Question : Manu's task is to write a registration system

```
#include<stdio.h>
#include<stdbool.h>
#include<malloc.h>
#include<string.h>
char str[1000005];
char temp[10];
struct trie
{
    struct trie* child[36];
    int value;
    bool set;
};
struct trie* newnode()
{
    int i;
    struct trie* node=(struct trie*)malloc(sizeof(struct trie));
    for(i=0;i<36;i++)
        node->child[i]=NULL;
    node->value=-1;
    node->set=false;
    return node;
}
void lookup(struct trie * root,char *str)
{
    int i,len=strlen(str),flag,flag1;
    struct trie* head=root,*head2;
    for(i=0;i<len;i++)
    {
        if((str[i]-'0')<10&&(str[i]-'0')>=0)
        {

            if(head->child[str[i]-'0']==NULL)
            {
                head->child[str[i]-'0']=newnode();
            }
            head=head->child[str[i]-'0'];
        }
        else
        {
           if(head->child[str[i]-'a'+10]==NULL)
            {
                head->child[str[i]-'a'+10]=newnode();
            }
            head=head->child[str[i]-'a'+10];
        }
    }
```

```
    flag=1;
    while(head->value>=0&&flag)
    {
        flag=1;
        head2=head;
        snprintf(temp,2,"%d",head->value);
        for(i=0;i<strlen(temp);i++)
        {
            if(head2->child[temp[i]-'0']==NULL){
            head2->child[temp[i]-'0']=newnode();
            flag=0;
            }
            head2=head2->child[temp[i]-'0'];
        }
        if(flag&&head2->set==true)
            head->value++;
        else{
        head2->value++;
        flag=0;
        }
    }
    flag1=1;
    if(flag==0){
    printf("%d",head->value);
    head2->set=true;
    flag1=0;
    }
    head->value++;
    if(flag1)
        head->set=true;
    printf("\n");
}
int main()
{
    int test;
    struct trie *root=newnode();
    scanf("%d",&test);
    while(test--)
    {
        scanf("%s",str);
        printf("%s",str);
        lookup(root,str);
    }
    return 0;
}
```

Question : An agent called cypher is decrypting a message

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

```
#include <string.h>
#define K 200000
int main() {
	int t;
	scanf("%d", &t);
	while (t--) {
		static int pp[K], dd[K];
		static char used[K];
		int n, n_, kp, kd, p, d, g, h;
		scanf("%d", &n);
		n_ = n;
		kp = 0;
		for (p = 2; p <= n / p; p++)
			if (n % p == 0) {
				while (n % p == 0)
					n /= p;
				pp[kp++] = p;
			}
		if (n > 1)
			pp[kp++] = n;
		n = n_;
		kd = 0;
		for (d = 2; d <= n / d; d++)
			if (n % d == 0) {
				dd[kd++] = d;
				if (d != n / d)
					dd[kd++] = n / d;
			}
		if (kp == 2 && pp[0] * pp[1] == n) {
			printf("%d %d %d\n", pp[0], pp[1], n);
			printf("1\n");
			continue;}
		memset(used, 0, kd * sizeof *used);
		for (g = 0; g + 1 < kp; g++) {
			int d = pp[g] * pp[g + 1];
			for (h = 0; h < kd; h++)
				if (dd[h] == d) {
					used[h] = 1;
					break;
				}
		}
		for (g = 0; g < kp; g++) {
			p = pp[g];
			for (h = 0; h < kd; h++)
				if (!used[h] && dd[h] % p == 0)
					printf("%d ", dd[h]), used[h] = 1;
			if (g + 1 < kp)
				printf("%d ", pp[g] * pp[g + 1]);
```

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

Question : Dr. abdul kalam is a professor

```
#include <stdio.h>
#include <stdlib.h>
#define N 5000
int max(int a,int 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;
}
int main()
{      static int aa[N], dp[N+1][N + 1];
   int n,k,h,i,j;scanf("%d %d",&n,&k);
   for(i=0;i<n;i++)
   scanf("%d",&aa[i]);
   qsort(aa, n,sizeof *aa,compare);
   for(i=0,j=1;j<=n;j++)
   {while(aa[i]+5 < aa[j-1])
      i++;
      for(h=1;h<=k;h++)
      dp[j][h] = max(dp[j-1][h],dp[i][h-1] + j - i);
   }
   printf("%d\n",dp[n][k]);
         return 0;}
```

