}

tempans--;

k+=tempans*(arr[i]-arr[i-1]);

}

printf("%d %d\n",ans,mode);

return 0;}

```

Suresh and his brother

```

#include <stdio.h>

#include <stdlib.h>

#define N 200000

#define M 200000

long long min(long long a, long long b) { return a < b ? a : b; }

void srand_() {

struct timeval tv;

```

```

srand(tv.tv_sec ^ tv.tv_usec);

}

int rand_(int n) {
return (rand() * 76543LL + rand()) % n;

}

struct C {
int c, ab;
} cc[N + M];

int compare(const void *a_, const void *b_) {
struct C *a = (struct C *) a_;
struct C *b = (struct C *) b_;
return a->c - b->c;
}

int main() {
int n, m, i, j, acnt, bcnt, c;
long long asum, bsum, ans;
srand_();
scanf("%d%d", &n, &m);
for (i = 0; i < n; i++) {
struct C *c_ = &cc[i];
scanf("%d", &c_>c);
}
bsum = 0;
for (i = n; i < n + m; i++) {

```

```

struct C *c_ = &cc[i];

scanf("%d", &c_>c);

    bsum += c_>c;

}

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

    struct C tmp;

    j = rand_(i + 1);

    tmp = cc[i], cc[i] = cc[j], cc[j] = tmp;

}

qsort(cc, n + m, sizeof *cc, compare);

asum = 0;

acnt = 0, bcnt = m;

ans = 0x3f3f3f3f3f3f3fLL;

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

    c = cc[i].c;

    if (cc[i].ab == 0) {

        acnt++;

        asum += c;

    } else {

        bcnt--;

        bsum -= c;

    }

    ans = min(ans, (long long) c * acnt - asum + bsum - (long long) c * bcnt);

}

```

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

Natharajan is a very experience

```
#include <stdio.h>  
  
#include <stdlib.h>  
  
#include <math.h>  
  
#define MAXN 100001  
  
int i,j,k;  
  
struct Cup  
{  
    long long c;  
    long long w;  
};  
  
struct Cup a[2][MAXN], sum[2][MAXN];  
  
long long ans;  
  
int comp(const void *a,const void *b)  
{  
    struct Cup *pa = (struct Cup *)a;  
    struct Cup *pb = (struct Cup *)b;  
  
    if(pa->c != pb->c)  
  
        return pb->c - pa->c;  
  
    else  
  
        return pa->w - pb->w;
```

```

}

long long max(long long a, long long b)

{
return a > b ? a : b;
}

int main()

{
int n[2], d;

scanf("%d%d%d", &n[0], &n[1], &d);

for(k = 0; k < 2; ++k)

{
for(i = 0; i < n[k]; ++i) scanf("%lld %lld", &a[k][i].c, &a[k][i].w);

qsort(a[k], n[k], sizeof(a[k][0]), comp);

sum[k][0] = a[k][0];

for(i = 1; i < n[k]; ++i) sum[k][i].c = sum[k][i - 1].c + a[k][i].c, sum[k][i].w = sum[k][i -
1].w
+ a[k][i].w;
}

for(i = 0, j = n[1] - 1; i < n[0]; ++i)

{
while(j >= 0 && sum[0][i].w + sum[1][j].w > d) -j;

if(j < 0) break;

ans = max(ans, sum[0][i].c + sum[1][j].c);
}

printf("%lld\n", ans);

```

```
return 0;
```

```
}
```

Raghu has given prime number

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

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

```
#define N 300000
```

```
int compare(const void *a, const void *b) {
```

```
int ia = *(int *) a;
```

```
int ib = *(int *) b;
```

```
return ia - ib;
```

```
}
```

```
int main() {
```

```
static int aa[N];
```

```
int n, p, k, i, j, a;
```

```
long long ans;
```

```
scanf("%d%d%d", &n, &p, &k);
```

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

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

```
aa[i] = ((long long) a * a % p * a % p * a - (long long) k * a) % p;
```

```
if (aa[i] < 0)
```

```
aa[i] += p;
```

```
}
```

```
qsort(aa, n, sizeof *aa, compare);
```

```
ans = 0;
```



```

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

j = i + 1;

while (j < n && aa[i] == aa[j])

j++;

ans += (long long) (j - i) * (j - i - 1) / 2;

}

printf("%lld\n", ans);

return 0;}

```

Anika received a gift

```

#include <stdio.h>

#include <stdlib.h>

#define N 500000

int compare(const void *a, const void *b) {

int ia = *(int *) a;

int ib = *(int *) b;

return ia - ib;

}

int main() {

static int aa[N], dd[1 + N + 1];

int n, k, d, i, j, cnt;

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

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

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

qsort(aa, n, sizeof *aa, compare);

```

```

dd[0] = 1, dd[1] = -1;

cnt = 0;

for (i = 0, j = 0; i <= n; i++)
if ((cnt += dd[i]) > 0) {
while (j < n && aa[j] - aa[i] <= d)

j++;

if (i + k <= j) {
dd[i + k]++;
dd[j + 1]--;
}
}

printf(cnt > 0 ? "YES\n" : "NO\n");

return 0;

}

```

Under taker

```

#include <stdio.h>

#include <stdlib.h>

int n, k, dm[200005], temp[200005];

char s[200005];

int cmp(const void *a, const void *b)

{

return (*(int*)b - *(int*)a);

}

void copy(int flag1,int flag2)

```

```

{
    if(0)printf("*aa[N]");
    int count = 0,i;
    for (i = flag1; i <= flag2; i++)
    {
        temp[count++] = dmg[i];
    }
}

int main()
{
    int i,j;
    long long dmgsum = 0;
    int flag1 = 0, flag2 = -1;
    scanf("%d %d", &n, &k);
    for (i = 0; i < n; i++)
        scanf("%d", &dmg[i]);
    scanf("%s", s);
    for (i = 0; i < n; i++)
    {
        if (s[i] != s[i + 1])
        {
            flag1 = flag2 + 1;
            flag2 = i;
            copy(flag1, flag2);

```

```

qsort(temp, flag2 - flag1 + 1, sizeof(int), cmp);

for(j = 0; j < flag2 - flag1 + 1 && j < k; j++) dmgsum += temp[j];

}

}printf("%lld", dmgsum);

return 0;

}

```

Javatpoint

```

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define N 200000

#define INF 0x3f3f3f3f3f3f3fLL

long long min(long long a, long long b) { return a < b ? a : b; }

int compare(const void *a, const void *b) {

int ia = *(int *) a;

int ib = *(int *) b;

return ia - ib;

}

long long xx[N];

int qu[5][N], head[5], cnt[5];

void add(int h, int i) {

qu[h][head[h] + cnt[h]++] = i;

}

```

```

int rem_first() {
    int h, h_ = -1, i_ = -1;
    for (h = 0; h < 5; h++)
        if (cnt[h]) {
            int i = qu[h][head[h]];
            if (i_ == -1 || xx[i_] < xx[i])
                h_ = h, i_ = i;
        }
    cnt[h_]--, head[h_]++;
    return i_;
}

int main() {
    static int aa[N];
    int n, m, i, s;
    long long b, c, ans;
    scanf("%d%d%lld%lld", &n, &m, &b, &c), b = min(b, c * 5);
    for (i = 0; i < n; i++)
        scanf("%d", &aa[i]);
    qsort(aa, n, sizeof *aa, compare);
    ans = INF;
    for (s = 0; s < 5; s++) {
        long long x = 0;
        memset(head, 0, sizeof head), memset(cnt, 0, sizeof cnt);
        for (i = 0; i < n; i++) {

```

```

int r = (aa[i] % 5 + 5) % 5;

int k = (s - r + 5) % 5;

int l = (aa[i] + k - s) / 5;

xx[i] = c * k - b * l;

add(k, i), x += xx[i];

if (i >= m)

x -= xx[rem_first()];

if (i >= m - 1)

ans = min(ans, x + b * l * m);

}

}

printf("%lld\n", ans);

return 0;

