COMPETITIVE PROGRAMMING

M.MOHANA

231901031

1. Find Duplicate in Array.

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

Input Format:

First Line - Number of elements

n Lines - n Elements

Output Format:

Element x - That is repeated

Solution:

```
#include <stdio.h>


int main()
{
   int n;
   scanf("%d", &n);
   int arr[n];
   int f = 0;


   for (int i = 0; i < n; i++)
   {
      scanf("%d", &arr[i]);
   }


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

```
    {
        for (int j = i + 1; j < n; j++)
        {
            if (arr[i] == arr[j])
            {
                f = arr[i];
                break;
            }
        }
        if (f == 1) break;
    }

    printf("%d", f);
}
```

2. Find Duplicate in Array.

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

Input Format:

First Line - Number of elements

n Lines - n Elements

Output Format:

Element x - That is repeated

Solution:

```
#include <stdio.h>

int main()
{
```

```
    int n;
    scanf("%d", &n);
    int arr[n];

    for (int i = 0; i < n; i++)
    {
        scanf("%d", &arr[i]);
    }

    int sum = n * (n + 1) / 2;
    int newsum = 0;

    for (int i = 0; i < n; i++)
    {
        newsum += arr[i];
    }

    printf("%d", n - sum + newsum);
}
```

3. Find the intersection of two sorted arrays.

OR in other words,

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

Solution:

```
#include <stdio.h>

int main() {
```

```
int k;
scanf("%d", &k);  // Number of test cases

for (int x = 0; x < k; x++) {
   int m, n;
   scanf("%d", &m);  // Size of first array
   int a[m];
   for (int i = 0; i < m; i++) {
      scanf("%d", &a[i]);
   }

   scanf("%d", &n);  // Size of second array
   int b[n];
   for (int i = 0; i < n; i++) {
      scanf("%d", &b[i]);
   }

   // Find and print intersection
   for (int i = 0; i < m; i++) {
      for (int j = 0; j < n; j++) {
         if (a[i] == b[j]) {
            printf("%d ", a[i]);
            break;  // Avoid printing duplicates if repeated in b[]
         }
      }
   }
```

```
        printf("\n");
    }

    return 0;
}
```

4. Find the intersection of two sorted arrays.

OR in other words,

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

Solution:

```
#include <stdio.h>

int main() {
    int k;
    scanf("%d", &k);
    for (int x = 0; x < k; x++) {
        int m, n;
        scanf("%d", &m);
        int a[m];
        for (int i = 0; i < m; i++) {
            scanf("%d", &a[i]);
        }

        scanf("%d", &n);
        int b[n];
        for (int i = 0; i < n; i++) {
```

```
        scanf("%d", &b[i]);
    }

    int p = 0, q = 0;
    while (p < m && q < n) {
        if (a[p] < b[q])
            p++;
        else if (a[p] > b[q])
            q++;
        else {
            printf("%d ", a[p]);
            p++;
            q++;
        }
    }
  }
  return 0;
}
```

5. 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.

Solution:

```
#include <stdio.h>

int main() {
  int n, k, flag = 0;
  scanf("%d", &n);
```

```
    int arr[n];

    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    scanf("%d", &k);

    for (int i = 0; i < n; i++) {
        for (int j = i + 1; j < n; j++) {
            if (arr[j] - arr[i] == k) {
                flag = 1;
                break;
            }
        }
    }

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

6. 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.

Solution:

```
#include <stdio.h>

int main() {
```

```
int n, k, flag = 0;
scanf("%d", &n);
int arr[n];

for (int i = 0; i < n; i++) {
   scanf("%d", &arr[i]);
}

scanf("%d", &k);

int i = 0, j = 1;

while (j < n) {
   if (arr[j] - arr[i] == k) {
      if (i == j)
         j++;
      else {
         flag = 1;
         break;
      }
   }
   else if (arr[j] - arr[i] < k) {
      j++;
   }
   else {
      i++;
```

```
            if (i == j)
                j++;
        }
    }

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