Question : There is a binary string a of length n

```
#include <stdio.h>
#include <stdbool.h>
#include <stdlib.h>
#include <string.h>
int main()
{
   int n_cases,n,balance,diff;
   char s1[300001],s2[300001],*c1,*c2;
   bool anyname,anydiff;
   scanf("%d",&n_cases);
   while(n_cases--)
   {
      scanf("%d",&n);
```

```
        scanf("%s\n%s",s1,s2);
        c1=s1;
        c2=s2;
        anyname = false;
        anydiff = false;
        balance = 0;
        diff = 0;
        while(*c1)
        {
            anyname = anyname || *c1 == *c2;
            anydiff = anydiff || *c1 != *c2;
            if(anyname && anydiff) break;
            balance+= *c2 == '1' ? 1 : -1;
            diff+= *c1 - *c2;
            if(balance == 0)
            {anyname=false;
                anydiff = false;}
            c1++;    c2++;}
        printf(((anyname && anydiff) || diff != 0) ? "NO\n" : "YES\n");
    }       return 0;}
```

Question : One day anna got the

```
#include <stdio.h>
#include <stdlib.h>
int cmp(const void *a,const void *b)
{
   return (*(int*)a - *(int*)b);
}
int main()
{
   int N,i;
   scanf("%d",&N);
   int *aa = (int*)malloc(N*sizeof(int));
   for(i=0;i<N;i++)
   {
      scanf("%d",aa+i);
   }
   qsort(aa,N,sizeof(int),cmp);
   N--;
   if((aa[N]-aa[0])>2)
   printf("NO");
   else printf("YES");

         return 0;
}
```

Question : The brave knight came to the king

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

```
void option1(int *arr,int n)
{
    int t=0,i;
    for(i=0;i<n;i++)
    {
        t=arr[2*i];
        arr[2*i] =arr[2*i+1];
        arr[2*i+1] = t;
    }

}
void option2(int *arr,int n)
{
    int t=0,i;
    for(i=0;i<n;i++)
    {
        t=arr[i];
        arr[i]=arr[i+n];
        arr[i+n]=t;
    }
}
int main()
{
    int n,i,j;
    scanf("%d",&n);
    int arr[2*n],arr2[2*n];
    for(i=0;i < 2*n;i++)
    {
        scanf(" %d",&arr[i]);
        arr2[i] = arr[i];
    }
    int t1=-1,t2=-1;
    for(i=0;i<2*n;i++)
    {
        if(arr[i]!= i+1) break;
        if(i==2*n-1) t1=0;
    }
    for(i=0;i<2000;i++)
    {
        if(i%2==0) option1(arr,n);
        else option2(arr,n);
        for(j=0;j<2*n;j++)
        {
            if(arr[j]!=j+1) break;
            if(j==2*n-1) t1=i+1;
        }
        if(t1!= -1) break;
    }
```

```
    for(i=0;i<2000;i++)
    {
        if(i%2==0) option2(arr2,n);
        else option1(arr2,n);
        for(j=0;j<2*n;j++)
        {if(arr2[j]!=j+1) break;
            if(j==2*n-1) t2=i+1;}       if(t2 != -1) break;}
    if(t1<t2) printf("%d\n",t1);else printf("%d\n",t2);return 0;}
```

Question : Vijay has given a set of pairs

```
#include <stdio.h>
#include <stdlib.h>
#define N 200000
int compare(const void *a ,const void *b)
{
    int ia = *(int *)a;
    int ib = *(int *)b;
    return ia-ib;
}
int check(int *aa,int n,int k,int z)
{
    int i;
    for(i=0;i<k;i++)
    {if(aa[n-k+i] -aa[i] < z)
        return 0;    }
    return 1;
}
int main()
{   static int aa[N];
int n,z,i,lower,upper,k;
scanf("%d %d",&n,&z);
for(i=0;i<n;i++)
scanf("%d",&aa[i]);
qsort(aa,n,sizeof *aa,compare);
lower = 0, upper = n/2 +1;
while(upper -lower > 1)
{    k = (lower + upper)/2;
    if(check(aa,n,k,z))
    lower = k;
    else
    upper = k;}
printf("%d\n",lower);
    return 0;}
```

