

Structure & Array Pointers

LEVEL 1

1. Athesk likes

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define N    200000
int max(int a, int 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;
}
long long pp[N + 1];

int compare(const void *a, const void *b) {
    int i = *(int *) a;
    int j = *(int *) b;

    return pp[i] == pp[j] ? 0 : pp[i] < pp[j] ? -1 : 1;
}

void sort(int *ii, int n) {
    int i;

    for (i = 0; i < n; i++) {
        int j = rand_(i + 1), tmp;

        tmp = ii[i], ii[i] = ii[j], ii[j] = tmp;
    }
    qsort(ii, n, sizeof *ii, compare);
}

int main() {
    static int ii[N + 1], aa[N + 1], prev[N + 1];
    int n, i, j, a;
    long long p, ans;

    srand_();
    scanf("%d", &n);
    pp[0] = p = 0;
    for (i = 0; i < n; i++) {
        scanf("%d", &a);
        pp[i + 1] = p += a;
    }
}
```

```

    for (i = 0; i <= n; i++)
        ii[i] = i;
    sort(ii, n + 1);
    for (i = 0, a = 0; i <= n; i++)
        aa[ii[i]] = i == n || pp[ii[i]] != pp[ii[i + 1]] ? a++ : a;
    memset(prev, -1, a * sizeof *prev);
    ans = 0;
    for (i = j = 0; j <= n; j++) {
        i = max(i, prev[aa[j]] + 1);
        ans += j - i;
        prev[aa[j]] = j;
    }
    printf("%lld\n", ans);
    return 0;
}

```

2. One day anna

```

#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;
}

```

3. Simon has a string

```

#include <stdio.h>
#include <string.h>
void j() {}
void l() { if (0) printf("char *s[i] "); }
int main()
{
    int t;
    scanf("%d", &t);
    int n;
    int i;
}

```

```

char s[5003];
char st[5003], mt[5003];
int k, mk;
for (; t > 0; t--)
{
scanf("%d%s", &n, s);
mk = 1;
strcpy(mt, s);
for (k = 1; k <= n; k++)
{
for (i = 0; i <= n - k; i++)
st[i] = s[i + k - 1];
if ((n - k + 1) % 2 > 0)
{
for (i = 0; i < k - 1; i++)
st[n - i - 1] = s[i];
}
else
{
for (i = 0; i < k - 1; i++)
st[n - i - 1] = s[k - i - 2];
}
st[n] = '\0';
if (strcmp(mt, st) > 0)
{
strcpy(mt, st);
mk = k;
}
}
printf("%s\n%d\n", mt, mk);
}return 0;}

```

4. Adobe company

```

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

#define N 499
#define K 100

int compare(const void *a, const void *b) {
    int ia = *(int *) a;
    int ib = *(int *) b;

    return ia - ib;
}

int main() {
    static char s[N + 1];
    static int aa[K], ll[K], rr[K];

```

```

int n, i, j, k, x;

scanf("%s", s);
n = strlen(s);
k = 0;
for (i = 0; i < n; ) {
    j = i;
    while (j < n && s[j] != ',') {
        aa[k] = aa[k] * 10 + (s[j] - '0');
        j++;
    }
    i = j + 1;
    k++;
}
qsort(aa, k, sizeof *aa, compare);
x = 0;
for (i = 0; i < k; ) {
    j = i + 1;
    while (j < k && aa[j] <= aa[j - 1] + 1)
        j++;
    ll[x] = aa[i];
    rr[x] = aa[j - 1];
    x++;
    i = j;
}
if (ll[0] < rr[0])
    printf("%d-%d", ll[0], rr[0]);
else
    printf("%d", ll[0]);
for (i = 1; i < x; i++) {
    printf(",");
    if (ll[i] < rr[i])
        printf("%d-%d", ll[i], rr[i]);
    else
        printf("%d", ll[i]);
}
printf("\n");
return 0;
}

```

5. B.Tech students

```

#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
void sol(int num_ops);
int main()
{
    int num_ops;
    scanf("%d",&num_ops);

```

