

# LAB REPORT

---

CSE 114 : Data Structure and Algorithms Sessional

PREPARED BY

Mehrin Farzana

ID: 2101013

Session: 2021-2022

Date: 16/07/2023

SUPERVISED BY

Suman Saha

Lecturer

Department of IRE, BDU



BANGABANDHU SHEIKH MUJIBUR

RAHMAN DIGITAL UNIVERSITY

(BDU)

## List of Problems

1. Write a program that reads n numbers from the user and stores them in an array. The program reads a number between 0-100 and shows the number at the corresponding index number. For example, N=4, array [4] = {50, 65, 34, 67}, if the user enters 65, your program should print array [2].
2. Write a program that reads n numbers from the user and stores them in an array. Now perform the following operation using the switch statement. i. Prints the even numbers in reverse order. ii. Read a number from the user, and print “YES” if the number exists in the array. Otherwise print “NO”
3. Write a program that reads n numbers from the user, but does not allow the user to enter duplicates. This means that if a number has been entered already, the program will not accept it as input again and instead ask the user to enter a different number.
4. Given two integer arrays of same size, “arr[]” and “index[]”, reorder elements in “arr[]” according to given index array. It is not allowed to given array arr’s length.

Input:            arr[] = [10, 11, 12]  
                  index[] = [1, 0, 2]

Output:           arr[] = [11, 10, 12]  
                  index[] = [0, 1, 2]

Input:            arr[] = [50, 40, 70, 60, 90]  
                  index[] = [3, 0, 4, 1, 2]

Output:           arr[] = [40, 60, 90, 50, 70]  
                  index[] = [0, 1, 2, 3, 4]

**Problem No.: 01****Problem Statement:**

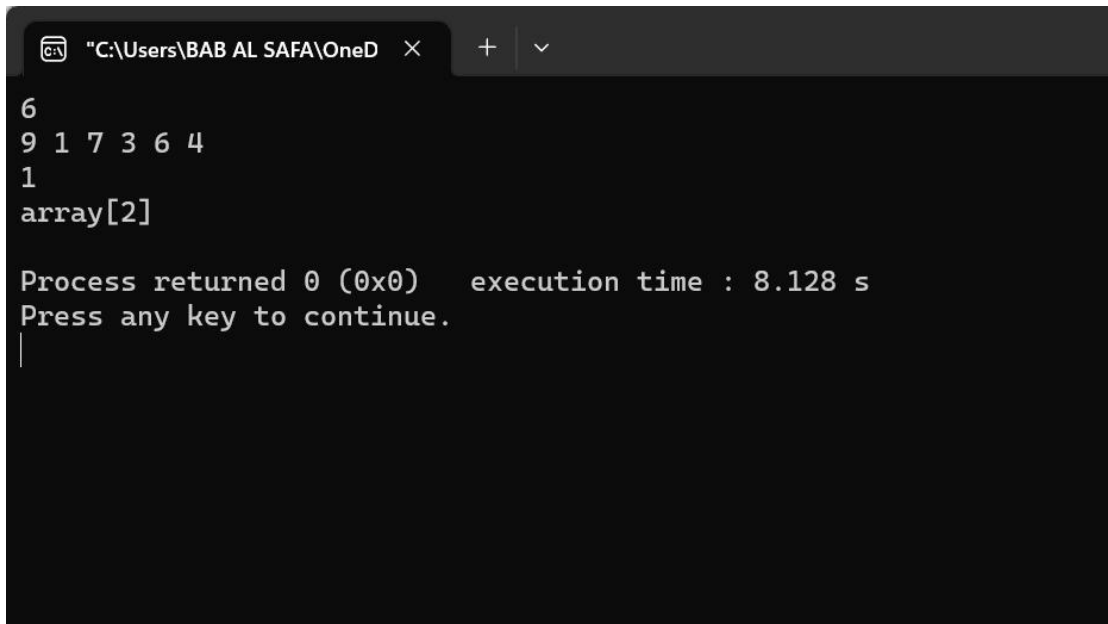
Write a program that reads n numbers from the user and stores them in an array. The program reads a number between 0-100 and shows the number at the corresponding index number. For example, N=4, array [4] = {50, 65, 34, 67}, if the user enters 65, your program should print array [2].