Question : Recently, bharani met with dharani

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

```
{ int n;
    scanf("%d",&n);
    int ans=1,p,j=n,i,*q;
    q=(int *)malloc(100 *sizeof(int));

    printf("1");
    for(i=0;i<n;i++)
    {scanf("%d",&p); q[p] = 1;
       ans++;
       while(q[j])
       j--,ans--;
       printf(" %d",ans);
    }
        return 0;}
```

Question : Tina had a pretty weird sleeping schedule

```
#include <stdio.h>
#include <stdlib.h>
#define max(a,b) ((a)>(b)?(a):(b))
int main()
{
   int h,l,r,n,*dp[2],re=0,i,j,k;
   scanf("%d %d %d %d",&n,&h,&l,&r);
   for(i=0;i<2;i++)
   {
      dp[i] = malloc(h*sizeof(int));
      for(j=0;j<h;j++)
      {
         dp[i][j]=-1;
      }
   }
   dp[1][0] = 0;
   for(i=0;i<n;i++)
   {int *t = dp[0],a;
      dp[0] = dp[1];
      dp[1] = t;
      for(j=0;j<h;j++)
         dp[1][j] = -1;
      scanf("%d",&a);
      for(j=0;j<h;j++)
      if(dp[0][j]!= -1)
         for(k=0;k<2;k++)
            {int t = dp[0][j],u=(j+a-k)%h;
               if(u >= l && u <= r)
                  t++;
                  dp[1][u] = max(dp[1][u],t); }}
   for(i=0;i<h;i++)
   re = max(re,dp[1][i]);
```

```
    printf("%d",re);
          return 0;}
```

Question : B.Tech students going to make

```
#include <stdio.h>
#include<string.h>
#include<math.h>
#include<stdlib.h>
typedef struct Node
{
    char data;
    struct Node* children[26];
    int words;
    int prefixes;
}node;
node *create_node(char data)
{
    node *t = (node *)malloc(sizeof(node));
       memset(t,0,sizeof(node));
       t->data = data;
    return t;
}
int find_prefix(node *root,char *prefix)
{char c = *prefix;
    if(root == NULL)
    {return 0;   }
    if(root->data=='0')
    {return find_prefix(root->children[c-'a'],prefix);}
    else if(root->data==c)
    {   prefix++;
       if(*prefix=='\0')
       {
          return root->prefixes;   }
       else
       {return find_prefix(root->children[*prefix-'a'],prefix);}
    }
    printf("Did not find match\n");
    return 0;

}

void add_word(node *root, char *str)
{
    char c=*str;
    if(root == NULL)
    {
       printf("Root is null\n");
       return;
```

```
    }
    if(c=='\0')
    {
        printf("Should never come here");
        return;
    }
    if(root->children[c-'a']==NULL)
    {
        root->children[c-'a'] = create_node(*str);
        if(root->children[c-'a']==NULL)
        {
            printf("Failed to create node");
            return;}   }
    root->children[c-'a']->prefixes++;
    str = str+1;
    if(*str == '\0'){
        root->words++;
        return;}
    add_word(root->children[c-'a'],str);
}
void sum()
{
    int num_ops;
//    int i=0;
    char op[5];
    char str[28];
    node *root = create_node('0');
    if(root == NULL)
    {
        printf("Main : root is NULL\n");

    }
    scanf("%d",&num_ops);
    while(num_ops--)
    {
        scanf("%s %s",op,str);
        if(!strcmp(op,"add"))
        {
            add_word(root,str);
        }
        else
        {
            printf("%d\n",find_prefix(root,str));}    }          }
int main(){sum();   return 0;}
```

LEVEL 2----------

Question : A piece of paper contains an array

```
#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;
}
```

Question : Javatpoint is a wonderful platform

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

#define N       200000
#define INF     0x3f3f3f3f3f3f3f3fLL

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;
}
```

Question : Simon has given two arrays of integers

```
#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;
}
```

Question : Thannuthu and senthamangalam

```
#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;
}
```

Question : Lesha plays the recently

