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

#include <stdio.h>

void print\_largest\_array(int \*arr, int size) {

int firstlargest = arr[0];

int secondlargest = -1;

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

if (arr[i] > firstlargest) {

secondlargest = firstlargest;

firstlargest = arr[i];

} else if ((arr[i] < firstlargest) && ((secondlargest == -1) || (arr[i] > secondlargest))) {

secondlargest = arr[i];

}

}

if (secondlargest != -1) {

printf(" %d\n", secondlargest);

} else {

printf("No second largest distinct element found.\n");

}

}

int main() {

int n, i;

printf("Enter the array size:");

scanf("%d", &n);

int array[n];

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

printf("Enter the element %d: ", i);

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

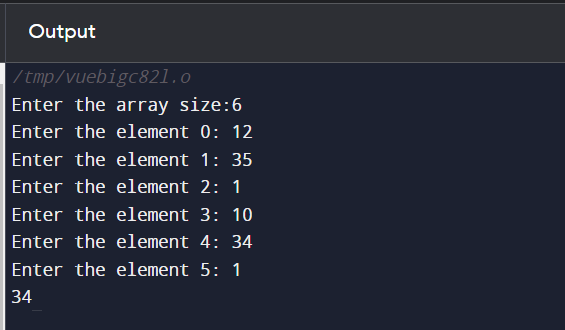
}

int size = sizeof(array) / sizeof(array[0]);

print\_largest\_array(array, size);

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]**

#include <stdio.h>

//N = 6, X = 16

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

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

int visited[100000] = {0};

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

int complement = X - arr[i];

if (visited[complement]) {

return 1; // Pair found

}

visited[arr[i]] = 1;

}

return 0;

}

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, sum is %d\n", X);

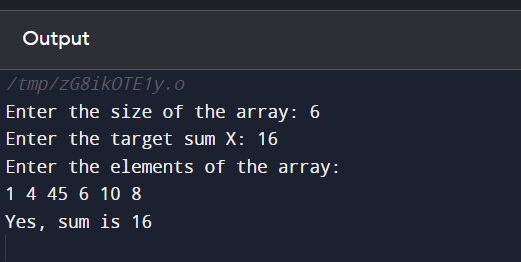
} else {

printf("No,sum is %d\n", X);

}

return 0;

}



**3. First and last occurrences of x**

#include <stdio.h>

//n=9, x=7

//1, 3, 5, 5, 5, 5, 7, 123, 125

void first\_and\_last\_occurrences(int arr[], int n, int x, int \*first\_index, int \*last\_index) {

\*first\_index = -1;

\*last\_index = -1;

// Binary search for the first occurrence

int left = 0, right = n - 1;

while (left <= right) {

int mid = left + (right - left) / 2;

if (arr[mid] == x) {

\*first\_index = mid;

right = mid - 1;

} else if (arr[mid] < x) {

left = mid + 1;

} else {

right = mid - 1;

}

}

// Binary search for the last occurrence

left = 0;

right = n - 1;

while (left <= right) {

int mid = left + (right - left) / 2;

if (arr[mid] == x) {

\*last\_index = mid;

left = mid + 1;

} else if (arr[mid] < x) {

left = mid + 1;

} else {

right = mid - 1;

}

}

}

int main() {

int num, i, x;

printf("Enter the array size:\n");

scanf("%d", &num);

int arr[num];

for (i = 0; i < num; i++) {

printf("Enter element %d: ", i);

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

}

printf("Enter the number to search for:");

scanf("%d", &x);

int first, last;

first\_and\_last\_occurrences(arr, num, x, &first, &last);

if (first != -1 && last != -1) {

printf("First occurrence of %d is at index %d\n", x, first);

printf("Last occurrence of %d is at index %d\n", x, last);

} else {

printf("The number %d is not found in the array.\n", x);

}

return 0;

}