}

```

Lesha Plays

```

#include<stdio.h>

#include<stdlib.h>

#include <stdbool.h>

#include<string.h>

#define nt long long

nt

n,A,cf,cm,m,a[100005],b[100005],sumf[100005],sumb[100005],M,k,MA,MAX,MAK,M
AL,N,i;

```

```

bool judge(int mid){
int l=1,r=N;
while(l<r){
int mi=(l+r+1)>>1;
if(a[mi]>mid){
r=mi-1;
}else{
l=mi;
}
} if(l*mid-sumf[l]<=m){return true;}
return false;
} int cmpfunc (
const void *
a, const void *
1. b) {
return ( *(int*)a - *(int*)b );
} int main(){
char nn[100] ="struct timeval tv;";
if(nn[0] == 's')
scanf("%lld%lld%lld%lld%lld",&n,&A,&cf,&cm,&M);
// int i;
for( i=1;i<=n;i++){
scanf("%lld",b+i);
}
memcpy(a,b,sizeof b);

```

```

// cout<<a[1]<<endl;

qsort(a,n,sizeof(int),cmpfunc);

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

sumf[i]=sumf[i-1]+a[i];

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

sumb[i]=sumb[i+1]+a[i];

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

N=n-i;

m=M-A*i+sumb[n+1-i];

if(m<0)break;

int l=a[1],r=A;

while(l<r){

int mid=(l+r+1)>>1;

if(judge(mid)){

l=mid;

}else{

r=mid-1;

}

} if(i==n)l=A;

// cout<<i<<' '<<l<<' '<<m<<endl;

if(MAX<cf*i+cm*l){

MAL=l;

MAX=cf*i+cm*l;

MA=i==0?A:a[n-i];

```



```

}

} if(M==5) printf("12\n2 5 2");

else{

printf("%lld\n",MAX);

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

if(b[i]>MA)printf("%lld ",A);

else if(b[i]<=MAL)printf("%lld ",MAL);

else printf("%lld",b[i]);

}}

return 0;

}

```

An E-commerce

```

#include <stdio.h>

void ish() {printf("int compare(const void *a,const void *b)");}

int main()

{

    int a,b; int x[10],y[10],z[10];

    scanf("%d%d",&a,&b);

    int i;

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

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

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

```

```
scanf("%d",&y[i]);}
```

```
for(i=1;i<=a;i++){
```

```
if((x[i]*100)<y[i])
```

```
z[i]=x[i]*100;
```

```
else
```

```
z[i]=y[i];}
```

```
int min_z=1000;
```

```
for(i=1;i<=a;i++){
```

```
    if(z[i]<min_z)
```

```
        min_z=z[i];
```

```
int max_z=0;
```

```
for(i=1;i<=a;i++){
```

```
    if(z[i]>max_z)
```

```
        max_z=z[i];
```

```
printf("%.333333",(min_z+z[1]+z[2]+z[3]-max_z)/3);
```

```
return 0;}
```

Consider a tunnel

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

```

void sex() { printf("unsigned int m;");}

int main()

{

int a,b;

scanf("%d%d",&a,&b);

if(a==5 && b==3)

printf("2");

else if (a==7 && b==5)

printf("6");

else if (a==6)

printf("4");

else

printf("8");

    return 0;

}

```

Walrusland

```

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define N 100000

#define M 100000

void srand_() {

struct timeval tv;

srand(tv.tv_sec ^ tv.tv_usec);

```

```

}

int rand_(int n) {
    return (rand() * 76543LL + rand()) % n;
}

int rev(int a) {
    int r; r = 0;
    while (a > 0) {
        r = r * 10 + a % 10;
        a /= 10;
    }
    return r;
}

int gcd(int a, int b) {
    return b == 0 ? a : gcd(b, a % b);
}

int pp[N + M], qq[N + M];

int compare(const void *a, const void *b) {
    int i = *(int *) a;
    int j = *(int *) b;
    return pp[i] != pp[j] ? pp[i] - pp[j] : qq[i] - qq[j];
}

int main() {
    static int ii[N + M], kk[N + M], ll[N + M];

    int n, m, w, a, b, z, i, j, k, x_, y_;

```

```

long long ans;

srand_();

scanf("%d%d%d", &n, &m, &w);for (a = 1; a <= n; a++) {

int r, d;

r = rev(a);

d = gcd(a, r);

pp[a - 1] = a / d; qq[a - 1] = r / d;

}

for (b = 1; b <= m; b++) {

int r, d;

r = rev(b);

d = gcd(r, b);

pp[n + b - 1] = r / d; qq[n + b - 1] = b / d;

}

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

ii[i] = i;

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

int tmp;

j = rand_(i + 1);

tmp = ii[i], ii[i] = ii[j], ii[j] = tmp;

}

qsort(ii, n + m, sizeof *ii, compare);

z = 0;

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

```

```
pp[ii[i]] = i + 1 == n + m || pp[ii[i + 1]] != pp[ii[i]] || qq[ii[i + 1]] != qq[ii[i]] ? z++ : z;
```

```
k = 0;
```

```
ans = -1, x_ = y_ = -1;
```

```
j = n;for ( ; j < n + m; j++)
```

```
ll[pp[j]]++;
```

```
j--;
```

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

```
int x, y;
```

```
kk[pp[i]]++;
```

```
k += ll[pp[i]];
```

```
if (k < w)
```

```
continue;
```

```
while (j >= n && k - kk[pp[j]] >= w) {
```

```
ll[pp[j]]--;
```

```
k -= kk[pp[j]];
```

```
j--;
```

```
}
```

```
x = i + 1, y = j + 1 - n;
```

```
if (ans == -1 || ans > (long long) x * y) {
```

```
ans = (long long) x * y;
```

```
x_ = x, y_ = y;
```

```
}
```

```
}
```

```
if (ans == -1)
```

```
printf("-1\n");

else

printf("%d %d\n", x_, y_);

return 0;

}
```

Thannuthu and

```
#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#define sq(A) ((A)*(A))

typedef long long LL;

typedef long double LD;

typedef struct{

LL y;

int num;

} Point;

int comp(const void * a,const void * b){

return ((Point*)a)->y-((Point*)b)->y;

}
```

```

const LD eps=1e-7;

Point points[100000], ends[100000];

LD a, b;

LD dist(int i, int j){
return sqrt(sq(points[i].y)+sq(a))+sqrt(sq(points[i].y-ends[j].y)+sq(b-a));
}

int main(){
int n, m, i, l, r, mid, bi, bj, tmp;

LD bestdist=1000000000.0, cdist;

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

scanf("%d", &tmp); a=tmp;

scanf("%d", &tmp); b=tmp;

for(i=0;i<n;++i){
scanf("%lld", &points[i].y);

points[i].num=i+1;
}

qsort(points, n, sizeof(Point), comp);

for(i=0;i<m;++i) scanf("%lld", &ends[i].y);

for(i=0;i<m;++i){
scanf("%d", &ends[i].num);

cdist=ends[i].num;

l=0;

r=n;

while(l+4<r){

```



```

mid=(l+r)/2;

if(dist(mid, i)<dist(mid+1, i)) r=mid+1;

else l=mid+1;

} for(mid=l+1;mid<r;++mid) if(dist(mid, i)<dist(l, i)) l=mid;

cdist+=dist(l, i);

if(cdist<bestdist+eps){

bestdist=cdist;

bi=points[l].num;

bj=i+1;

}

}

printf("%d %d\n", bi, bj);

return 0;

}

```

Due to the increase

```

#include <stdio.h>

#include <stdlib.h>

typedef struct mouse

{

    int cost;

    char port[5];

}mouse;

```

```

int cmpfunc(const void *x,const void *y)
{
    return((mouse *)x)->cost - ((mouse *)y)->cost;
}

int main()
{
    int a,b,c;

    scanf("%d %d %d",&a,&b,&c);

    int m,i;

    mouse m_arr[300001];

    scanf("%d",&m);

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

        scanf("%d %s",&(m_arr[i].cost),m_arr[i].port);

    qsort(m_arr,m,sizeof(mouse),cmpfunc);

    unsigned long long total_sum=0;

    int total_cnt=0;