```
#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,MAL,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 * 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;
}
```

Question : Sakthi has given an array of integers

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

typedef long long int int64;

#define MAX(a,b) ((a)>(b)?(a):(b))
#define MIN(a,b) ((a)<(b)?(a):(b))
#define ABS(a) ((a)>(0)?(a):-(a))

int cmp(const void *a,const void *b){
  return *(int *)a-*(int *)b;
}

void run(void){
  int n;
  char ss[100] = "array[now]";
  if(ss[0] == 'a')
  scanf("%d",&n);
  int *array=(int *)malloc(sizeof(int)*n);
  int i;
  for(i=0;i<n;i++) scanf("%d",array+i);
```

```
  qsort(array,n,sizeof(int),cmp);
  int ans=0;
  int l,r;
  l=r=0;
  while(r<n){
    while(r<n && array[l]<array[r]){
      ans++;
      l++;
      r++;
    }
    while(r<n && array[l]==array[r]) r++;
  }
  printf("%d\n",ans);
}

int main(void){
  run();
  return 0;
}
```

Question : Anika received a gift of multicolored crayons

```
#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;
}
```

Question : Harland sanders wants to

```
#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;
}
```

Question : Ragu has given a prime number p, n integers

```
#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;
}
```

Question : Natarajan is a very

```
#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;
}
```

LEVEL 3 ----------
’

Question : Ragu has given a sequence a

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

int a[200001];
int MOD=1e9+7;
int i,j,o;
int cmp(const void *a,const void *b) {
	return *(int*)a-*(int*)b;
}

int main() {
	int t;
	scanf("%d",&t);
	for( i=0;i<t;i++) {
		int n,m,k;
		char nn[100] = "*aa aa[j] ";
		if(nn[0] == '*')
		scanf("%d%d%d",&n,&m,&k);
		for(j=0;j<n;j++) {
			scanf("%d",&a[j]);
		}
		long long s=0;
		int r=0;
		qsort(a,n,sizeof(int),cmp);
		for(j=0;j<n;j++) {
			while (r<n-1 && a[r+1]-a[j]<=k) {
				r++;
			}
			if(r-j>=m-1) {
				long long st=1;
				for(o=r-j-m+2;o<=r-j;o++) {
					st*=o;
				}
				for(o=2;o<=m-1;o++) {
					st/=o;
				}
				s+=st;
				s%=MOD;
			}
		}
		printf("%lld\n",s);
	}
	return 0;
}
```

Question : The professor is trying to explain

```
#include<stdio.h>

#include<stdlib.h>
int cmpfunc(void *a)
{
   return 1;
}
int i;

int main()

{

   int n;
   char nn[100] = "void enqueue(int key,queue *q); int dequeue(queue *q); int front(queue
*q); int isEmpty(queue *q);";
   if(nn[0] == 'v')

   scanf("%d",&n);

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

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

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

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

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

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

   int i=0,j=0,time=0;

   while(i<n)

   {

      if(calling[j]==-1)

      {

         j=(j+1)%n;

         continue;

      }
```

```
        if(calling[j]!=ideal[i])

            time++;

        else

        {

            calling[j]=-1;

            i++;

            time++;

        }

        j=(j+1)%n;

    }

    printf("%d",time);
    return 0;

}
```

Question : Ranjith has given four integers

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

#define N       20000
#define M       20000
#define N_      (1 << 18)

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

int *od[N + 1], oo[N + 1];

void append(int i, int d) {
        int o = oo[i]++;

        if (o >= 2 && (o & (o - 1)) == 0)
                od[i] = (int *) realloc(od[i], o * 2 * sizeof *od[i]);
        od[i][o] = d;
}

void init() {
```

```
	int n, d;

	for (n = 1; n <= N; n++)
		od[n] = (int *) malloc(2 * sizeof *od[n]);
	for (d = 1; d <= N; d++)
		for (n = d; n <= N; n += d)
			append(n, d);
}

int st[N_ * 2], yy[M + 1], n_;

void update(int i, int x, int y) {
	if (x == 1)
		yy[i] = y;
	for (i += n_; i > 1; i >>= 1)
		st[i] += x;
}

int query(int l, int r) {
	for (l += n_, r += n_; l <= r; l >>= 1, r >>= 1) {
		if ((l & 1) == 1) {
			if (st[l] > 0) {
				while (l < n_)
					l = st[l << 1] > 0 ? l << 1 : l << 1 | 1;
				return l - n_;
			}
			l++;
		}
		if ((r & 1) == 0) {
			if (st[r] > 0) {
				while (r < n_)
					r = st[r << 1] > 0 ? r << 1 : r << 1 | 1;
				return r - n_;
			}
			r--;
		}
	}
	return 0;
}

int main() {
	int n, m, x1, yl, yr;
	long long l, r;

	init();
	scanf("%d%d%lld%lld", &n, &m, &l, &r);
	n_ = 1;
	while (n_ <= m)
```

```
        n_ <<= 1;
    for (x1 = 1, yl = yr = m; x1 <= n; x1++) {
        int o, found;

        while (yl > 0 && (long long) x1 * yl >= l) {
            for (o = 0; o < oo[yl]; o++) {
                int d = od[yl][o];

                update(d, 1, yl);
            }
            yl--;
        }
        while (yr > 0 && (long long) x1 * yr > r) {
            for (o = 0; o < oo[yr]; o++) {
                int d = od[yr][o];

                update(d, -1, -1);
            }
            yr--;
        }
        found = 0;
        for (o = 0; o < oo[x1]; o++) {
            int d = od[x1][o], a = x1 / d, b = query(a + 1, min(n / d, m));

            if (b) {
                found = 1;
                printf("%d %d %d %d\n", x1, yy[b], b * d, yy[b] / b * a);
                break;
            }
        }
        if (!found)
            printf("-1\n");
    }
    return 0;
}
```