**Code:**

```
#include <stdio.h>
int main() {
    int n, x, i=0;
    scanf("%d", &n);
    int a[n];
    for(int i=0; i<n; i++)
        scanf("%d", &a[i]);
    scanf("%d", &x);
    while(i<n){
        if(a[i]==x){
            printf("array[%d]\n", i+1);
            break;
        }
        i++;
    }

    return 0;
}
```

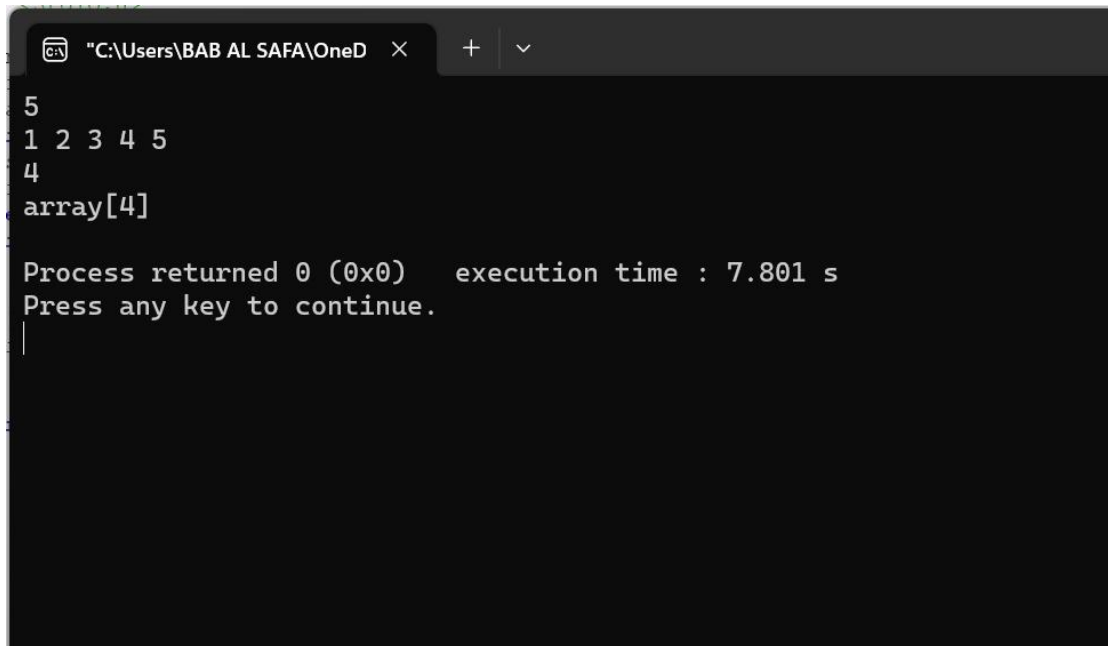
## Output:



```
"C:\Users\BAB AL SAFA\OneD" x + v
6
9 1 7 3 6 4
1
array[2]

Process returned 0 (0x0)   execution time : 8.128 s
Press any key to continue.
|
```

Fig 1.1: Output on console for case 1.



```
"C:\Users\BAB AL SAFA\OneD" x + v
5
1 2 3 4 5
4
array[4]

Process returned 0 (0x0)   execution time : 7.801 s
Press any key to continue.
|
```

Fig 1.2: Output on console for case 2.

## Problem No.: 02

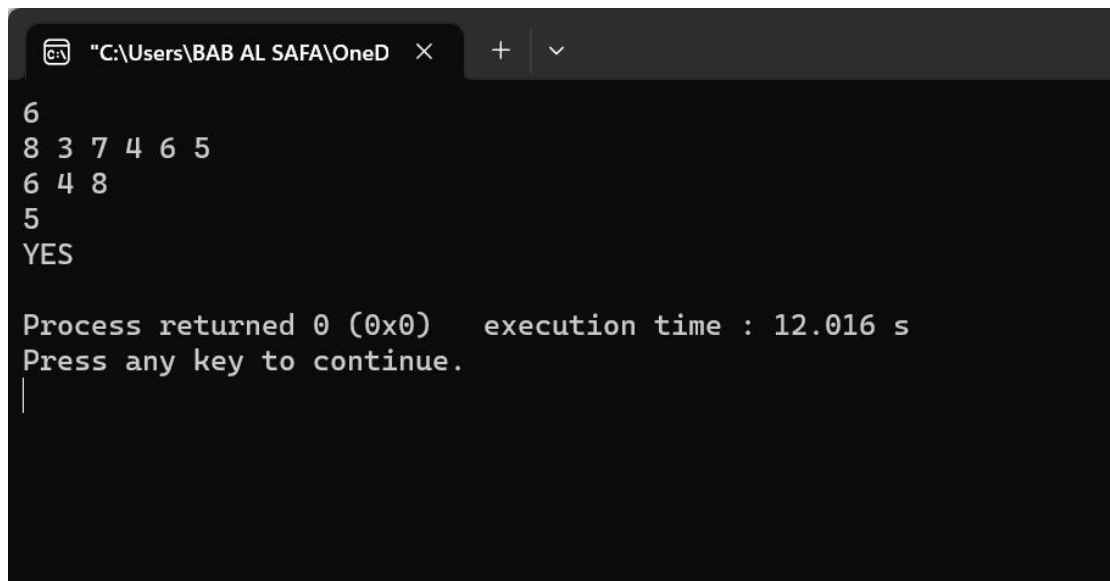
### Problem Statement:

