Course: C **Session:** Arrays **Timestamp:** 2021-1-9 19:06:14 **Register Number:** RA2031241010094

Q. Array Deletion

Write a program to delete an element from the array.

Input and Output Format:

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

Refer sample input and output for formatting specifications.

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

Input:
1. The number of inputs to be entered by the user
2. The array elements
3. The array index to be deleted

Example 1:
Input 1:
4
11 12 13 14
1

Output 1:
Array after deletion is
11 13 14

Explanation:
Array Index starts from 0 the user entered array index as 1 so the number 12 is deleted.
arr[0]=11
arr[11=12
arr[21=13
arr[3]=13

Deleted Element=12 a[1]

Source Code

```
#include <stdio.h>
int main()
{
  int n,i,ar[100],d;
  scanf("%d\n",&n);
  for(i=0;i<n;i++)
  {
    scanf("%d",&ar[i]);
  }
    scanf("%d",&d);
    if(d<0||d>n)
    {
      printf("Invalid position");
    }
    else
    {
       for(i=d;i<n-1;i++)
        {
          ar[i]=ar[i+1];
        }
        n--;
        printf("Array after deletion is\n");
        for(i=0;i<n;i++)
         {
           printf("%d ",ar[i]);
         }
     }
  return 0;
}
```

Sample Input

```
5
14 25 13 34 45
4
```

Sample Output

```
Array after deletion is
14 25 13 34
```

Result

Thus, Program " **Array Deletion** " has been successfully executed

Course: C **Session:** Arrays **Timestamp:** 2021-1-9 19:06:45 **Register Number:** RA2031241010094

Q. Catch the Second..

A bag contains set of integers and it is discovered by Ram. There is sheet in that bag. The person who found the bag should find the second largest number from the set of numbers. If the rightly found the will get a gold.. If you wish get the prize you can also try...

Source Code

```
#include <stdio.h>
int main()
{
  int i,ar[100],n,big,sbig,j=0;
    scanf("%d",&n);
  for(i=0;i<=n;i++)
    scanf("%d",&ar[i]);
    big=ar[0];
      for(i=1;i<n;i++)
      {
         if(big<ar[i])
           {
            big=ar[i];
           j=i;
           }
      }
  sbig=ar[n-j-1];
    for(i=1;i<n;i++)
    {
      if(sbig<ar[i]&&j!=i)
        sbig=ar[i];
    }
    printf("%d",sbig);
 return 0;
}
```

Sample Input

```
5
5 3 1 2 6
```

Sample Output

```
5
```

Result

Thus, Program " **Catch the Second..** " has been successfully executed

Course: C **Session:** Arrays **Timestamp:** 2021-1-9 19:07:13 **Register Number:** RA2031241010094

Q. Big Buddy

Princess Rupsa saw one of her friends playing a special game. Her friend gave a bag that has a set of stones with different weighs. Se is assigned with a task to add the weighs and also find the average weight of the bag.

Source Code

```
#include <stdio.h>
int main()
{
  int i,n;
  float ar[100],s=0.0;
  scanf("%d",&n);
  for(i=0;i<n;i++)
  {
    scanf("%f",&ar[i]);
      s=s+ar[i];
  }
 printf("%.1f\n",s);
  printf("%.1f",s/n);
 return 0;
}
```

Sample Input

```
3
2.0 4.0 5.0
```

Sample Output

```
11.0
3.7
```

Result

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

Course: C **Session:** Arrays **Timestamp:** 2021-1-9 19:07:39 **Register Number:** RA2031241010094

Q. Raju Dilemma

Given an array of size N and an integer X, Raju wants to know how many elements in the array are divisible by X. Help him find the answer.

Input Format

First line contains 2 integers N and X.
Second line contains N integers, the array arr[N].
1 <= N <= 100
1 <= arr[i], X <= 100

Output Format

One number denoting the count of elements that are divisible by X.

Source Code

```
#include <stdio.h>
int main()
{
  int i,n,x,ar[100],c=0;
  scanf("%d",&n);
  scanf("%d",&x);
  for(i=0;i<n;i++)
  {
    scanf("%d",&ar[i]);
    if(ar[i]%x==0)
      c++;
  }
 printf("%d",c);
 return 0;
}
```

Sample Input

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

Sample Output

```
3
```

Result

Thus, Program " **Raju Dilemma** " has been successfully executed

Course: C **Session:** Arrays **Timestamp:** 2021-1-9 19:08:07 **Register Number:** RA2031241010094

Q. Sum of odd numbers

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

Input Format:

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

Output Format:

Refer sample output for details.