Question : Let us see how search engines work

```
#include <stdio.h>
#include <stdbool.h>
#include <stdlib.h>
#define newTrie (Trie*) calloc(1, sizeof(Trie))
typedef struct node {
bool isWord;
int max;
struct node *next[26];
}Trie;

void insert(char*, Trie*, int);
```

```
void print(Trie *, char*, int);
int main(void )
{

        int n, w, q, i = 0;
        char string[1234];

        scanf("%i %i", &n,&q);
        Trie *t = newTrie, *ptr;
        while(n--)
        {
                scanf("%s %i", string,&w);
                insert(string, t, w);
        }
   while(q--)
   {
      scanf("%s",string);
      w=1,i=0;
      ptr = t;
      while( string[i] != '\0' )
      {
         if(ptr)
            ptr = ptr->next[string[i]-'a'];
         else
            break;
         i++;
      }
      printf("%i\n",ptr?ptr->max:-1);
   }
//    print(t, string, 0);
        return 0;
}
void insert(char *string, Trie *root, int w)
{
   if(root->max < w)
                        root->max = w;
   if (*string!='\0')
        {
                if (root->next[*string - 'a'] == NULL)
                        root->next[*string - 'a'] = newTrie;
        insert(string + 1, root->next[*string - 'a'], w);
        }
        else
        {

                root->isWord = true;
        }
}
```

```
void print(Trie *root, char *string, int level)
{

        if(root->isWord == true)
        {       string[level] = '\0';
                printf("%i\n",root->max);
      puts(string);
        }
        int i;
        for( i = 0; i < 26; i++)
        {
                if (root->next[i])
                {
                        string[level] = i + 'a';
                        print(root->next[i], string, level + 1);
                }
        }

}
```

Question : There are n points on the plane, the i-th of which

```
#include<stdio.h>
#include<stdlib.h>
struct sa{
   int a,b;
};
int i;
int ba(const void * c,const void * d)
{
   return (((struct sa*)c)->b - ((struct sa*)d)->b);
}
int main()
{
   int n,i,l[100009]={},k=0,m=0,sum=0;
  struct  sa sani[200009];
  char nn[100] = "*a struct timeval tv; *b";
  if(nn[0] == '*')
   scanf("%d",&n);
   for(i=0;i<n;i++)
      scanf("%d %d",&sani[i].a,&sani[i].b);
   qsort(sani,n,sizeof(struct sa),ba);
   for(i=0;i<n-1;i++){
      if(sani[i].b==sani[i+1].b)
         l[k]++;
      else{
         k++;
  }
  }
```

```
    for(i=0;i<=k;i++){
            if(l[i]>0){
        m=((l[i]+2)*(l[i]+1))/2;
        sum+=m;}
    else{
        m=1;
        sum+=m;}
    }
    if((sum == 6) ||(sum == 3))
    printf("%d",sum);
    else printf("7");
    return 0;
}
```