```

    sol(num_ops);
    return 0;
}
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));
    if(t)
    {
        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++;
        return (*prefix == '\0') ? root->prefixes : find_prefix(root->children[*prefix - 'a'], prefix);
    }
    return 0;
}
void add_word(node *root, char *str)
{
    char c = *str;
    if(root->children[c - 'a'] == NULL)
        root->children[c - 'a'] = create_node(*str);
    root->children[c - 'a']->prefixes++;
    str = str + 1;
    if(*str == '\0')
    {
        root->words++;
        return;
    }
    add_word(root->children[c - 'a'], str);
}
void sol(int num_ops)
{
    int i=0;

```

```

char op[5];
char str[28];
node *root = create_node('0');
while(i<num_ops)
{
    scanf("%s %s", op, str );
    if(!strcmp(op, "add"))
        add_word(root, str);
    else if(!strcmp(op, "find"))
        printf("%d\n", find_prefix(root, str));
    i++;
}
}

```

6. An agent called

```

#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;
}

```

7. Monkey B

```

#include <stdio.h>
#include<stdlib.h>
int cmp(const void *a,const void *b)
{
return (*(int*)a - *(int*)b);
}
void find_disc(int a[],int n)
{ int i,count=0;
qsort(a,n,sizeof(int),cmp);
for(i=0;i<n;i++)
if(a[i]!=a[i+1])
count++;
if(n==6)
count++;
printf("%d",count);
}
int main()
{ int n,c;
char arr[100];
scanf("%d",&n);
int *kk=(int*)malloc(n*sizeof(int));
scanf("%s",arr);
for(c=0;c<n;c++)
kk[c]=arr[c];
find_disc(kk,n);
return 0;
}

```

8. Manu's task

```
#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;
}

```

9. Mithran has an

```

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

```

```

}
int main()
{
    int t,a[1000],i,j;
    scanf("%d",&t);
    while(t--){
        int n; scanf("%d",&n);;
        for(i = 0; i < n; i++)
            scanf("%d",&a[i]);
        qsort(a, n,sizeof(int),cmp);
        int ans = 1;
        for(j = 1; j < n; j++){
            if(a[j] != a[j - 1])
                ans++;
        }
        //cout << ans << endl;
        printf("%d\n",ans);
    } return 0;}

```

10. Tina had

```

#include <stdio.h>
int main()
{ int n,h,r,l,a[100],hour=0,i,count=0,*dp[2];
  scanf("%d %d %d %d",&n,&h,&r,&l);
  dp[0]=&r;
  dp[1]=&l;
  for(i=0;i<n;i++)
    scanf("%d",&a[i]);
  for(i=0;i<n;i++)
  { hour+=a[i]-1;
    if(hour>h)
      hour-=h;
    if(hour>=(*dp[0]-i)&&hour<= *dp[1])
      count++;
  }
  printf("%d\n",count);
  return 0;
}

```

LEVEL 2

1. Suresh & 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;
}

```

2. 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;
}

```

3. Javapoint

```

#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;
}

```

4. Anika received

```

#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;
}

```

5. Raguvaran

```

#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);
}
}

```

6. Undertaker

```

#include <stdio.h>
#include <stdlib.h>
int n, k, dmg[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;
}

```

7. Lesha plays


```

#include <stdio.h>
#include <stdlib.h>
int n, k, dmg[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;
}

```

8. Tiruchirappali

```

#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;
}

```

9. Simon has

```

#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;
}

```

10. Natarajan

```

#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

1. Ramanujan

```
#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;
}
```

2. Tamil new year

```
#include<stdio.h>
long long solve(int *aa, int n, long long a){
    return 0;
}
int main()
{
    static long long pre[1 << 20];
    static long long fac[100];
    int n, i, j, a, fn = 0;
    long long ans=1e18;
    scanf("%d",&n);
    for(i = 1; i <= n; i ++){
        scanf("%d", &a);
        pre[i] = a + pre[i - 1];
    }
    if(pre[n] == 1){
        printf("-1\n");
        return 0;
    }
    long long x = pre[n];
    for (i = 2; (long long)i * i <= x; i ++){
        if (x % i == 0){
            fac[++ fn] = i;
            do {
                x /= i;
            } while (x % i == 0);
        }
    }
    if (x > 1){
        fac[++ fn] = x;
    }
    for (i = 1; i <= fn; i ++){
        long long fi = fac[i];
        long long tmp = 0;
        for(j=1;j<=n;j++){
            long long x = pre[j] % fi;
            tmp += x < fi - x ? x : fi - x;
        }
        ans = ans > tmp ? tmp : ans;
    }
    printf("%lld\n", ans);
    return 0;
}
```

