**WEEK 1**

1. Given an array of non-negative integers, design a linear algorithm and implement it using a program to find whether given key element is present in the array or not. Also, find total number of comparisons for each input case. (Time Complexity = 0(n), where n is the size of input).

**Sample I/O Problem - 1:**

|  |  |
| --- | --- |
| **Input:**  3  8  34 35 65 31 25 89 64 30  89  5  977 354 244 546 355  244  6  23 64 137 67 43 56  63 | **Output:**  Present 6  Present 3  Not Present 6 |

**Program:**

//06:06PM 20-07-2021

#include<stdio.h>

#include<stdlib.h>

int main(){

int i, j, loop, n, element;

int steps = 0, found = 0;

int \*arr = (int \*) malloc(sizeof(int));

scanf("%d", &loop);

for(i=0; i<= loop-1; i++){

steps = found = 0;

scanf("%d", &n);

arr = (int \*) realloc(arr, n);

for(j=0; j<=n-1; j++){

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

}

scanf("%d", &element);

for(j=0; j<=n-1; j++){

steps++;

if(element == arr[j]){

found = 1;

break;

}

}

if(found == 1){

printf("Present %d\n", steps);

}else{

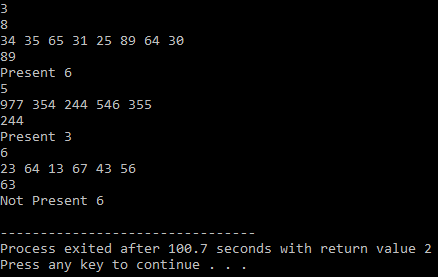
printf("Not Present %d\n", steps);

}

}

}

**Output-**



1. Given an already sorted array of positive integers, design an algorithm and implement it using a program to find whether given key element is present in the array or not. Also, find total number of comparisons for each input case. (Time Complexity = 0(n log(n)), where n is the size of input).

**Sample I/O Problem - 1:**

|  |  |
| --- | --- |
| **Input:**  3  8  25 30 31 34 35 64 65 89  89  5  244 354 355 546 977  244  6  23 43 56 64 67 137  63 | **Output:**  Present  Present  Not Present |

**Program:**

//06:06PM 20-07-2021

#include<stdio.h>

#include<stdlib.h>

int main(){

int i, j ,k;

int loop, n;

int first, mid, last, element;

int arr[100];

int steps, found, temp;

scanf("%d", &loop);

for(i=0; i<=loop-1; i++){

steps = found = 0;

scanf("%d", &n);

for(j=0; j<=n-1; j++){

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

}

scanf("%d", &element);

if(element % 2 == 0){

mid = n/2;

}else{

mid = (n+1)/2;

}

if(element < arr[mid]){

first = 0;

last = mid;

}else if(element >= arr[mid]){

first = mid;

last = n;

}

for(j=first; j<=last-1; j++){

steps++;

if(element == arr[j]){

found = 1;

break;

}

}

if(found == 1){

printf("Present %d\n", steps);

}else{

printf("Not Present %d\n", steps);

}

}

}

**Output:**

|  |
| --- |
|  |