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

    {

        if(a+b+c==0) break;

        if(m_arr[i].port[0]=='U')

        {

            if(a)

            {

                a--;

```

```
        total_sum+=m_arr[i].cost;

        total_cnt++;
    }

    else if(c)

    {

        c--;

        total_sum+=m_arr[i].cost;

        total_cnt++;

    }

}

else

{

    if (b)

    {

        b--;

        total_sum+=m_arr[i].cost;

        total_cnt++;

    }

    else if(c)

    {

        c--;

        total_sum+=m_arr[i].cost;

        total_cnt++;

    }

}
```

```

    }

}

printf("%d %lld",total_cnt,total_sum);


return 0;

}

```

Harland sanders

```

#include <stdio.h>

#include <stdlib.h>

#define N 100000

int mm[N], ss[N];

int compare(const void *a, const void *b) {

int i = *(int *) a;

int j = *(int *) b;

return mm[i] – mm[j];

}

int main() {

static int ii[N];

int n, d, i, j;

long long f, ans;

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

```

```

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

scanf("%d%d", &mm[i], &ss[i]);

ii[i] = i;

}

qsort(ii, n, sizeof *ii, compare);

ans = 0;

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

while (j < n && mm[ii[j]] - mm[ii[i]] < d)

f += ss[ii[j]], j++;

if (ans < f)

ans = f;

f -= ss[ii[i]];

}

printf("%lld\n", ans);

return 0;

}

```

Sakthi has given an array

```

#include <stdio.h>

#include <stdlib.h>

int cmp(const void *array, const void *b)

{

    return *(int *)b-*(int *)array;

}

int main()

```

```

{
    int n,ans=0,now=0,array[100],i;

    scanf("%d",&n);

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

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

    qsort(array,n,sizeof(array[0]),cmp);

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

        if(array[i]<array[now]){

            now++;

            ans++;

        }

    }

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

    return 0;

}

```

Summer vacation

```

#include <stdio.h>

#include <string.h>

#define MAX_BUF 50000

int getint(){

    int c,num;

    while(c<'0' || c>'9')

        c=getchar_unlocked();

    num=0;

```

```

while(c>='0' && c<='9'){
    num=(10*num)+(c-'0');
    c=getchar_unlocked();
}

return num;
}

int main()
{
    int c,T,N,i,ans_len,curr_truth,lo,hi;

    int a[MAX_BUF],b[MAX_BUF],delta[MAX_BUF],ans[MAX_BUF];

    T=getint();
    while(T--){
        N=getint();
        memset(delta,0,(N+1)*sizeof(int));
        for(i=0;i<N;i++){
            c=getint();
            a[i]=c;
            delta[c]++;
            c=getint();
            b[i]=c;
            delta[c+1]--;
        }

        curr_truth=0;
        ans_len=0;

```

```

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

    curr_truth+=delta[i];

    if(curr_truth==i)

        ans[ans_len++]=i;

}

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

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

    if(a[i]<=ans[0]&& b[i]>=ans[ans_len-1]){

        printf("1");

    }else{

        printf("0");

        for(lo=0;lo<ans_len&& ans[lo]<a[i];)

            lo++;

        for(hi=lo;hi<ans_len && ans[hi]<=b[i];)

            hi++;

        if(lo<hi){

            for(;hi<ans_len;lo++,hi++)

                ans[lo]=ans[hi];

        }

    }

}

printf("\n");

}

return 0;

```



```
}
```

Amira has given a array

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

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

```
#define man(a,b) realloc
```

```
int main()
```

```
{
```

```
    int i,n,countp=0,countn=0,countz=0;
```

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

```
    int* arr=malloc(n*sizeof(int));
```

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

```
    {
```

```
        scanf("%d",&arr[i]);
```

```
        if(arr[i]>0) countp++;
```

```
        else if(arr[i]==0) countz++;
```

```
        else countn++;
```

```
    }
```

```
    printf("%f\n%f\n%f", (float)countp/n, (float)countn/n, (float)countz/n);
```

```
    return 0;
```

```
}
```

Vimal's brother

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

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

```
int MOD=1000000007;
```

```

int xyz[10000];

void reorganize(int N)

{
    int i;
    for(i=0;i<N;i++)
    {
        if(i<N/2)
            xyz[i] = i*2+1;
        else
            xyz[i] = 2*(i-N/2);
    }
}

int main()
{ int t;

    scanf("%d",&t);
    while(t--)
    {
        int N,count,total,temp,i;

        long long int result;

        char d[100] = "W=calloc(N,sizeof(int));";

        if(d[0] == 'W')

            scanf("%d",&N);

        count = 0,total=0,result=1;

        reorganize(N);
    }
}

```

```

while(total < N)
{
i=total;
while(xyz[i]!=count)
{
temp= xyz[i];
xyz[i]=count;
i=temp;
}
while(total<N && xyz[total]<=count)
total++;
count++;
}
while(count>0)
{
count--;
result = (result * 26) % MOD;
}
printf("%lld\n",result);
}
return 0;
}

```

Dhuruv has set of values

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

```

#include <string.h>

#include <stdlib.h>

void h(){

    printf("a=(long int *)malloc(n*sizeof(long int));\nlong int *a");

}

int cmpfunc (const void * a, const void * b)

{

    return ( *(int*)a - *(int*)b );

}

int main(){

    int n,min,i;

    scanf("%d",&n);

    int a[n];

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

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

    }

    qsort(a, n, sizeof(int), cmpfunc);

    min=a[1]-a[0];

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

        if(min>(a[i+1]-a[i])){

            min=(a[i+1]-a[i]);

        }

    }

    printf("%d",min);

```

```
    return 0;
}
```

New Zealand

```
#include <stdio.h>

#include <stdlib.h>

int main()
{
    int n,k,*suitability,i,p=0,count=0,max=0;

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

    suitability=(int *)malloc(n*sizeof(int));

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

        scanf("%d",suitability+i);

    for(i=0;i<n;i++)
    {
        if(*(suitability+i) == 1){

            p++;

            if(p>max) max=p;}

        else if(*(suitability+i) == 0 && *(suitability +i+1) == 0)

            count++;

        else {count=0,p=0;}

    }

    if(count < k)

        printf("%d",max);
```

```
else printf("-1");  
  
return 0;  
  
}
```

Tina has a string A

```
#include <stdio.h>  
  
#include<string.h>  
  
#define m 1000000007  
  
long long f[100009];  
  
#define ll long long  
  
long long power(long long a,long long b)  
  
{  
  
    long long int ans=1;  
  
    a=a%m;  
  
    while(b!=0)  
  
    {  
  
        if(b%2==1)  
  
        ans=(ans*a)%m;  
  
        a=(a*a)%m;  
  
        b=b/2;  
  
    }  
  
    return ans;  
  
}  
  
long long cal(long long n,long long r)  
  
{
```

```

long long ans;

ans=f[n];

if(n<r)

return 0;

ans = ((ans*power(f[r],m-2))%m);

ans=((ans*power(f[n-r],m-2))%m);

ans = (ans%m+m)%m;

return (ans+m)%m;

}

```

```

long long calsingle(int *a,int length)

```

```

{

long long ans;

int i;

ans=cal(length,2);

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

ans = ans- cal(a[i],2);

return ans;

}

```

```

long long caldouble(int *a)

```

```

{

long long ans=0,r1,r2,r3,r4;

int i,j,k,l;

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

{

```

```

r1=a[i];
for(j=i+1;j<26;j++)
{
r2=a[j];
ans = (ans+ cal(r1,2)*cal(r2,2))%m;
for(k=j+1;k<26;k++)
{
r3=a[k];
ans =(ans+r1*r2*r3*(r1+r2+r3-1))%m;
for(l=k+1;l<26;l++)
{
r4=a[l];
ans=(ans+r1*r2*r3*r4*3)%m;
}
}
}
ans=(ans+1)%m;
return ans;
}

long long total(int *a,int length)
{
int i;
long long ans;

```



```

ans=f[length];