Source Code

```
#include <stdio.h>
int main()
{
  int i,ar[100],n,sum=0;
  scanf("%d",&n);
  for(i=1;i<=n;i++)
    scanf("%d",&ar[i]);
    for(i=0;i<=n;i++)
    {
      if(ar[i]%2==1)
    sum=sum+ar[i];
  }
  printf("odd=%d",sum);
  return 0;
}
```

Sample Input

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

Sample Output

```
odd=9
```

Result

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

Q. Colored Array

"Chef had a hard time arguing with his friend, and after getting a great old kick Chef saw a colored array with N cells, numbered from 1 to N. Explanation:
For this sample, we can repaint only once, since K = 1. We should repaint 4th cell with color 1. We will pay 1 for this, and receive:
1 (1st cell - 1st color) +
1 (2nd cell - 1st color) +
1 (3rd cell - 2nd color) +
3 (4th cell - 1st color) = 6.
Hence we get 6 1 = 5 points in total, and it is the optimal answer.

The kick was so strong that Chef suddenly understood the rules of the game.
Each cell is painted with a color. Here the colors are numbered from 1 to M.
For any cell i, Chef can repaint it with any color q, and the cost of such operation is Ci,q points.
However Chef can do at most K repaintings (0 repaintings is possible).
After performing all repaintings, each cell will have some color. For each cell i, if cell i has color q then Chef will receive Bi,q points.
Now Chef is wondering how many points can he receive in total when he repaints optimally.
Input
The first line of the input contains an integer T, denoting the number of test cases. The description of T test cases follows.
The first line of each test case contains three space-separated integers N, M and K, denoting the number of cells and the number of colors, the maximal possible number of repaintings respectively. The next line contains N space-separated integers A1, A2, ..., AN, denoting the initial colors of the cells. Then N lines follow. The ith line of them contains M integers Bi1, Bi2, ..., BiM, where Bij denotes how many points Chef will receive if the cell i will be painted with j-th color after all operations. Then N lines follow. The ith line of them contains M integers Ci1, Ci2, ..., CiM, where Cij denotes how many points Chef will lose if he repaints the cell i with color j.
Note: Be careful that the size of input files can be large.
Output
For each test case, output a single line containing the maximal possible points.
Constraints
1 <= T <= 5
0 <= K <= 1000
1 <= N, M <= 1000
1 <= Ai <= M
0 0 < Cij < 1000
If j = Ai, then Cij = 0

Source Code

```
#include <stdio.h>
int main()
{
  int n,m,k,i,j,t;
  int col[1001],cost[1001][1001],points[1001][1001];
  scanf("%d",&t);
    while(t--)
    {
      scanf("%d %d %d\n",&n,&m,&k);
      int max[n+1],pts[n+1],temp,x,tot=0;
      for(i=1;i<=n;i++)
        max[i]=0;
      for(i=1;i<=n;i++)
        scanf("%d",&col[i]);
      for(i=1;i<=n;i++)
        for(j=1;j<=m;j++)
          scanf("%d",&points[i][j]);
      for(i=0;i<=n;i++)
        for(j=0;j<=m;j++)
          scanf("%d",&cost[i][j]);
      for(i=1;i<=n;i++)
        pts[i]=points[i][col[i]];
      for(i=1;i<=n;i++)
          {
            for(j=1;j<=m;j++)
            {
              x=(points[i][j]-cost[i][j]);
              if(max[i]<(x-pts[i]))
                max[i]=x-pts[i];
            }
      }
      if(k>n)
        k=n;
      for(i=1;i<=k;i++)
      {
        for(j=i+1;j<=n;j++)
        {
          if(max[i]<max[j])
          {
            temp=max[i];
            max[i]=max[j];
            max[j]=temp;
            temp=pts[i];
            pts[i]=pts[j];
            pts[j]=temp;
          }
        }
        pts[i]=pts[i]+max[i]-k;
      }
      for(i=1;i<=n;i++)
      {
        if(k>0)
        {
          tot=tot+pts[i];
        }
        else
        {
          tot=tot+pts[i];
        }
      }
      printf("%d",tot);
    }
 return 0;
}
```

Sample Input

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

Sample Output

```
5
```

Result

Thus, Program " **Colored Array** " has been successfully executed

Course: C **Session:** Arrays **Timestamp:** 2021-1-9 19:09:13 **Register Number:** RA2031241010094

Q. HOLIDAYS

"School holidays come in Berland. The holidays are going to continue for n days. The students of school No. N are having the time of their lives and the IT teacher Marina Sergeyevna, who has spent all the summer busy checking the BSE (Berland State Examination) results, has finally taken a vacation break! Some people are in charge of the daily watering of flowers in shifts according to the schedule. However when Marina Sergeyevna was making the schedule, she was so tired from work and so lost in dreams of the oncoming vacation that she perhaps made several mistakes. In fact, it is possible that according to the schedule, on some days during the holidays the flowers will not be watered or will be watered multiple times. Help Marina Sergeyevna to find a mistake.