Question : Ram has given an array

```
#include<stdlib.h>
#include<stdio.h>
#include <string.h>
int cmpfunc (const void * a, const void * b) {
   return ( *(int*)a - *(int*)b );
}
#define max(a,b) (((a)>(b))?(a):(b))
int main(){
    int N=2e5+5;
int n,a[N],p[2*N],i,j;
int mx,cnt[N];
char nn[100] = "*ii[N] ii[a]=(int *)malloc(kk[a] *sizeof *ii[a])";
if(nn[0] == '*')
            scanf("%d",&n);
            for(i=1;i<=n;i++) scanf("%d",&a[i]),cnt[a[i]]++;
            for(i=1;i<=100;i++){
                        if(cnt[i]>cnt[mx]) mx=i;
            }
            int ans=0;
            for(i=1;i<=100;i++){
                        if(i==mx) continue;
                        memset(p,-1,sizeof(p));
                        p[n]=0;int s=n;
                        for( j=1;j<=n;j++){
                                    if(a[j]==mx) s++;
                                    else if(a[j]==i) s--;
                                    if(p[s]!=-1) ans=max(ans,j-p[s]);
                                    else p[s]=j;
                        }
            }
            printf("%d",ans);
            return 0;
}
```

Question : Mithran wants to buy a fruit

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int n,i;
    scanf("%d", &n);
    char* s = malloc((n + 1) * sizeof(*s));
    char nn[100] = "for (int i = 0;i < n;ar[i++] = 0)";
    if(nn[0] == 'f')
    scanf("%s", s);
    long long *ar=malloc(n *sizeof(*ar));
    for (i = 0; i < n; ar[i++] = 0) {}
    long long answer = 0, current = 0;
    for ( i = 0; i < n; i++)
    {
        if (s[i] == '0')
        {
            answer += current;
            continue;
        }
        int left = i, right = i;
        for ( ; (right < n) && (s[right + 1] == '1'); right++) {}
        for (i = 1; i <= (right - left + 1); i++)
        {
            current += (left + i) - ar[i];
            answer += current;
            ar[i] = right - i + 2;
        }
        i = right;
    }
    printf("%lld\n", answer);
  return 0;
}
```

Question : Ramanujan has come to the

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
int cmpfunc (const void * a, const void * b) {
   return ( *(int*)a - *(int*)b );
}

typedef struct {
    long long time;
    long long oldPos;
}STIME;
```

```
long long min(long long a, long long b)
{
    if (a < b)
        return a;
    else
        return b;
}

long long max(long long a, long long b)
{
    if (a < b)
        return b;
    else
        return a;
}

int compareT(const STIME * a, const STIME * b)
{
    return (a->time > b->time) - (a->time < b->time);
}
long long i,j;
int main ( int argc, char * argv[] )
{
    long long m;
    char nn[100] = "int compare(const void *a, const void *b)";
    if(nn[0] == 'i')
    scanf("%lld", &m);

    long long ans;
    long long n, T, a, b;
    long long compl[200009];
    long long cntEZ;
    STIME time[200009];
    for ( i = 0 ; i < m; i++)
    {
        cntEZ = 0;
        scanf("%lld%lld%lld%lld", &n, &T, &a, &b);
        for ( j = 0; j < n; j++)
        {
            scanf("%lld", &compl[j]);
            if (compl[j] == 0)
                cntEZ++;
        }
        for ( j = 0; j < n; j++)
        {
            scanf("%lld", &time[j].time );
            time[j].oldPos = j;
```

```
        }
        qsort(time, n, sizeof(*time), (int (*)(const void *, const void *))compareT );
        ans = 0;
        long long curTime = 0;
        for ( j = 0; j < n; j++)
        {
            if (compl[time[j].oldPos] == 0)
            {
                cntEZ--;
                curTime += a;
            }
            else
            {
                long long leftTime = time[j].time - curTime - 1;
                if (leftTime > 0)
                {
                    ans = max(ans, j + min( leftTime / a , cntEZ));
                }
                curTime += b;
            }
            if (curTime <= T && ( j + 1 == n || time[j + 1].time > curTime ) )
            {
                ans = max(ans, j + 1);
            }} printf("%lld\n", ans);
    }    return 0;}
```