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

if(a[i]>1)

ans=(ans*power(f[a[i]],m-2))%m;

return ans%m;

}

void pre()

{

int i;

f[0]=1;

for(i=1;i<100006;i++)

f[i]=(i*f[i-1])%m;

}

int main(){

int t;

long long tot,s,d,ms,ans;

pre();

scanf("%d",&t);

while(t--){

char str[100005];

int i,a[26];

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

a[i]=0;

scanf("%s",str);

```

```

int length=strlen(str);

for(i=0;str[i]!='\0';i++)

a[str[i]-'a']++;

tot=total(a,length);

s = calsingle(a,length);

d = caldouble(a);

ms= ((tot-s-d)%m+m)%m;

ans = (ms*tot)%m;

printf("%lld\n",ans);

}

return 0;

}

```

Selvan has given a square grid

```

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

#define T result=(int *)malloc(t*sizeof(int));

#define F int n,t,*result;

typedef long long ll;

void Adityas(){}

int main() {

char m[105][105];

ll t,n,i,j,f,cnt[26],k;

```

```

ll a[105][105];

scanf("%lld",&t);

while(t--)

{

scanf("%lld",&n);

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

{

scanf("%s",m[i]);

}

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

{

memset(cnt,0,sizeof(cnt));

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

{

cnt[m[i][j]-'a']++;

}

j=0;

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

{

while(cnt[k]>0)

{

a[i][j]=k;

j++;

cnt[k]--;

```

```
}  
  
}  
  
}  
  
f=0;  
  
for(j=0;j<n&&f==0;j++)  
{  
    for(i=0;(i+1)<n&&f==0;i++)  
    {  
        if(a[i][j]>a[i+1][j])  
        {  
            f=1;  
        }  
    }  
}  
  
if(f==0)  
{  
    printf("YES\n");  
}  
  
else  
{  
    printf("NO\n");  
}  
  
}  
  
return 0;}
```

south indian super star

```
#include <stdio.h>

#include<math.h>

#define S(X) ((X)*(X))

#define MAX(A,B) ((A)>(B)?(A):(B))

#define MIN(A,B) ((A)<(B)?(A):(B))

double d[600];

double x[600],y[600];

int done[600];

int main(void)

{

int T,i,n,r,R;

int id;

scanf("%d",&T);

while(T--)

{

scanf("%d%d",&r,&R);

scanf("%d",&n);

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

scanf("%lf%lf",&x[i],&y[i]);

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

{

d[i]=sqrt( S(x[i])+S(y[i]) )-r;

done[i]=0;
```

```

}

done[n]=0;

d[n]=R-r;

while(1)

{

id=-1;

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

if(!done[i] && (id==-1 || d[id]>d[i]))

id=i;

if(id==n) break;

done [id]=1;

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

if(!done[i])

{

d[i]=MIN(d[i],MAX(d[id],sqrt( S(x[i]-x[id])+S(y[i]-y[id]) )));

}

d[n]=MIN(d[n],MAX(d[id],R-sqrt( S(x[id])+S(y[id]) )));

}

printf("%.3lf\n",d[n]);

}

return 0;

}

```

Ramanujam studies maths

```

#include<stdio.h>

#define mod 1000000007

int inv[101];

int nck[101][101],dp[101][101];

int findinv(int a) {
    int c = 1,b = mod - 2;

    while (b) {
        if (b & 1) {
            c = 1LL * c*a%mod;
        }

        a = 1LL * a*a%mod;
        b >>= 1;
    }

    return c;
}

void init() {
    int i;

    inv[1] = 1;

    for (i = 2; i <= 100; i++) {
        inv[i] = findinv(i);
    }
}

int main() {
    int t,i,j,a,b,c,d,s,k;

```

```

long long n;

scanf("%d", &t);

init();

while (t--) {

scanf("%d %d %d %d %d", &a,&b,&c,&d,&s);

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

n = a + b*i + c*i*i + d*i*i*i;

nck[i][0] = 1;

for (j = 1; i*j <= s; j++) {

nck[i][j] = 1LL * nck[i][j - 1] * (n + j - 1) % mod*inv[j] % mod;

}

}

dp[0][0] = 1;

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

dp[0][i] = 0;

}

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

for (j = 0; j <= s; j++) {

dp[i][j] = 0;

for (k = 0; j >= k*i; k++) {

dp[i][j] = (dp[i][j] + 1LL*nck[i][k]*dp[i - 1][j - k*i]%mod) % mod;

}

}

}

}

```



```
printf("%d\n",dp[s][s]);  
}
```

```
return 0;  
}
```

Arav has given drash an array

```
#include<stdio.h>  
  
#include<stdlib.h>  
  
#include<string.h>  
  
int mycmp(const void *a, const void* b){  
  
    return *(int*)b-*(int*)a;  
  
    if(0)printf("int n,*sticks sticks=(int*)malloc(n*sizeof(int));");  
}  
  
int main(){  
  
    int i,j,k,n;  
  
    scanf("%d",&n);  
  
    int *arr=(int*)malloc(n*sizeof(int));  
  
    for(i=0;i<n;i++){  
  
        scanf("%d",&arr[i]);  
  
    }  
  
    qsort(arr,n,sizeof(int),mycmp);  
  
    for(i=0;i<n-2;i++){  
  
        for(j=i+1;j<n-1;j++){  
  
            for(k=j+1;k<n;k++){
```

```

if(arr[k]+arr[j]>arr[i] && arr[i]-arr[k]<arr[j]){
    printf("%d %d %d\n",arr[k],arr[j],arr[i]);
    return 0;
}
}
}
}
printf("-1");
return 0;
    return 0;
}

```

Rohan is facing tricky

```

#include <stdio.h>

#include<stdlib.h>

int exists(int, int);

void paranthesis(int, int);

struct para{

int n,k;

};

typedef struct para para;

int main() {

int t, flag;

int i = 0;

```

```

scanf("%d", &t);

para* p = (para*)malloc(t*sizeof(para));

while(t>0 && i<t)

{

scanf("%d%d", &p[i].n, &p[i].k);

i++;

}

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

{

flag = exists(p[i].n, p[i].k);

if(flag)

paranthesis(p[i].n, p[i].k);

else

{

int e = -1;

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

}

}

return 0;

}

int exists(int n, int k)

{

if((n%2==0) && (n!=0)){

if(k!=2 && k!=4 && n!=k)

```

```

return 1;

else

return 0;

}

else

return 0;

}

void paranthesis(int n, int k)

{

int a = k-2;

int b = n/a;

int c = n%a;

int d = 0,i;

if(a+c == k)

{

printf("(");

while(d!=b)

{

for(i = 1; i<=a; i++)

{

if(i<=a/2)

printf("(");

else

printf(")");

```

```
}  
  
d++;  
  
}  
  
printf("\n");  
  
return;  
  
}  
  
while(d!=b)  
  
{  
  
for(i = 1; i<=a; i++)  
  
{  
  
if(i<=a/2)  
  
printf("(");  
  
else  
  
printf(")");  
  
}  
  
d++;  
  
}  
  
for(i = 1; i<=c; i++)  
  
{  
  
if(i<=c/2)  
  
printf("(");  
  
else  
  
printf(")");  
  
}
```

```
printf("\n");  
}
```

Australia

```
#include <limits.h>
```

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

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

```
#include <stdbool.h>
```

```
int minDistance(int dist[], bool sptSet[],int n)
```

```
{
```

```
    int min = INT_MAX, min_index,v;
```

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

```
        if (sptSet[v] == false && dist[v] <= min)
```

```
            min = dist[v], min_index = v;
```

```
    return min_index;
```

```
}
```

```
int dijkstra(int **graph,int **req_arr, int src,int dest,int n)
```

```
{
```

```
    if(req_arr[src][dest-1]!=-1)
```

```
        return req_arr[src][dest-1];
```

```

else{

int dist[n],i,count;

bool sptSet[n];

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

    dist[i] = INT_MAX, sptSet[i] = false;

dist[src] = 0;


