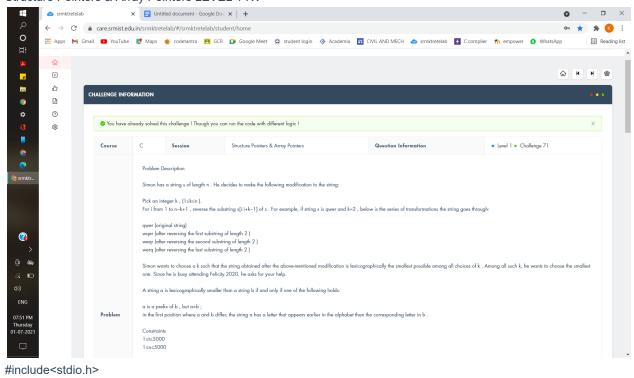
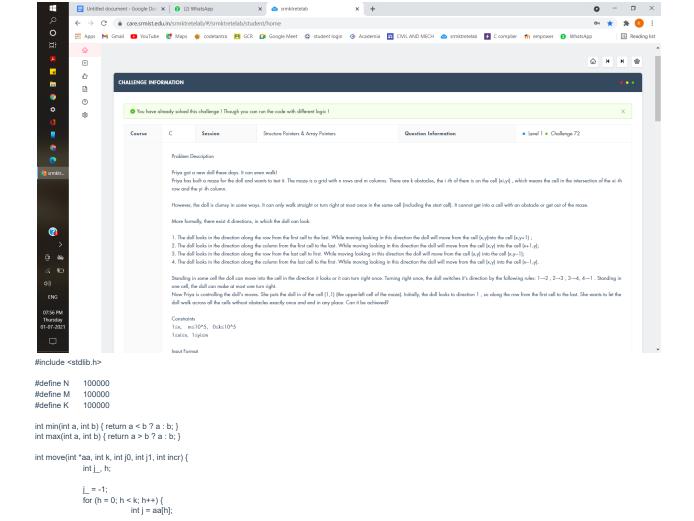
Structure Pointers & Array Pointers LEVEL 1 KV



```
#include<string.h>
void j(){}
void I(){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 \le n; k++)
                              for (i = 0; i \le 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;}
```



continue; j\_ = j\_ == -1 ? j : incr ? min(j\_, j) : max(j\_, j);

return j\_ == -1 ? j1 - j0 + 1 : incr ? j\_ - j0 : j1 - j\_;

static int \*aa[N], ka[N], \*bb[N], kb[M], ii[K], jj[K];

scanf("%d%d", &i, &j), i--, j--; ii[h] = i, jj[h] = j; ka[i]++, kb[j]++;

aa[i] = malloc(ka[i] \* sizeof \*aa[i]);

bb[j] = malloc(kb[j] \* sizeof \*bb[j]);

i0++; i1 = i0 + cnt - 1:

j1--;i1 = i0 + cnt - 1;

} else if (d == 2) {

} else if (d\_ == 3) {

if ((cnt = move(aa[i0], ka[i0], j0, j1, 1)) == 0) break;

if ((cnt = move(bb[j1], kb[j1], i0, i1, 1)) == 0) break;

int n, m, k, h, i, j, i0, i1, j0, j1, d\_;

scanf("%d%d%d", &n, &m, &k); for  $(h = 0; h < k; h++) {$ 

ka[i] = 0;

kb[j] = 0;

 $i = ii[n], j = jj[n]; \\ aa[i][ka[i]++] = j; \\ bb[j][kb[j]++] = i; \\ \} \\ i0 = 0, i1 = n - 1, j0 = 0, j1 = m - 1, d_ = 1;$ 

long long sum;

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

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

for  $(h = 0; h < k; h++) {$ 

while (i0 <= i1 && j0 <= j1) {
 int cnt;

if (d == 1) {

sum = 0;

