**QUIZ 8**

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.

#include <stdio.h>

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

if (n < 2) {

printf("Array size should be at least 2\n");

return;

}

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("All elements are the same in the array\n");

} else {

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

}

}

int main() {

int N1 = 6;

int Arr1[] = {12, 35, 1, 10, 34, 1};

findSecondLargest(Arr1, N1);

int N2 = 3;

int Arr2[] = {10, 5, 10};

findSecondLargest(Arr2, N2);

return 0;

}

OUTPUT:

The second largest distinct element is: 34

The second largest distinct element is: 5

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

#include <stdio.h>

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

int hashTable[100000] = {0};

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

int complement = X - arr[i];

if (hashTable[complement] == 1) {

return 1; // Yes, there exists a pair

}

hashTable[arr[i]] = 1;

}

return 0;

}

int main() {

int N1 = 6, X1 = 16;

int Arr1[] = {1, 4, 45, 6, 10, 8};

if (hasPairWithSum(Arr1, N1, X1)) {

printf("Yes\n");

} else {

printf("No\n");

}

int N2 = 5, X2 = 10;

int Arr2[] = {1, 2, 4, 3, 6};

if (hasPairWithSum(Arr2, N2, X2)) {

printf("Yes\n");

} else {

printf("No\n");

}

return 0;

}

OUTPUT:

Yes

Yes

**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.

#include <stdio.h>

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

int firstOccurrence = -1;

int 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("%d %d\n", firstOccurrence, lastOccurrence);

}

int main() {

int n1 = 9, x1 = 5;

int arr1[] = {1, 3, 5, 5, 5, 5, 67, 123, 125};

findFirstAndLast(arr1, n1, x1);

int n2 = 9, x2 = 7;

int arr2[] = {1, 3, 5, 5, 5, 5, 7, 123, 125};

findFirstAndLast(arr2, n2, x2);

return 0;

}

OUTPUT:

2 5

6 6