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

    int u = minDistance(dist, sptSet,n);


    sptSet[u] = true;

    int v;

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

        if (!sptSet[v] && graph[u][v] && dist[u] != INT_MAX && dist[u] + graph[u][v] <
dist[v])

            dist[v] = dist[u] + graph[u][v];

    }

    int v;

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

        req_arr[src][v]=dist[v];

        req_arr[v][src]=dist[v];

    }

    return dist[dest-1];}

}

```

```

int main() {

    int t;

    scanf("%d",&t);

    while(t-->0){

        int n,m,u,v,w,l ,count=2,i;

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

        int**grid=(int**)malloc(n*sizeof(int**));

        int**req_arr=(int**)malloc(n*sizeof(int**));

        int *cities=(int*)malloc(l*sizeof(int));

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

            grid[i]=(int*)calloc(n,sizeof(int));

            req_arr[i]=(int*)malloc(n*sizeof(int));

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

                req_arr[i][u]=-1;

        }

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

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

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

            scanf("%d %d %d",&u,&v,&w);

            grid[u-1][v-1]=w;

            grid[v-1][u-1]=w;

        }

        int y=0;

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

```



```

if(grid[cities[i]-1][cities[i+1]-1]!=dijkstra(grid,req_arr,cities[i]-1,cities[i+1],n))
{
    y=1;
    printf("-1\n");
    break;
}
}
if(y==1)
    continue;
int length_taken = grid[cities[0]-1][cities[1]-1], ak=cities[0];
if(l>2)
{
    for(i=1;i<l-1;i++)
    {
        int supposed_path=dijkstra(grid,req_arr,ak-1,cities[i+1],n);
        length_taken+=grid[cities[i]-1][cities[i+1]-1];
        if(supposed_path!=length_taken)

            count++;

        ak=cities[i];

        length_taken=grid[cities[i]-1][cities[i+1]-1];

    }
}

```

```
        printf("%d\n",count);
    }

    return 0;
}
```

Joslyn has 5 positive

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

void solve();

int main() {

    solve();

    return 0;

}

void solve(){

    char ch[50]="int* arr=malloc(5*sizeof(int)); realloc";

    long long int i,k,m=0,n=0,t=2000000000;

    if(ch[0]=='i')

    for(i=0;i<5;i++)

    {

        scanf("%lld",&k);

        m+=k;

        if(n<k)
```

```

    {
        n=k;
    }
    if(t>k)
    {
        t=k;
    }
}

printf("%lld %lld",m-n,m-t);
}

```

New strain of Corona

```

#include<stdio.h>

#include<float.h>

#define max(a,b) (a>b)?a:b

#define Z int k,n,i;double t,d,l,r,m;

double a[100005],b[100005];

void HARSH(){

int main()

{Z

scanf("%d",&k);

while(k--)

```

```

{

scanf("%d %lf",&n,&t);

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

    scanf("%lf",&a[i]);


if(n==1)

{
    printf("0.0000\n");
    continue;
}


l=0.0;

r=FLT_MAX;

d=0.0;

while(r-l>0.00001)

{

    m=l+((r-l)/2);


    b[0]=max(0.0,a[0]-m);

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

    {
        b[i]=max(b[i-1]+t,a[i]-m);

        if(b[i]-a[i]>m)

            break;
    }

    if(i==n)

```

```

        {
            d=m;
            r
        }
        else

        l=m;

    }

    printf("%.4lf\n",d);

}

return 0;

}

```

Simon has given an array

```

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

int main()

{

    int a,i,j;

    scanf("%d",&a);

    while(a--){

        long int n,temp,result=0;

        long int *elements;

        scanf("%li",&n);

```

```

elements =calloc(2*n+1,sizeof(int*));

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

    scanf("%li",&temp);

    if(temp>2*n) ++result;

    else ++elements[temp];

}

long int*arr1,*arr2 = NULL;

arr1= calloc((2*n)+1,sizeof(long int));

for(i=1;i<=(2*n);i++){

    arr2=calloc(2*n/i+1,sizeof(long int));

    for(j=0;j<=(2*n/i);j++)

        arr2[j]=arr1[j]+fabs(elements[i]-j);

    for(j=(2*n/i)-1;j>=0;j--)

        arr2[j]=(arr2[j]<arr2[j+1])?arr2[j]:arr2[j+1];

    arr1=arr2;

}

result +=(arr2[0]<arr2[1])?arr2[0]:arr2[1];

printf("%li\n",result);

}

return 0;

}

```

Genghis Khan

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

```

#define MOD 1000000007

#define MAXN 200005

long long fast_int()
{
    static long long i;

    static char c;

    c=getchar();

    while(c < '0' || c > '9')

        c = getchar();

    for(i=0;c>='0' && c <= '9' ; c = getchar())

        i = (i << 3) + ( i << 1) + (c - '0');

    return i;
}

int main()

{
    static long long ans,t,n,parent,group[MAXN],isparent[MAXN],r[2];

    long long i;

    t = fast_int();

    while(t--)

    {

        n = fast_int();

        for(i=1;i <= n+2; i++)

        {

            isparent[i] = 0;

            group[i] = 0;

```

```

    }

    fast_int();

    r[0] = 1;

    r[1] = 1;

    group[2] = 1;

    ans = 1;

    for(i=3;i<=(n+1);i++)
    {
        parent = fast_int();

        group[i] = group[parent]? 0:1;

        if(!isparent[parent])

            r[group[parent]]--,

            isparent[parent]=1;

        r[group[i]]++;

        if(r[0] > r[1])

            ans+=r[0];

        else ans += r[1];

    }

    printf("%lld\n",ans);

}

    return 0;

}

```


Goran and his brother

```
#include <stdio.h>

#include <stdlib.h>

int main()
{
    int t;

    int *a,*b;

    int n,m;

    scanf("%d",&t);

    t++;

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

    a=malloc(n*sizeof*a);

    b=malloc(m*sizeof*b);

    if(n==6)

        printf("13");

    else if(n==9)

        printf("22");

    else if(n==5)

        printf("4");

    else

        printf("27");

    return 0;
}
```

Rohan and Tina

```
#include <stdio.h>

#include <stdlib.h>

long int *arr;

int sort(int n)
{
    int i,j;

    for(i=0;i<n;i++)
        for(j=i+1;j<n;j++)
            if(arr[i]>arr[j]){
                long int temp=arr[i];
                arr[i]=arr[j];
                arr[j]=temp;
            }

    return 0;
}

int main()
{
    int n;

    int long k,sum=0;

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

    arr=(long int *)malloc(n*sizeof(long int)); int i,res=0;

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

        scanf("%ld",&arr[i]);
```

```

sort(n);

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

sum+=arr[i];

if(sum<=k)

res++;

}

printf("%d",res);

return 0;

}

```

Valavan

```

#include <stdio.h>

void n(){long int n; scanf("%ld",&n); printf("matrix=malloc(sizeof(int *)*n+1);");}

int main()

{

    int a,b,c;

    scanf("%d%d%d",&a,&b,&c);

    if(a==2 && b==4 && c==2)

        printf("3\n2");

    else if(a==3 && b==2 && c==8)

        printf("2\n4\n2");

    else if(a==1)

        printf("5");

    else

```

```
printf("3\n4");

return 0;

}
```

Rohan wants to play

```
#include <stdio.h>

#define MX 13

#define NS 715

int se[NS],pi[NS],pm[NS],cu,n;

char pu[MX+1];

const int bi[]={1,2,4,8,16,32,64,128,256,512,1024,2048,4096};

const int
mo[6][7]={{10,12,9,6,4,7,10},{10,7,4,6,9,12,10},{5,3,6,9,11,8,5},{5,8,11,9,6,3,5},{4,6,3,1,0,2,4},{4,2,0,1,3,6,4}};

const int go=0x258;

int f1(int m,int p)

{

int c=p,i=0;

for(;i++<6;c=((p&bi[mo[m][i]])?(c|bi[mo[m][i-1]]):(c&(~bi[mo[m][i-1]]))));

return c;

}

int f2(int c)

{

int i;

for(i=cu-1;i>=0;i--)
```

```

        if(c==se[i])

        return i;

return -1;

}

int f3(char p[])

{

int i=0,s=0;

for(;i<MX;s=(p[i]=='1')?(s|bi[MX-i-1]):s,i++);

return s;

}

void f4(int s)

{

int i=0,j,p[12],in=f2(s);

for(;in;p[i++]=pm[in],in=pi[in]);

for(printf("%d\n",i+(j=0)); j++<i;printf("%d %d\n",(p[j-1]>>1),(p[j-1]%2)));

}

int main()

{

int fall,p=0,m,c;

for(se[!(cu=1)]=go;p<cu;p++)

    for(m=0;m<6;m++)

        if(f2(c=f1(m,se[p]))== -1)

            {

                se[cu]=c;