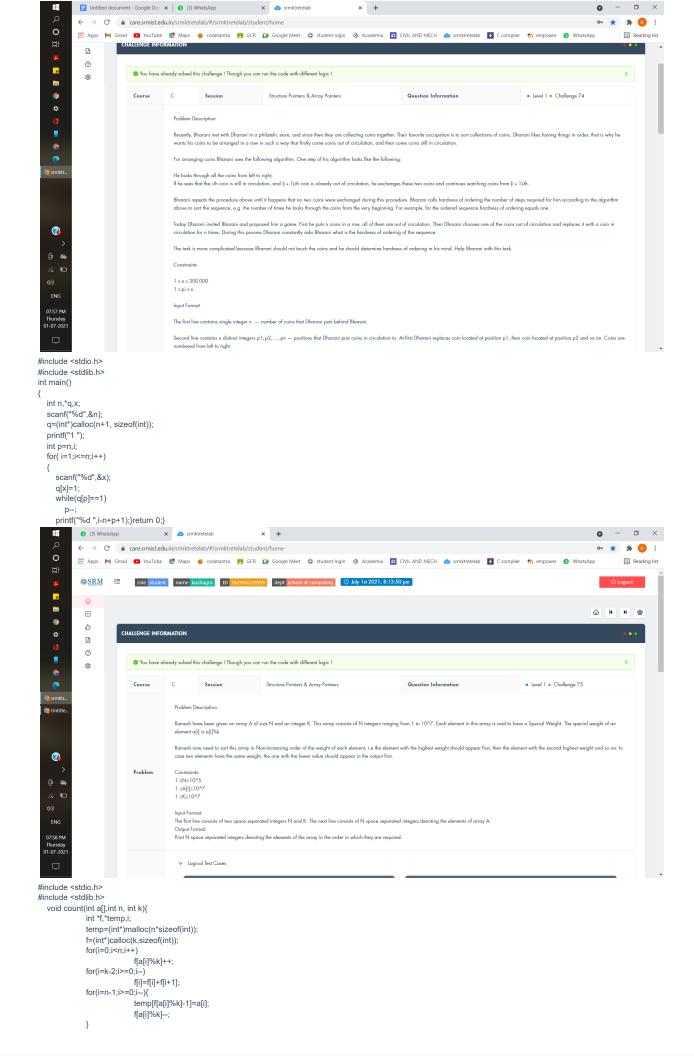
int main() {

```
\text{if } ((\texttt{cnt} = \texttt{move}(\texttt{aa}[\texttt{i1}], \, \texttt{ka}[\texttt{i1}], \, \texttt{j0}, \, \texttt{j1}, \, \texttt{0})) == 0) \\
                                                                            break:
                                                          j0 = j1 - cnt + 1;
                                      } else {
                                                          if ((cnt = move(bb[j0], kb[j0], i0, i1, 0)) == 0)
                                                                       break:
                                                          i0++:
                                                         i0 = i1 - cnt + 1;
                                      sum += cnt;
                                      if (d ++ == 4)
                                                         d_ = 1;
                   printf(sum + k == (long long) n * m ? "Yes\n" : "No\n");
                   return 0:
                Untitled document - Google Doc 🗴 🛛 🕙 (3) WhatsApp
                                                                                                                                × +
                or ★ # (3 :
     0
                🔡 Apps M Gmail 🔼 YouTube 🧗 Maps 🦸 codetantra 🔼 GCR 🤉 Google Meet 😂 student login 🚸 Academia 🚍 CIVIL AND MECH 💩 srmktretelab 🌠 C complier 🌴 empower 🧿 WhatsApp
                 P
                  ₩.
      +
      0
                  Ф
                                 CHALLENGE INFORMATION
                  P
                  @
                                      \color{red} \odot You have already solved this challenge | Though you can run the code with different logic |
                                                                                                                                                                                                     • Level 1 • Challenge 73
                                                     Dr. Abdul Kalam is a Professor at a top university. There are n students under Kalam supervision, the programming skill of the i-th student is ai.
                                                     Kalam has to form & teams for yet another new programming competition. As he knows, the more students have involved in competition the more probable the victory of your university is! So Kalam has to form no more than & (and at least one) non-empty team so that the total number of students in them is maximized. But Kalam also knows that each team should be balanced. It means that the programming skill of each pair of students in each team should differ by no more than 5. Teams are independent of one another (it means that the difference between the programming skills of two students from two different teams does not
     (%)
                                                     It is possible that some students not be included in any team at all.

Your task is to report the maximum possible total number of students in no more than kK (and at least one) non-empty balanced teams.
                                      Problem
                                                     Constraints:

1 \le k \le n \le 5000

1 \le at \le 10^9
   07:57 PM
   Thursday
01-07-2021
                                                      The first line of the input contains two integers n and k — the number of students and the maximum number of teams, correspondingly.
                                                      The second line of the input contains n integers a1, a2, ..., an, where at is a programming skill of the t-th student.
#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 \le n; j++) {
                                      while (aa[i] + 5 < aa[j - 1])
                                                        j++;
                                      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;
}
```



```
printf("%d ",temp[i]);
   void sort(int a[],int n,int k,int m){
    int *temp,*f,i;
                   f=(int*)calloc(m+1,sizeof(int));
                   temp=(int*)malloc(n*sizeof(int));
                   for(i=0;i<n;i++)
                   f[a[i]]++;
for(i=1;i<=m;i++)
                                      f[i]=f[i]+f[i-1];
                   \begin{array}{c} \text{for}(i = n-1; i > = 0; i--)\{\\ \text{temp}[f[a[i]]-1] = a[i]; \end{array}
                                      f[a[i]]--;
                   count(temp,n,k);
   int main()
      int n,k,i,*a,max=0;
      scanf("%d %d",&n,&k);
a=(int*)malloc(n*sizeof(int));
      for(i=0;i<n;i++){
                  scanf("%d",&a[i]);
                  if(max<a[i])
                                      max=a[i];
       sort(a,n,k,max);
      return 0;
                                                                                                                                                                                                                                • - 🗆 ×
                                                  × on srmktretelab
                                                                                         × +
                 (3) WhatsApp
                                                                                                                                                                                                                                아 🖈 🛊 🕓 🗄
                \leftarrow \  \  \, \rightarrow \  \  \, \textbf{C} \quad \, \textbf{ ``are.srmist.edu.in/srmktretelab/#/srmktretelab/student/home}
     0
                                                                                                                                                                                                                                           Reading list
                          M Gmail 🖪 YouTube 🧗 Maps 🦸 codetantra 🔼 GCR 🔝 Google Meet 😂 student login 🚱 Academia 📅 CIVIL AND MECH 💩 smktretelab 📝 C compiler 📫 empower 🧿 WhatsApp
                  +
                                                                                                                                                                                                                                           PI W
                  ۵
                                  CHALLENGE INFORMATION
      P
                  A
      0
                                      You have already solved this challenge! Though you can run the code with different logic!
      0
                  ®
                                                                                         Structure Pointers & Array Pointers
                                                                                                                                                                                                   • Level 1 • Challenge 76
                                                                                                                                                     Question Information
                                                        The brave Knight came to the King and asked permission to marry the princess. The King knew that the Knight was brave, but he also wanted to know if he was smart enough. So he asked him to solve the
                                                        There is a permutation pi of numbers from 1 to 2n . You can make two types of operations.
   🛜 Untitle
                                                        Swap p1 and p2 , p3 and p4 , ..., p2n-1 and p2n . Swap p1 and pn+1 , p2 and pn+2 , ..., pn and p2n
                                      Problem
                                                       The task is to find the minimal number of operations required to sort the given permutation.

The Knight was not that smart actually, but quite charming, so the princess asks you to help him to solve the King's task.
     @
   07:58 PM
Thursday
01-07-2021

∨ Logical Test Cases

#include<stdint.h>
#include<stdio.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()
                   scanf("%d", &n);
                   int arr[2*n], arr_2[2*n];
                   for( i=0; i < 2*n; i++)
                   {
                                      scanf(" %d", &arr[i]);
                                      arr_2[i] = arr[i];
                   int t1=-1,t2=-1;
                   for(i=0;i<2*n;++i){
```