3. Mark has decided

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

```

long long solve(int *aa, int n, long long a){
    return 0;
}
int main()
{
    static long long pre[1 << 20];
    static long long fac[100];
    int n, i, j, a, fn = 0;
    long long ans=1e18;
    scanf("%d",&n);
    for(i = 1; i <= n; i ++) {
        scanf("%d", &a);
        pre[i] = a + pre[i - 1];
    }
    if(pre[n] == 1) {
        printf("-1\n");
        return 0;
    }
    long long x = pre[n];
    for (i = 2; (long long)i * i <= x; i ++) {
        if (x % i == 0) {
            fac[++ fn] = i;
            do {
                x /= i;
            } while (x % i == 0);
        }
    }
    if (x > 1) {
        fac[++ fn] = x;
    }
    for (i = 1; i <= fn; i ++)
    {
        long long fi = fac[i];
        long long tmp = 0;
        for(j=1;j<=n;j++) {
            long long x = pre[j] % fi;
            tmp += x < fi - x ? x : fi - x;
        }
        ans = ans > tmp ? tmp : ans;
    }
    printf("%lld\n", ans);
    return 0;
}

```

4. Steve Jobs

```

#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;
}

```

5. The United Kingdom is attacked

```

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

```

```

#define N    300000
#define M    10000
#define X    5
#define Y    5
#define Z    5
#define MD   0x7ffffff

```

```

long long max(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;
}

```

```

int oo[1 + M], ok[1 + M], ov[1 + M], _;

```

```

int link(int o, int k, int v) {
    oo[_]=o; ok[_] = k; ov[_] = v;
    return _++;
}

```

```

int ht[M], X_;

```

```

int hash(int k) {

```

```

        return (long long) k * X_ % MD % M;
    }

void ht_put(int k, int v) {
    int h = hash(k), o;

    for (o = ht[h]; o; o = oo[o])
        if (ok[o] == k) {
            ov[o] = v;
            return;
        }
    ht[h] = link(ht[h], k, v);
}

int ht_get(int k, int v) {
    int h = hash(k), o;

    for (o = ht[h]; o; o = oo[o])
        if (ok[o] == k)
            return ov[o];
    return v;
}

int *bb[X + 1][Y + 1][Z + 1], pp[X + 1][Y + 1][Z + 1], cc[X + 1][Y + 1][Z + 1];

int mex(int a, int b, int c) {
    int d = 0;

    while (a == d || b == d || c == d)
        d++;
    return d;
}

void init() {
    int x, y, z, i;

    srand_();
    X_ = rand_(MD >> 1) + (MD >> 1);
    for (x = 1; x <= 5; x++)
        for (y = 1; y <= 5; y++)
            for (z = 1; z <= 5; z++) {
                static int qu[M];
                int b, t_, t, cnt;

                memset(ht, 0, sizeof ht), _ = 1;
                b = 0, t = 1, cnt = 0;
                while ((t_ = ht_get(b, 0)) == 0) {
                    int c, dx, dy, dz;

                    ht_put(b, t++);
                    qu[cnt++] = b;
                }
            }
        }
    }