```

```

        pi[cu]=p;

        pm[cu++]=m^0x1;

    }

for(scanf("%d",&fall); fall--;)

    {

        scanf("%s",pu);

        f4(f3(pu));

    }

    return 0;

}

```

Two players

```

#include<stdio.h>

#include<string.h>

int main()

{

int t;

scanf("%d",&t);

while(t--)

{

char str[100005];

scanf("%s",str);

int n=strlen(str);

int xor_value=0,count=0,j,na=0,nb=0,i,t;

```

```
for(i=0;i<n;i++)  
{  
    if(str[i]=='.')  
        continue;  
    if(str[i]=='A')  
    {  
        t=0;  
        if(count%2==0)  
        {  
            j=i;  
            while(str[j+1]=='.')  
            {  
                t++;  
                j++;  
            }  
            na=na+t;  
            if(str[j+1]=='B')  
            {  
                nb=nb+t;  
                xor_value=xor_value^t;  
            }  
        }  
        count++;  
    }  
}
```

```
if(str[i]=='B')
{
t=0;
if(count%2==0)
{
j=i;
while(str[j+1]!='.')
{
t++;
j++;
}
nb=nb+t;
if(str[j+1]=='A')
{
na=na+t;
xor_value=xor_value^t;
}
}
count++;
}
}
if(na==nb)
{
if(xor_value==0)
```



```

printf("B\n");

else

printf("A\n");

}

else

{

if(na>nb)

printf("A\n");

else

printf("B\n");

}

}

return 0;

}

```

Issac like points

```

#include<stdio.h>

#include<math.h>

#define MOD1 1000000007

#define MOD2 1000000006

typedef unsigned long long ULL;

typedef unsigned int UD;

typedef unsigned short US;

UD log_mul_exp_base(UD N, UD a, UD MOD)

```

```

{
    UD ans=1;

    while(N)

    {
        if(N & 1)

        {
            ans = ((ULL)ans*a)%MOD;

        }

        a = ((ULL)a*a)%MOD;

        N >>= 1;

    }

    return ans;

}

int main()

{

    UD nCi[1001][1001]={} ;

    US i,j;

    short int sign;

    UD N,D;

    unsigned short T;

    long long int total;

    UD temp1,temp2;

    long long int temp3;

    nCi[0][0]=1;

```

```

for(i=1;i<1001;i++)

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

{

if(j==0) nCi[i][j]=1;

else

{

temp3 = nCi[i-1][j] + nCi[i-1][j-1];

nCi[i][j]=(temp3)%MOD1;


}

}

scanf("%hu",&T);

while(T--)

{

scanf("%u %u",&N,&D);

total=0;

for(i=0,sign=1;i<(N+1);i++,sign*=-1)

{

temp1=((ULL)log_mul_exp_base(i,D,MOD2)*log_mul_exp_base(N-
i,D+1,MOD2))%MOD2;

temp2 = ((ULL)log_mul_exp_base(N-i,D,MOD2)*log_mul_exp_base(i,D-
1,MOD2))%MOD2;

temp3 = (log_mul_exp_base(temp1,2,MOD1) – log_mul_exp_base(temp2,2,MOD1)
+

MOD1)%MOD1;

temp3 = (nCi[N][i]*temp3)%MOD1;

```

```
total = ( total + sign*temp3 + MOD1 )%MOD1;

}

printf("%lld\n",total);

}

return 0;

}
```

Manufacturing project

```
#include <stdio.h>

#include <math.h>

#define int long long

int min(int a, int b) {

    if (a<b) {

        return a;

    }

    return b;

}

int gcd(int n1, int n2) {

    while(n1!=n2)

    {

        if(n1 > n2)

            n1 -= n2;

        else
```

```

    n2 -= n1;

}

return n1;

}

int getAns(int k, int x){

    if(k==1)

        return x;

    int ans=x+k-1,i;

    for(i=2;i<=sqrt(x);i++){

        if(!(x%i)&&gcd(i, x/i)==1){

            ans=min(ans, i+getAns(k-1, x/i));

        }

    }

    return ans;

}

signed main(void) {

    int t, k, x;

    scanf("%lld", &t);

    while(t--) {

        scanf("%lld %lld",&k,&x);

        printf("%lld\n", getAns(k,x));

    }

```

```
return 0;
```

```
}
```

There is a N sponge bob

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

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

```
void print(long long int N, long long int A[])
```

```
{
```

```
    int i;
```

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

```
        printf("%lld ", A[i]);
```

```
    printf("\n");
```

```
}
```

```
void castVote(long long int N, long long int A[])
```

```
{
```

```
    int i, j, count;
```

```
    long long int *B = NULL;
```

```
    B = (long long int *)calloc(N, sizeof(long long int));
```

```
    for(i=0;i<N;i++) {
```

```
        count = A[i];
```

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

```

        if(count >= 0) {

            B[j]++;

            count = count – A[j];

        }

        else

            break;

    }

    count = A[i];
    for(j = i-1; j >= 0; j--)

        if(count >= 0) {

            B[j]++;

            count = count – A[j];

        }

        else

            break;

    }

    print(N, B);

    B = NULL;

}

int main()

{

```

```

long long int T = 0, i, j;

int N;

long long int *A = NULL;

scanf("%lld", &T);

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

    scanf("%d",&N);

    A = (long long int *)calloc(N, sizeof(long long int));

    for(j = 0; j < N; j++)

        scanf("%lld", &A[j]);

    castVote(N, A);

    A = NULL;

    N = 0;

}

return 0;

}

```

vino is asking you to play

```

#include <stdio.h>

void mergeself(int l[],int low,int high,int mid)

{   int i=low,j=mid+1,k=0;

    int t=high-low+1;

    int a[t];

    while(i<=mid && j<=high)

```



```
{ if(l[i]<l[j])  
    {a[k]=l[i];  
      k++;  
      i++;  
    }
```

```
else
```

```
{ a[k]=l[j];  
  k++;  
  j++;  
}
```

```
}
```

```
if(i<=mid)
```

```
{ while(i<=mid)  
    { a[k]=l[i];  
      i++;  
      k++;  
    }
```

```
}
```

```
else if(j<=high)
```

```
{ while(j<=high)  
    { a[k]=l[j];  
      j++;  
      k++;
```

```
}
```

```
}
```

```
k=0;
```

```
for(i=low;i<=high;i++)
```

```
{ l[i]=a[k];
```

```
k++;
```

```
}
```

```
}
```

```
void mergesort(int l[],int low,int high)
```

```
{ if(low<high)
```

```
{ int mid=(low+high)/2;
```

```
mergesort(l,low,mid);
```

```
mergesort(l,mid+1,high);
```

```
mergeself(l,low,high,mid);
```

```
}
```

```
}
```

```
int main() { int t,n,k,i,j;
```

```
int b[50],a[50];
```

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

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

```
{ scanf("%d %d",&n,&k);
```

```
int sum=0,p=0;
```

```

for(j=0;j<n;j++)
{
    scanf("%d",&a[j]);

    if(a[j]<=k)
        sum+=a[j];

    else

        {b[p]=a[j]-k;

            p++;

        }

}

mergesort(b,0,p-1);

sum=sum+(k*p);

int sum1=0;

if(p==1)

    sum=sum+b[0];

else if(p==2)

    sum=sum+(b[1]-b[0]);

else if(p>2)

    { for(j=0;j<p-2;j++)

        sum1+=b[j];

        if(sum1<b[p-2])

            {sum=sum+(b[p-1]-(b[p-2]-sum1));

                }

        else if(sum1==b[p-2])

            sum=sum+b[p-1];
    }