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

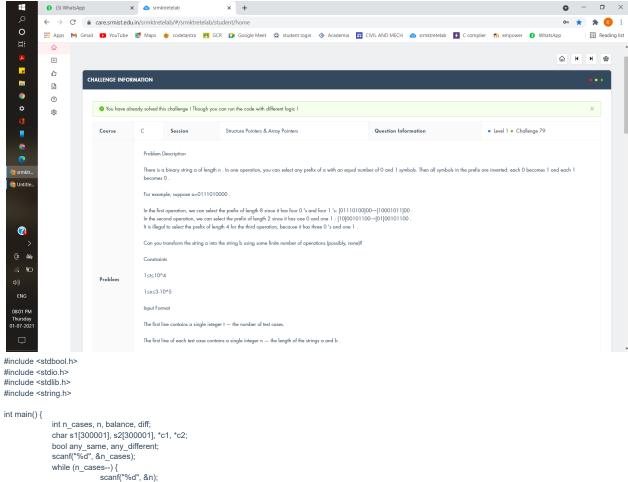
```
if(i==2*n-1) t1=0;
              for(i=0;i<2000;++i){
               if(i%2==0) option1(arr,n);
else option2(arr,n);
               for( j=0;j<2*n;++j){
 //printf("%d",arr[j]);
                if(arr[j]!=j+1) break;
                if(j==2*n-1) t1=i+1;
                            if(t1!=-1) break;
              //printf("\n");
              for(i=0;i<2000;++i){
               if(i%2==0) option2(arr_2,n);
               else option1(arr_2,n);
               for(j=0;j<2*n;++j){
                if(arr_2[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:
    \blacksquare
            (4) WhatsApp
                                    × osrmktretelab
                                                                  × +
            ← → C 🕯 care.srmist.edu.in/srmktretelab/#/srmktretelab/student/home
                                                                                                                                                                      O- * * (1)
    0
                   M Gmail 🖪 YouTube 🧗 Maps 🦸 codetantra 🔼 GCR 🔝 Google Meet 😂 student login 🚱 Academia 📅 CIVIL AND MECH 💩 smktretelab 📝 C compiler 📫 empower 🐧 WhatsApp
    F
    Problem Description:
    0
  🔁 Untitle
                                         1≤t≤100
                                         4≤n≤109
    @
  08:00 PM
Thursday
01-07-202
                            Problem
                                        Explanation:
#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 \le 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");
```