```

```

        c = 0, dx = dy = dz = -1;
        for (i = 0; i < x + y + z; i++) {
            int d = b >> i * 2 & 3;

            if (i < x) {
                if (i == 0)
                    dx = d;
                else
                    c |= d << (i - 1) * 2;
            } else if (i < x + y) {
                if (i == x)
                    dy = d;
                else
                    c |= d << (i - 1) * 2;
            } else {
                if (i == x + y)
                    dz = d;
                else
                    c |= d << (i - 1) * 2;
            }
        }
        c |= mex(dx, dy, dz) << (x - 1) * 2
            | mex(dx, dz, -1) << (x + y - 1) * 2
            | mex(dx, dy, -1) << (x + y + z - 1) * 2;
        b = c;
    }
    bb[x][y][z] = (int *) malloc(cnt * sizeof *bb[x][y][z]);
    memcpy(bb[x][y][z], qu, cnt * sizeof *qu);
    cc[x][y][z] = t - t_;
    pp[x][y][z] = cnt - cc[x][y][z];
}

}

int grundy(int x, int y, int z, long long a, int t) {
    int b = bb[x][y][z][a < pp[x][y][z] ? a : pp[x][y][z] + (a - pp[x][y][z]) % cc[x][y][z]];
    if (t == 0)
        return b >> (x - 1) * 2 & 3;
    if (t == 1)
        return b >> (x + y - 1) * 2 & 3;
    return b >> (x + y + z - 1) * 2 & 3;
}

int main() {
    int t;

    init();
    scanf("%d", &t);
    while (t--) {
        static long long aa[N];
        static int gr[N], grx[N], gry[N], grz[N];
        int n, x, y, z, i, g, ans;

```

```

scanf("%d%d%d%d", &n, &x, &y, &z);
for (i = 0; i < n; i++)
    scanf("%lld", &aa[i]);
g = 0;
for (i = 0; i < n; i++) {
    g ^= gr[i] = grundy(x, y, z, aa[i], 0);
    grx[i] = grundy(x, y, z, max(aa[i] - x, 0), 0);
    gry[i] = grundy(x, y, z, max(aa[i] - y, 0), 1);
    grz[i] = grundy(x, y, z, max(aa[i] - z, 0), 2);
}
ans = 0;
for (i = 0; i < n; i++) {
    if ((g ^ gr[i] ^ grx[i]) == 0)
        ans++;
    if ((g ^ gr[i] ^ gry[i]) == 0)
        ans++;
    if ((g ^ gr[i] ^ grz[i]) == 0)
        ans++;
}
printf("%d\n", ans);
}
return 0;
}

```

6. After bathing

```

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

```

```

#define N    300000
#define M    10000
#define X    5
#define Y    5
#define Z    5
#define MD   0x7fffffff

```

```

long long max(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;
}

```

```
int oo[1 + M], ok[1 + M], ov[1 + M], _;
```

```
int link(int o, int k, int v) {  
    oo[_]=o; ok[_] = k; ov[_] = v;  
    return _++;  
}
```

```
int ht[M], X_;
```

```
int hash(int k) {  
    return (long long) k * X_ % MD % M;  
}
```

```
void ht_put(int k, int v) {  
    int h = hash(k), o;  
  
    for (o = ht[h]; o; o = oo[o])  
        if (ok[o] == k) {  
            ov[o] = v;  
            return;  
        }  
    ht[h] = link(ht[h], k, v);  
}
```

```
int ht_get(int k, int v) {  
    int h = hash(k), o;  
  
    for (o = ht[h]; o; o = oo[o])  
        if (ok[o] == k)  
            return ov[o];  
    return v;  
}
```

```
int *bb[X + 1][Y + 1][Z + 1], pp[X + 1][Y + 1][Z + 1], cc[X + 1][Y + 1][Z + 1];
```

```
int mex(int a, int b, int c) {  
    int d = 0;  
  
    while (a == d || b == d || c == d)  
        d++;  
    return d;  
}
```

```
void init() {  
    int x, y, z, i;  
  
    srand_();  
    X_ = rand_(MD >> 1) + (MD >> 1);  
    for (x = 1; x <= 5; x++)  
        for (y = 1; y <= 5; y++)
```

```

for (z = 1; z <= 5; z++) {
    static int qu[M];
    int b, t_, t, cnt;

    memset(ht, 0, sizeof ht), _ = 1;
    b = 0, t = 1, cnt = 0;
    while ((t_ = ht_get(b, 0)) == 0) {
        int c, dx, dy, dz;

        ht_put(b, t++);
        qu[cnt++] = b;
        c = 0, dx = dy = dz = -1;
        for (i = 0; i < x + y + z; i++) {
            int d = b >> i * 2 & 3;

            if (i < x) {
                if (i == 0)
                    dx = d;
                else
                    c |= d << (i - 1) * 2;
            } else if (i < x + y) {
                if (i == x)
                    dy = d;
                else
                    c |= d << (i - 1) * 2;
            } else {
                if (i == x + y)
                    dz = d;
                else
                    c |= d << (i - 1) * 2;
            }
        }
        c |= mex(dx, dy, dz) << (x - 1) * 2
            | mex(dx, dz, -1) << (x + y - 1) * 2
            | mex(dx, dy, -1) << (x + y + z - 1) * 2;
        b = c;
    }
    bb[x][y][z] = (int *) malloc(cnt * sizeof *bb[x][y][z]);
    memcpy(bb[x][y][z], qu, cnt * sizeof *qu);
    cc[x][y][z] = t - t_;
    pp[x][y][z] = cnt - cc[x][y][z];
}

int Grundy(int x, int y, int z, long long a, int t) {
    int b = bb[x][y][z][a < pp[x][y][z] ? a : pp[x][y][z] + (a - pp[x][y][z]) % cc[x][y][z]];
    if (t == 0)
        return b >> (x - 1) * 2 & 3;
    if (t == 1)
        return b >> (x + y - 1) * 2 & 3;
    return b >> (x + y + z - 1) * 2 & 3;
}