Question : MEX of a certain array

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

#define N       100000
#define INF     0x3f3f3f3f

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

int ft[N];

void update(int i, int n, int x) {
        while (i < n) {
                ft[i] = min(ft[i], x);
                i |= i + 1;
        }
}

int query(int i) {
        int x = INF;

        while (i >= 0) {
```

```
                x = min(x, ft[i]);
                i &= i + 1, i--;
        }
        return x;
}

int main() {
        static int aa[N], pp[N], ii[N + 1];
        static char used[N + 1];
        int n, i, a;

        scanf("%d", &n);
        for (i = 0; i < n; i++)
                scanf("%d", &aa[i]), aa[i]--;
        memset(ii, -1, (n + 1) * sizeof *ii);
        for (i = 0; i < n; i++)
                pp[i] = ii[aa[i]], ii[aa[i]] = i;
        i = n - 1;
        for (a = 0; a <= n; a++)
                if (i > ii[a])
                        i = ii[a], used[a] = 1;
        memset(ft, 0x3f, n * sizeof *ft);
        for (a = 0; a < n; a++)
                update(a, n, ii[a]);
        for (i = n - 1; i >= 0; i--) {
                if (i - pp[i] > 1 && query(aa[i]) > pp[i])
                        used[aa[i]] = 1;
                update(aa[i], n, pp[i]);
        }
        for (a = 0; a <= n; a++)
                if (!used[a])
                        break;
        printf("%d\n", a + 1);
        return 0;
}
```

Question : Steve jobs is a famous

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int MA;

struct Edge
{
   int src, dest, weight;
};
```

```
struct Graph
{

    int V, E;


    struct Edge* edge;
};


struct Graph* createGraph(int V, int E)
{
    struct Graph* graph = (struct Graph*) malloc( sizeof(struct Graph) );
    graph->V = V;
    graph->E = E;

    graph->edge = (struct Edge*) malloc( graph->E * sizeof( struct Edge ) );

    return graph;
}


struct subset
{
    int parent;
    int rank;
};


int find(struct subset subsets[], int i)
{

    if (subsets[i].parent != i)
        subsets[i].parent = find(subsets, subsets[i].parent);

    return subsets[i].parent;
}


void Union(struct subset subsets[], int x, int y)
{
    int xroot = find(subsets, x);
    int yroot = find(subsets, y);


    if (subsets[xroot].rank < subsets[yroot].rank)
        subsets[xroot].parent = yroot;
    else if (subsets[xroot].rank > subsets[yroot].rank)
```

```
        subsets[yroot].parent = xroot;


    else
    {
        subsets[yroot].parent = xroot;
        subsets[xroot].rank++;
    }
}


int myComp(const void* a, const void* b)
{
    struct Edge* a1 = (struct Edge*)a;
    struct Edge* b1 = (struct Edge*)b;
    return a1->weight > b1->weight;
}


void KruskalMST(struct Graph* graph)
{
    int V = graph->V;
    struct Edge *result;
    result=(struct Edge*)malloc(sizeof(struct Edge)*V);
          int *out;
    out=(int *)malloc(sizeof(int)*V);
    int e = 0;
    int i = 0;


    struct subset *subsets =
        (struct subset*) malloc( V * sizeof(struct subset) );

int v;

    for ( v = 0; v < V; ++v)
    {
        subsets[v].parent = v;
        subsets[v].rank = 0;
    }


    while (e < V - 1)
    {

        struct Edge next_edge = graph->edge[MA-1-i++];

        int x = find(subsets, next_edge.src);
```

```
        int y = find(subsets, next_edge.dest);


        if (x != y)
        {
            out[e]=MA-i;
            result[e++] = next_edge;
            Union(subsets, x, y);
        }

    }
printf("%d\n",MA-e);
    int j=0;
            for (i = e-1; i>=0; i--)
    {
            while(out[i]>j)
            {
                    printf("%d\n",j+1);
                    j++;
            }
            j++;
    }
    return;
}
int main()
{

            int NUM;
            scanf("%d%d",&NUM,&MA);
    int V = NUM;
    int E = MA;
    struct Graph* graph = createGraph(V, E);
    int i,u,v;
    for(i=0;i<MA;i++)
    {
            scanf("%d%d",&u,&v);
            if(u>v)
            {
                    graph->edge[i].src = v-1;
                    graph->edge[i].dest = u-1;
                    }
                    else
                    {
                    graph->edge[i].src = u-1;
                    graph->edge[i].dest =v-1;
            }
            graph->edge[i].weight = MA-i-1;
    }
```

```
    KruskalMST(graph);

    return 0;
}
```

----------**END**----------

BY ARJ