ROLL NO:230701036

NAME: ARUN MC

TOPIC: COMPETIVE PROGRAMMING

1-Finding Duplicates-O(n^2) Time Complexity,O(1) Space Complexity

AIM:

Given a read only array of n integers between 1 and n, find one number that repeats.

CODE:

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

}

INPUT:

First Line - Number of elements

n Lines - n Elements

OUTPUT:

	Input	Expected	Got	
✔	11 10 9 7 6 5 1 2 3 8 4 7	7	7	✔
✔	5 1 2 3 4 4	4	4	✔
✔	5 1 1 2 3 4	1	1	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

2-Finding Duplicates-O(n) Time Complexity,O(1) Space Complexity

AIM:

Given a read only array of n integers between 1 and n, find one number that repeats.

CODE:

```
#include<stdio.h>
int main()
{
   int a;
   scanf("%d",&a);
   int b[a],c=0;
   for(int i=0;i<a;i++)
     scanf("%d",&b[i]);
   for(int i=0;i<a;i++)
      for(int j=0;j<a;j++)
        if(b[i]==b[j]&&i!=j){
              c=b[i];
              break;}
   printf("%d",c);
}
```

INPUT:

First Line - Number of elements

n Lines - n Elements

OUTPUT:

	Input	Expected	Got	
✔	11 10 9 7 6 5 1 2 3 8 4 7	7	7	✔
✔	5 1 2 3 4 4	4	4	✔
✔	5 1 1 2 3 4	1	1	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

3-Print Intersection of 2 sorted arrays-O(m*n)Time Complexity,O(1) Space Complexity

AIM:

Given 2 sorted arrays, find all the elements which occur in both the arrays.

CODE:

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

} **INPUT:**

1

3 10 17 57

6 2 7 10 15 57 246

OUTPUT:

	Input	Expected	Got	
✔	1 3 10 17 57 6 2 7 10 15 57 246	10 57	10 57	✔
✔	1 6 1 2 3 4 5 6 2 1 6	1 6	1 6	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

4-Print Intersection of 2 sorted arrays-O(m+n)Time Complexity,O(1) Space Complexity

AIM:

Given 2 sorted arrays, find all the elements which occur in both the arrays.

INPUT:

The first line contains T, the number of test cases. Following T lines contain:

1. Line 1 contains N1, followed by N1 integers of the first array
2. Line 2 contains N2, followed by N2 integers of the second array

CODE:

```
#include<stdio.h>
int main(){
   int t;
   scanf("%d",&t);
   for(int k=0;k<t;k++){
      int n,m;
      scanf("%d",&n);
      int a[n];
      for(int i=0;i<n;i++)
        scanf("%d",&a[i]);
      scanf("%d",&m);
      int b[m];
      for(int i=0;i<m;i++)
        scanf("%d",&b[i]);
      for(int i=0;i<n;i++)
         for(int j=0;j<m;j++)
            if(a[i]==b[j]){
```

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

OUTPUT:

	Input	Expected	Got	
✔	1 3 10 17 57 6 2 7 10 15 57 246	10 57	10 57	✔
✔	1 6 1 2 3 4 5 6 2 1 6	1 6	1 6	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

5-Pair with Difference-O(n^2)Time Complexity,O(1) Space Complexity

AIM:

Given an array A of sorted integers and another non negative integer k, find if there exists 2 indices i and j such that A[j] - A[i] = k, i != j.

INPUT:

First Line n - Number of elements in an array

Next n Lines - N elements in the array

k - Non - Negative Integer

CODE:

```
#include<stdio.h>
int main(){
   int n,k,flg=0;
   scanf("%d",&n);
   int a[n];
   for(int i=0;i<n;i++)
    scanf("%d",&a[i]);
   scanf("%d",&k);
   for(int i=0;i<n;i++){
    for(int j=0;j<n&&i!=j;j++){
      if(a[i]-a[j]==k){
         flg=1;
         break;
      }}
      if(flg==1){
     printf("1");break;}
```

```
    }
  if(flg==0)
    printf("0");
}
```

OUTPUT:

	Input	Expected	Got	
✔	3 1 3 5 4	1	1	✔
✔	10 1 4 6 8 12 14 15 20 21 25 1	1	1	✔
✔	10 1 2 3 5 11 14 16 24 28 29 0	0	0	✔
✔	10 0 2 3 7 13 14 15 20 24 25 10	1	1	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

6-Pair with Difference -O(n) Time Complexity,O(1) Space Complexity

AIM:

Given an array A of sorted integers and another non negative integer k, find if there exists 2 indices i and j such that A[j] - A[i] = k, i != j.

INPUT:

First Line n - Number of elements in an array

Next n Lines - N elements in the array

k - Non - Negative Integer

CODE:

```
#include<stdio.h>
int main(){
   int n,k,flg=0;
   scanf("%d",&n);
   int a[n];
   for(int i=0;i<n;i++)
    scanf("%d",&a[i]);
   scanf("%d",&k);
   for(int i=0;i<n;i++){
    for(int j=0;j<n&&i!=j;j++){
      if(a[i]-a[j]==k){
         flg=1;
         break;
      }}
      if(flg==1){
     printf("1");break;}
```

```
    }
  if(flg==0)
    printf("0");
}
```

OUTPUT:

	Input	Expected	Got	
✔	3 1 3 5 4	1	1	✔
✔	10 1 4 6 8 12 14 15 20 21 25 1	1	1	✔
✔	10 1 2 3 5 11 14 16 24 28 29 0	0	0	✔
✔	10 0 2 3 7 13 14 15 20 24 25 10	1	1	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.