**FUNCTIONS**

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.

Example 2:  
Input:  
N = 3  
Arr[] = {10, 5, 10}  
Output: 5  
Explanation: The largest element of  
the array is 10 and the second  
largest element is 5.

**CODE:**

#include <stdio.h>

void findSecondLargest(int arr[], int n) {

int firstLargest = arr[0];

int secondLargest = -1;

for (int i = 1; i < n; i++) {

if (arr[i] > firstLargest) {

secondLargest = firstLargest;

firstLargest = arr[i];

} else if (arr[i] < firstLargest && arr[i] > secondLargest) {

secondLargest = arr[i];

}

}

if (secondLargest == -1) {

printf("There is no second largest distinct element.\n");

} else {

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

}

}

int main() {

int N;

printf("Enter the size of the array: ");

scanf("%d", &N);

int Arr[N];

printf("Enter the elements of the array:\n");

for (int i = 0; i < N; i++) {

scanf("%d", &Arr[i]);

}

findSecondLargest(Arr, N);

return 0;

}

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

Example 2:  
Input:  
N = 5, X = 10  
Arr[] = {1, 2, 4, 3, 6}  
Output: Yes  
Explanation: Arr[2] + Arr[4] = 4 + 6 = 10

**CODE:**

#include <stdio.h>

int hasPairWithSum(int arr[], int n, int X) {

int present[X+1];

for (int i = 0; i <= X; i++) {

present[i] = 0;

}

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

int complement = X - arr[i];

if (present[complement] == 1) {

return 1; // Pair found

}

present[arr[i]] = 1;

}

return 0; // No pair found

}

int main() {

int N, X;

printf("Enter the size of the array: ");

scanf("%d", &N);

printf("Enter the target sum X: ");

scanf("%d", &X);

int Arr[N];

printf("Enter the elements of the array:\n");

for (int i = 0; i < N; i++) {

scanf("%d", &Arr[i]);

}

if (hasPairWithSum(Arr, N, X)) {

printf("Yes, there exist two elements whose sum is %d.\n", X);

} else {

printf("No, there are no two elements whose sum is %d.\n", X);

}

return 0;

}

**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.

Example 2:  
Input:  
n=9, x=7  
arr[] = { 1, 3, 5, 5, 5, 5, 7, 123, 125 }  
Output:  
6 6  
Explanation: First and last occurrence of 7 is at index 6.

**CODE:**

#include <stdio.h>

void findOccurrences(int arr[], int n, int x) {

int firstOccurrence = -1, lastOccurrence = -1;

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

if (arr[i] == x) {

firstOccurrence = i;

break;

}

}

for (int i = n - 1; i >= 0; i--) {

if (arr[i] == x) {

lastOccurrence = i;

break;

}

}

printf("First occurrence: %d\n", firstOccurrence);

printf("Last occurrence: %d\n", lastOccurrence);

}

int main() {

int n, x;

printf("Enter the size of the array: ");

scanf("%d", &n);

int arr[n];

printf("Enter the sorted elements of the array:\n");

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

scanf("%d", &arr[i]);

}

printf("Enter the element to search: ");

scanf("%d", &x);

findOccurrences(arr, n, x);

return 0;

}