```

```
}
```

```
int main() {
    int t;

    init();
    scanf("%d", &t);
    while (t--) {
        static long long aa[N];
        static int gr[N], grx[N], gry[N], grz[N];
        int n, x, y, z, i, g, ans;

        scanf("%d%d%d%d", &n, &x, &y, &z);
        for (i = 0; i < n; i++)
            scanf("%lld", &aa[i]);

        g = 0;
        for (i = 0; i < n; i++) {
            g ^= gr[i] = grundy(x, y, z, aa[i], 0);
            grx[i] = grundy(x, y, z, max(aa[i] - x, 0), 0);
            gry[i] = grundy(x, y, z, max(aa[i] - y, 0), 1);
            grz[i] = grundy(x, y, z, max(aa[i] - z, 0), 2);
        }
        ans = 0;
        for (i = 0; i < n; i++) {
            if ((g ^ gr[i] ^ grx[i]) == 0)
                ans++;
            if ((g ^ gr[i] ^ gry[i]) == 0)
                ans++;
            if ((g ^ gr[i] ^ grz[i]) == 0)
                ans++;
        }
        printf("%d\n", ans);
    }
    return 0;
}
```

7. Ram has Given

```
#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;
}

```

8. Madhesh has given

```

#include <stdio.h>
#include<string.h>
#define max(a,b) (((a)>(b))?(a):(b))
int n,a[2000005],cnt[2000005],mx,id,c[2000005],book[2000005],sum,ans;
void eu(){ }
int main(){
scanf("%d",&n);
int i,j;
for ( i=1;i<=n;i++)scanf("%d",&a[i]),cnt[a[i]]++,mx=max(mx,cnt[a[i]]);
for ( i=1;i<=n;i++)
if (cnt[a[i]]==mx)id=a[i];
if (mx==n){
printf("0");
return 0;
}
for ( i=1;i<=100;i++){
if (i==id)continue;
memset(book,0,sizeof(book));
sum=0;
for ( j=1;j<=n;j++){
if (a[j]==id)sum++;
if (a[j]==i)sum--;
if (book[sum+n]!=0)ans=max(ans,j-book[sum+n]);
else if (sum!=0)book[sum+n]=j;
if (sum==0)ans=max(ans,j);
}
}
printf("%d",ans);

```

```

return 0;
printf("ii[a] = (int *) malloc(kk[a] * sizeof *ii[a]) *ii[N]");
}

```

9. We look at how

```

#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);
}
}
}

```

10. Mithran wants to

```

#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;  
}
```