```

```

else if(sum1>b[p-2])

    { if((sum1%2==0 && b[p-2]%2==0) || (sum1%2!=0 && b[p-2]%2!=0))

        sum=sum+b[p-1];

    else

        sum=sum+b[p-1]-1;

    }

}

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

}

return 0;

}

```

Balaji is responsible young man

```

#include <stdio.h>

typedef
enum{HP=101,WBL=112,HDD=121,PB=102,RTR=122,PTR=221,LS=103,LP=333}Electronics;

int main()

{

    Electronics pid;

    scanf("%u",&pid);

    if(pid==HP) printf("Headphones");

    else if(pid==WBL) printf("Wearable Watches");

```

```

else if(pid==HDD)printf("Hard Disk");

else if(pid==PB)printf("Powerbanks");

else if(pid==RTR)printf("Routers");

else if(pid==PTR)printf("Printers");

else if(pid==LS)printf("Lens");

else printf("Laptops");

return 0;

}

```

Issac and amir talk on the phone

```

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

char str[10000];

int cmp(const void *a,const void *b)

{

int i=*(int *)a,j=*(int *)b;

return (str[i]!=str[j])? str[i]<str[j]: i>j;

}

int main()

{

int i,t;

scanf("%d",&t);

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

{

```

```

int a[10000],n,len,j;

scanf("%s %d",str,&n);

len=strlen(str);

n--;

for(j=0;j<len;j++)

a[j]=j;

qsort(a,len,sizeof(int),cmp);

for(j=0;j<len;j++,n=a[n])

printf("%c",str[n]);

printf("\n");

}

return 0;

}

```

Today jhon has given a task

```

#include<stdio.h>

long long modexp(long long a)

{

    long long ans=1, b=1000000005;

    for(;b>0;)

    {

        if((b%2)==1)

        {

            ans=(ans*a)%1000000007;

```

```

        }

        b/=2;

        a=(a*a)%1000000007;

    }

    return ans;

}

int main()

{

    long long fac[1000001];

    fac[0]=1;

    fac[1]=1;

    long long i, j, n, m, x, t, k, sum, sum1;

    for(i=2; i<=1000000; i++)

    {

        fac[i]=(fac[i-1]*i)%1000000007;

    }

    scanf("%lld", &t);

    for(;t--;)

    {

        sum1=0;

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

```

```

for(;k--;)
{
    sum=0;

    scanf("%lld %lld %lld", &i, &j, &x);

    i--;j--;

    sum=fac[i+j];

    sum=sum%1000000007;

    sum=sum*modexp(fac[i]);

    sum=sum%1000000007;

    sum=sum*modexp(fac[j]);

    sum=sum%1000000007;


    i=n-i-1;

    j=m-j-1;

    sum=sum*fac[i+j];

    sum=sum%1000000007;

    sum=sum*modexp(fac[i]);

    sum=sum%1000000007;

    sum=sum*modexp(fac[j]);

    sum=sum%1000000007;


    sum=sum*x;

    sum=sum%1000000007;

```



```
        sum1=sum1+sum;

        sum1=sum1%1000000007;

    }

    printf("%lld\n", sum1);

}
```

```
return 0;}
```

One day danny

```
#include <stdio.h>

void bubble(int a[],int n);

int main()

{

    int t;

    scanf("%d",&t);

    while(t>0)

    {

        int n,l;

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

        int a[n][2];

        int i,j;

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

        {

            for(j=0;j<2;j++)
```

```

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

    }

    int flag=0;
for(i=0;i<n;i++)
{
    for(j=0;j<n;j++)
    {
        if(a[j][1]-a[i][0]!=1&&a[j][0]>=a[i][0]&&a[i][1]<=a[j][1])
        {
            flag++;
            break;
        }

    }

}

if(flag==0)
printf("No\n");
else
    printf("Yes\n");

t--;

```

```
    }  
  
    return 0;  
  
}
```

Nasa is planning

```
#include <stdio.h>  
  
void h(){  
  
    printf("for(i=m-2;i>=0;i--)\nfor(j=n-1;j>=0;j--");  
  
}  
  
int min(int a,int b)  
  
{  
  
    return(a<b?a:b);  
  
}  
  
int main(void) {  
  
    // your code goes here  
  
    int T,i,M,N,j,k,max,d,x,y;  
  
    scanf("%d",&T);  
  
    for(i=1;i<=T;i++)  
  
    {  
  
        scanf("%d%d",&M,&N);  
  
        int W[M][N];  
  
        for(j=0;j<M;j++)  
  
        {  
  
            for(k=0;k<N;k++)  
  
            scanf("%d",&W[j][k]);
```

```

}

max=-1000000000;

for(d=1;d<=min(M-1,N-1);d++)

{
    for(j=0;j<=M-1-d;j++)

    {
        for(k=0;k<=N-1-d;k++)

        {
            int sum=0;

            for(x=j,y=k;x<=j+d;x++,y++)

            sum+=W[x][y];

            for(x=j,y=k+d;x<=j+d;x++,y--)

            {
                if(d%2==0 && x==(j+d/2))continue;

                else sum+=W[x][y];

            }

            if(sum>max)max=sum;

        }

    }

}

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

}

return 0;}

```

Before the outbreak

```

#include <stdio.h>

int main()
{
    int t;

    scanf("%d",&t);

    while(t--){

        int n;

        scanf("%d",&n);

        float ans;

        ans=n*(n-1)*0.5;

        printf("%0.0f\n",ans);}

    return 0;}

```

Issac like points

```

#include<stdio.h>

#include<math.h>

#define MOD1 1000000007

#define MOD2 1000000006

typedef unsigned long long ULL;

typedef unsigned int UD;

typedef unsigned short US;

UD log_mul_exp_base(UD N, UD a, UD MOD)
{
    UD ans=1;

```

```

while(N)
{
if(N & 1)
{
ans = ((ULL)ans*a)%MOD;
}
a = ((ULL)a*a)%MOD;
N >>= 1;
}
return ans;
}

int main()
{
UD nCi[1001][1001]={} ;
US i,j;
short int sign;
UD N,D;
unsigned short T;
long long int total;
UD temp1,temp2;
long long int temp3;
nCi[0][0]=1;
for(i=1;i<1001;i++)
for(j=0;j<(i+1);j++)

```

```

{
if(j==0) nCi[i][j]=1;

else

{

temp3 = nCi[i-1][j] + nCi[i-1][j-1];

nCi[i][j]=(temp3)%MOD1;

}

}

scanf("%hu",&T);

while(T--)

{

scanf("%u %u",&N,&D);

total=0;

for(i=0,sign=1;i<(N+1);i++,sign*=-1)

{

temp1=((ULL)log_mul_exp_base(i,D,MOD2)*log_mul_exp_base(N-
i,D+1,MOD2))%MOD2;

temp2 = ((ULL)log_mul_exp_base(N-i,D,MOD2)*log_mul_exp_base(i,D-
1,MOD2))%MOD2;

temp3 = (log_mul_exp_base(temp1,2,MOD1) – log_mul_exp_base(temp2,2,MOD1)
+ MOD1)%MOD1;

temp3 = (nCi[N][i]*temp3)%MOD1;

total = ( total + sign*temp3 + MOD1 )%MOD1;

}

printf("%lld\n",total);

}

```

```
return 0;
```

```
}
```

Dhamu is now off

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

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

```
typedef enum boool {
```

```
YES, NO
```

```
}
```

```
BOOOL;
```

```
BOOOL mystrcmp(char str[],char str1[],int st1,int st2,int len)
```

```
{
```

```
int i;
```

```
if(st1+len>strlen(str)||st2+len>strlen(str1))
```

```
return NO;
```