Input
The first input line contains two numbers n and m (1n,m100) the number of days in Berland holidays and the number of people in charge of the watering respectively. The next m lines contain the description of the duty schedule. Each line contains two integers aiand bi (1aibin), meaning that the i-th person in charge should water the flowers from the ai-th to the bi-th day inclusively, once a day. The duty shifts are described sequentially, i.e. biai+1 for all i from 1 to n-1 inclusively.

Output
Print "OK"" (without quotes), if the schedule does not contain mistakes. Otherwise you have to find the minimal number of a day when the flowers will not be watered or will be watered multiple times, and output two integers the day number and the number of times the flowers will be watered that day.

Note
Keep in mind that in the second sample the mistake occurs not only on the second day, but also on the sixth day, when nobody waters the flowers. However, you have to print the second day, i.e. the day with the minimal number.

Source Code

```
#include <stdio.h>
int main()
{
  int n,m,a,b,i,j,h[100]={0};
  scanf("%d %d",&n,&m);
  for(i=1;i<=m;i++)
  {
    scanf("%d %d",&a,&b);
    for(j=a;j<=b;j++)
      h[j]++;
  }
  for(i=1;i<=n;i++)
     {
       if(h[i]!=1)
         break;
     }
  if(i==n+1)
    printf("OK\n");
  else
    printf("%d %d\n",i,h[i]);
  return 0;
}
```

Sample Input

```
10 5
1 2
3 3
4 6
7 7
8 10
```

Sample Output

```
OK
```

Result

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

Course: C **Session:** Arrays **Timestamp:** 2021-1-9 19:09:40 **Register Number:** RA2031241010094

Q. game

" You are playing following game: given an array A of N natural numbers. All numbers in the array A are at most M. On every turn you may pick any two different elements Ai and Aj (ij), such that Ai, Aj M, and add K to both. The game ends when you are not able to continue. That is, when there is no pair (i,j) left such that both of them are less than equal to M.

Let's call two arrays different if the sum of all their elements is different. When the game ends, you note down the final array A. How many different final arrays can you have.
Input

The first line contains three integers N, M and K. N elements of the array follow in the next line.
Constraints

1 <= N <= 1000000
1 <= M,K<= 10000000000000
1 <= Ai <= M
test case 1:All possible sums are 14 and 10. You can get them by, for example, these arrays:

A=(5, 4, 5),

A=(1, 4, 5)

The above arrays are different because their sums are different."

Source Code

```
#include <stdio.h>
int main()
{
  int n,m,k,i,sum=0,max=0,min=0,ans;
  int a[10000];
  scanf("%d %d %d",&n,&m,&k);
  for(i=0;i<=n;i++)
  {
    scanf("%d",&a[i]);
    a[i]=(m-a[i])/k+1;
    if(max<a[i])
      max=a[i];
    sum=sum+a[i];
  }
  if((sum-max)%2==0)
  {
    min=(sum-max)/2;
  }
    else
    {
    min=(sum-max)/2+1;
    }
      ans=sum/2-min+1;
 printf("%d",ans);
 return 0;
}
```

Sample Input

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

Sample Output

```
2
```

Result

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

Course: C **Session:** Arrays **Timestamp:** 2021-1-9 19:10:32 **Register Number:** RA2031241010094

Q. Sequence

Chef has a sequence of N numbers. He like a sequence better if the sequence contains his favorite sequence as a substring.

Given the sequence and his favorite sequence(F) check whether the favorite sequence is contained in the sequence

Input

The first line will contain the number of test cases and are followed by the cases.
Each test case consists of four lines: The length of the sequence, the sequence N,the length of F and the sequence F

Output

Print ""Yes"" if the sequence contains the favourite sequence int it otherwise print ""No""

Constraints
1<=T<=10

Source Code

```
#include <stdio.h>
int main()
{
  int t;
  scanf("%d",&t);
  while(t--)
  {
    int n1,n2,i,f=0,k=0;
    scanf("%d",&n1);
    int a[n1];
    for(i=0;i<n1;i++)
    {
      scanf("%d",&a[i]);
    }
    scanf("%d",&n2);
    int b[n2];
    for(i=0;i<n2;i++)
    {
      scanf("%d",&b[i]);
    }
    for(i=0;i<n1&&k<n2;i++)
    {
      if(a[i]==b[k])
      {
        f++;
        k++;
        if(f==n2)
          break;
      }
    }
    if(f>=n2)
      printf("Yes\n");
    else
      printf("No\n");
  }
  return 0;
}
```

Sample Input

```
2
6
1 2 3 4 5 6
3
2 3 4
6
22 5 6 33 1 4
2
4 15
```

Sample Output

```
Yes
No
```

Result

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