if(arr[i]!=i+1) break;

```
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;
              (3) WhatsApp
    0
                     M Gmail 🖪 YouTube 🧗 Maps 🦸 codetantra 🔼 GCR 🔝 Google Meet 😂 student login 🚱 Academia 📅 CIVIL AND MECH 💩 smktretelab 📝 C compiler 📫 empower 🧿 WhatsApp
    P
    0
                                             Problem Description:
  🔁 Untitle
                                             1≤t≤100
4≤n≤109
                                             Input Format:
    @
                                             For each test case in the first line output the initial order of divisors, which are greater than 1, in the circle. In the second line output, a minimal number of moves needed to decrypt the message
  08:00 PM
Thursday
01-07-202
#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;
                               kd = 0;
                               for (d = 2; d \le 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;
                               }
```

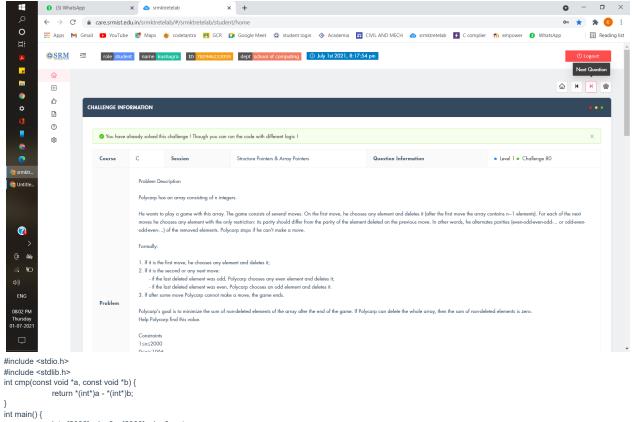
continue:

```
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:
     \blacksquare
              (3) WhatsApp × 🐽 srmktretelab
              ← → C 🔒 care.smist.edu.in/smktretelab/#/smktretelab/student/home
    0
             🔛 Apps M Gmail 💽 YouTube 🧗 Maps 🐐 codetantra 🔼 GCR 🔉 Google Meet 👶 student login 🚸 Academia 🛱 CIVIL AND MECH 💩 srmktretelab 📝 C complier 📫 empower 🗿 WhatsApp
                                                                                                                                                                                                Reading list
               +
    F
               ۵
     CHALLENGE INFORMATION
               A
     0
               (?)
                               You have already solved this challenge ! Though you can run the code with different logic!
                                                                                                                                                • Level 1 • Challenge 78
                                                                     Structure Pointers & Array Pointers
                                                                                                                          Question Information
                                              Vijay has given a set of points x1, x2, ..., xn on the number line.
                                              Two points i and j can be matched with each other if the following conditions hold:
   Untitle
                                                neither i nor j is matched with any other point;
                                                |xt-xt| \ge z.
                                              What is the maximum number of pairs of points you can match with each other?
                                              Constraints: 2 \le n \le 2 \cdot 10^5,
    @
                                                1 ≤ z ≤ 10^9
                                               1 ≤ xt ≤ 10^9
                               Problem
                                              The first line contains two integers n and z — the number of points and the constraint on the distance between matched points, respectively
                                               Print the output in a single line contains the maximum number of pairs of points you can match with each other.
   08:01 PM
Thursday
01-07-202
                                              Assume the input as follows:
                                              421337
#include<stdio.h>
#include<stdlib.h>
void i(){}
int comp(const void*a,const void*b)
{
                return *(int *)a - *(int *)b;
                if(0)printf("static int aa[N];*aa");
int main()
                int n, z, a[200009], i, sum=0;
                scanf("%d %d", &n, &z);
                for(i=0; i<n; i++)
                              scanf("%d", a+i);
                qsort(a, n, sizeof(int), comp);
                int I = 0, r = n&1 ? (n>>1)+1 : n>>1;
                for(i=0; i<n; i++)
                while(r < n)
                {
                               if(a[r]\text{-}a[l]>=z)
                                               sum++, I++;
                printf("%d", sum);
                return 0;
}
```



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

}



int o[2000], ol = 0, e[2000], el = 0, n, t; scanf("%d", &n); while(n--) { scanf("%d", &t); if(t % 2) o[ol++] = t;else e[el++] = t;qsort(o, ol, sizeof(int), cmp); qsort(e, el, sizeof(int), cmp); while(ol && el) { ol--: el--; } t = 0; if(ol) { ol--: while(ol) t += o[--ol];} else if(el) { el--; while(el) t += e[--el];printf("%d", t);

return 0;}