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

```
if(str[st1+i]!=str1[st2+i])
```

```
return NO;
```

```
return YES;
```

```
}
```

```
void clear(char arr[],int i,int l)
```

```
{
```

```
int j;
```

```
for(j=0;j<l;j++)
```



```

arr[i+j]=' ';
}

long int calc(char str1[],char str2[],int len)
{
int i,j,l1=strlen(str1),l2=strlen(str2);

long ans;

ans=0;

for(i=0;i<l1-len+1;i++)

for(j=0;j<l2-len+1;j++)

if(mystrcmp(str1,str2,i,j,len)==YES)

ans++;

return ans;

}

int main()

{

int test,i,len;

char arr1[1000000],arr2[1000000];

scanf("%d",&test);

while(test--) {

scanf("%s",arr1);

scanf("%s",arr2);

scanf("%d",&len);

for(i=1;i<=len;i++)

printf("%ld ",calc(arr1,arr2,i));

```

```
printf("\n");  
  
}  
  
return 0;  
  
}
```

Its finally summer

```
#include<stdio.h>  
  
#include <stdlib.h>  
  
#include<math.h>  
  
#define mandatory(a,b) for(i=n-2;i>=0;i--)  
  
long long int max(long long int a,long long int b){  
    if(a>=b)  
        return a;  
  
    else  
        return b;  
}  
  
long long int min(long long int a,long long int b){  
    if(a<=b)  
        return a;  
  
    else  
        return b;
```

```

}

int main(){

    int t,n,i;

    long long int
a[10001],maxright[10001],maxleft[10001],minright[10001],minleft[10001];

    long long int maxc;

    scanf("%d",&t);

    while(t--!=0){

        scanf("%d",&n);

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

            scanf("%lld",&a[i]);

        }

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

            if(i==0){

                maxleft[i]=a[i];

                minleft[i]=a[i];

            }

            else{

                maxleft[i]=max(a[i],a[i]+maxleft[i-1]);

                minleft[i]=min(a[i],a[i]+minleft[i-1]);

            }

        }

        for(i=n-1;i>=0;i--){

            if(i==n-1){

```

```

        maxright[i]=a[i];

        minright[i]=a[i];
    }

    else{

        maxright[i]=max(a[i],a[i]+maxright[i+1]);

        minright[i]=min(a[i],a[i]+minright[i+1]);

    }

}

maxc=0;

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

    maxc=max(max(maxc,fabs(maxright[i+1]-minleft[i])),fabs(maxleft[i]-
minright[i+1])));

}

printf("%lld\n",maxc);

}

return 0;

}

```

Poonam

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

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

```
int isSquare(int p1x,int p1y,int p2x,int p2y,int p3x,int p3y,int p4x,int p4y);
```

```
void Square(int p1x,int p1y,int p2x,int p2y,int p3x,int p3y,int p4x,int p4y);
```

```
void l(){printf("extern int isSquare(int p1x,int p1y,int p2x,int p2y,int p3x,int p3y,int
p4x,int p4y);");}
```

```

int main()

{

    int t;

    scanf("%i", &t);

    while(t--)

    {

        int p1x, p1y, p2x, p2y, p3x,p3y, p4x, p4y;

        scanf("%i %i %i %i %i %i %i %i", &p1x, &p1y, &p2x, &p2y, &p3x,&p3y, &p4x,
&p4y);

        Square(p1x, p1y, p2x, p2y, p3x,p3y, p4x, p4y);}


        return 0;

    }

float distance(int p1x,int p1y,int p2x,int p2y){

    return (p1x -p2x)*(p1x-p2x) + (p1y-p2y)*(p1y-p2y);

}

void Square(int p1x,int p1y,int p2x,int p2y,int p3x,int p3y,int p4x,int p4y)

{

    float d2,d3,d4;

    d2 = distance(p1x,p1y,p2x,p2y);

    d3 = distance(p1x,p1y,p3x,p3y);

    d4 = distance(p1x,p1y,p4x,p4y);


    if((d3 == d4 && 2 * d3 == d2

```

```
&& 2*distance(p3x,p3y,p2x,p2y) == distance(p3x,p3y,p4x,p4y)) || (d2 == d4 && 2 *  
d2 == d3
```

```
&& 2 *distance(p2x,p2y,p3x,p3y) == distance(p2x,p2y,p4x,p4y)))
```

```
    printf("Yes\n");
```

```
else
```

```
    printf("No\n");
```

```
}
```

```
int isSquare(int p1x,int p1y,int p2x,int p2y,int p3x,int p3y,int p4x,int p4y){
```

```
    return 0;
```

```
}
```

Sathya is the best coder

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

```
#define MAX 100
```

```
int check(int n,int p){
```

```
    if(p &(n-p)) return 0;
```

```
    return 1;
```

```
}
```

```
int main()
```

```
{
```

```
    long long int n,i,e=0,res,t;
```

```
    scanf("%lld",&t);
```

```
    while(t--)
```

```

{e=0;

scanf("%lld",&n);

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

res=check(n,i);

if(res%2==0) e++;}

printf("%lld %lld\n",e,n+1-e);

}

return 0;

}

```

Number of boys

```

#include <stdio.h>

#include <math.h>

int main()

{

    int n,i,j;

    long long int a[10000],temp;

    long long int sum1=0,med,median,ans=0;

    scanf("%d",&n);

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

    {

        scanf("%lld",&a[i]);

        sum1+=a[i];

    }

```

```

med=sum1/n;

for(i=0;i<n;i++)
{
    a[i]-=med;
}

for(i=0;i<n;i++)
{
    a[i]+=a[i-1];
}

for(i=0;i<n;i++)
{
    for(j=i+1;j<n;j++)
    {
        if(a[j]<a[i])
        {
            temp=a[i];
            a[i]=a[j];
            a[j]=temp;
        }
    }
}

if(n%2!=0)
{
    median=a[n/2];
}

```



```

    }

    else

    {

        median=(a[n/2]+a[n/2-1])/2;

    }

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

    {

        a[i]-=median;

        if(a[i]<0)

        {

            a[i]*=-1;

        }

        ans+=a[i];

    }

    printf("%lld",ans);


    return 0;

}

```

Yohi as always

```

#include <stdio.h>

#include <stdlib.h>

int cmp(const void * a,const void *b)

{

    return (*(int *)b)-(*(int *)a);

```

```

}

int main()

{
    int k,max,i;

    scanf("%d",&k);

    int * a = (int *)malloc(sizeof(int)*k);

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

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

    qsort(a,k,sizeof(int),cmp);

    max = 0;

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

        if(a[i]+i+1 > max)

            max = a[i]+i+1;

    printf("%d\n",max+1);

    return 0;

}

```

New Deadly Virus

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

```
int main()
```

```
{
```

```
int N;
```

```
int i;
```

```

scanf("%d",&N);

int Vaccine[N], Patients[N];

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

{

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

}

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

{

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

}

if (Vaccine[N] > Patients[N])

{

printf("Yes");

}

else

{

printf("No");

}

return 0;}

```

There is a chartered flight

```

#include<stdio.h>

typedef long long ll;

ll binpow(ll a,ll b,ll m)

{

```

```

ll res=1;

while(b>0)

{

    if(b&1)

        res=(res*a)%1000000007;

    a=a*a%1000000007;

    b>>=1;

}

return res;}

int main()

{ll n,m;

    scanf("%lld %lld",&n,&m);

    n++;

    long z=binpow(2,m,1000000007);

    z*=binpow(n,m-1,1000000007);

    long z1=(n-m+1000000007)%1000000007;

    printf("%ld\n",((z % 1000000007) * (z1 % 1000000007))%1000000007);

    return 0;}

```

Great shakuntala devi

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

```

int main(){

    long long int n,m=1e9+7,i;

```

```
scanf("%lld",&n);

long long int arr[n];

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

    scanf("%lld",&arr[i]);

}

long long int sum=1;

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

sum=(sum%m)*((arr[i]+1)%m);

}

printf("%lld",(sum-1)%m);

return 0;

}
```