

1. Given an array Arr of size N, print second largest distinct element from an array. Find the second largest without sorting.

Example 1:

Input:

N = 6

Arr[] = {12, 35, 1, 10, 34, 1}

Output: 34

Explanation: The largest element of the array is 35 and the second largest element is 34.

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

```
void findSecondLargest(int arr[], int size)
```

```
{
    int i;
    if (size < 2)
    {
        printf("Array size should be at least 2\n");
        return;
    }
    int largest = arr[0];
    int secondLargest = arr[1];

    if (largest < secondLargest)
    {
        int temp = largest;
        largest = secondLargest;
        secondLargest = temp;
    }

    for (i = 2; i < size; i++)
    {
        if (arr[i] > largest)
        {
            secondLargest = largest;
            largest = arr[i];
        }
    }
}
```

```

        else if (arr[i] > secondLargest && arr[i] != largest)
        {
            secondLargest = arr[i];
        }
    }

    printf("The second largest element is: %d\n", secondLargest);
}

int main() {
    int size,i;

    printf("Enter the size of the array: ");
    scanf("%d", &size);

    int arr[size];

    printf("Enter %d elements:\n", size);
    for (i = 0; i < size; i++)
    {
        scanf("%d", &arr[i]);
    }

    findSecondLargest(arr, size);

    return 0;
}

```

```

Enter the size of the array: 6
Enter 6 elements:
12
35
1
10
34
1
The second largest element is: 34

-----
Process exited after 15.91 seconds with return value 0
Press any key to continue . . . █

```

2. Given an array Arr of N positive integers and another number X. Determine whether or not there exist two elements in Arr whose sum is exactly X. [Without Sorting]

Example 1:

Input:

N = 6, X = 16

Arr[] = {1, 4, 45, 6, 10, 8}

Output: Yes

Explanation: Arr[3] + Arr[4] = 6 + 10 = 16

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

```
int check(int arr[], int n, int x)
{
    int i,j;
    for (i = 0; i < (n - 1); i++)
    {
        for (j = (i + 1); j < n; j++)
        {
            if (arr[i] + arr[j] == x)
            {
                return 1;
            }
        }
    }

    return 0;
}
```

```
int main()
{
    int n,i,x;
    printf("Enter n:");
    scanf("%d",&n);
    int arr[n];
    printf("\nEnter the %d array elements:");
    for(i = 0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    printf("\nEnter the value of X:");
    scanf("%d",&x);
    if (check(arr, n, x))
    {
        printf("Yes\n");
    }
    else
    {
        printf("No\n");
    }

    return 0;
}
```

```
Enter n:6
Enter the 6 array elements:1 4 45 6 10 8
Enter the value of X:16
Yes
-----
Process exited after 59.85 seconds with return value 0
Press any key to continue . . .
```

3.First and last occurrences of x

Given a sorted array arr containing n elements with possibly some duplicate, the task is to find the first and last occurrences of an element x in the given array.

Note: If the number x is not found in the array then return both the indices as -1.

Example 1:

Input:

n=9, x=5

arr[] = { 1, 3, 5, 5, 5, 5, 67, 123, 125 }

Output:

2 5

Explanation: First occurrence of 5 is at index 2 and last occurrence of 5 is at index 5.

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

```
void check(int arr[], int n, int x)
```

```
{
    int i, j, start = -1, end = -1;
    for (i = 0; i < n; i++)
    {
        if (arr[i] == x)
        {
            start = i;
            for (j = i + 1; j < n; j++)
            {
                if (arr[i] == arr[j])
                {
                    end = j;
                }
            }
        }
    }
}
```

```

        break;
    }
}
if (start != -1)
{
    printf("\nStart: %d", start);
    printf("\nEnd: %d", end);
}
else
{
    printf("\nNumber %d not found in the array.", x);
}
}

```

```

int main() {
    int n, i, x;
    printf("Enter n: ");
    scanf("%d", &n);
    int arr[n];
    printf("\nEnter the %d array elements: ", n);
    for (i = 0; i < n; i++)
    {
        scanf("%d", &arr[i]);
    }
    printf("Enter x: ");
    scanf("%d", &x);
    check(arr, n, x);
    return 0;
}

```

```

Enter n: 9
Enter the 9 array elements: 1 3 5 5 5 5 67 123 125
Enter x: 5

Start: 2
End: 5
-----
Process exited after 48.77 seconds with return value 0
Press any key to continue . . .

```