Write a program that reads n numbers from the user and stores them in an array. Now perform the following operation using the switch statement. i. Prints the even numbers in reverse order. ii. Read a number from the user, and print “YES” if the number exists in the array. Otherwise print “NO”

### Code:

```
#include <stdio.h>
int main() {
    int n, x, h, l, m, flag=0;
    scanf("%d", &n);
    int a[n];
    for(int i=0; i<n; i++)
        scanf("%d", &a[i]);
    for(int i=n-1; i>=0; i--)
        if(a[i]%2==0)
            printf("%d ", a[i]);
    printf("\n");
    for(int i=0; i<n-1; i++){
        for(int j=0; j<n-1-i; j++){
            if(a[j]>a[j+1]){
                int tmp = a[j];
                a[j] = a[j+1];
                a[j+1] = tmp;
            }
        }
    }
    scanf("%d", &x);
    h=n, l=0;
    while(l<=h){
        m=(l+h)/2;
        if(a[m]==x){
            printf("YES\n");
            flag=1;
            break;
        }
        else if(a[m]<x)
            l=m+1;
        else if(a[m]>x)
            h=m-1;
    }
    if(!flag)
        printf("NO");
    return 0;
}
```

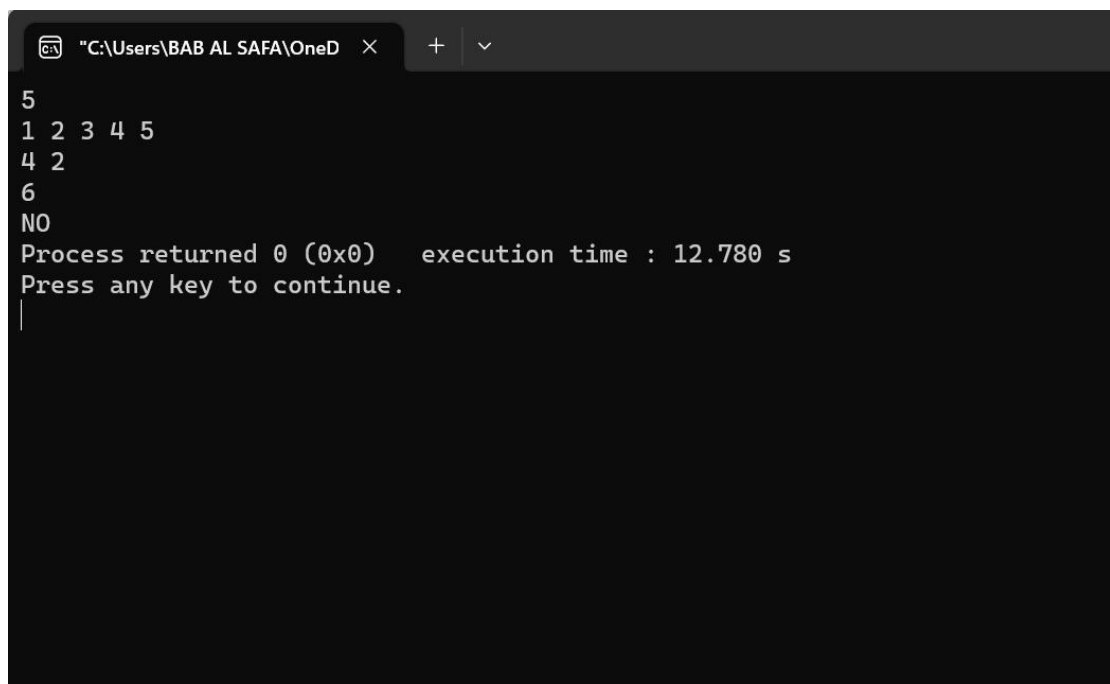
## Output:



```
C:\Users\BAB AL SAFA\OneD
6
8 3 7 4 6 5
6 4 8
5
YES

Process returned 0 (0x0)   execution time : 12.016 s
Press any key to continue.
|
```

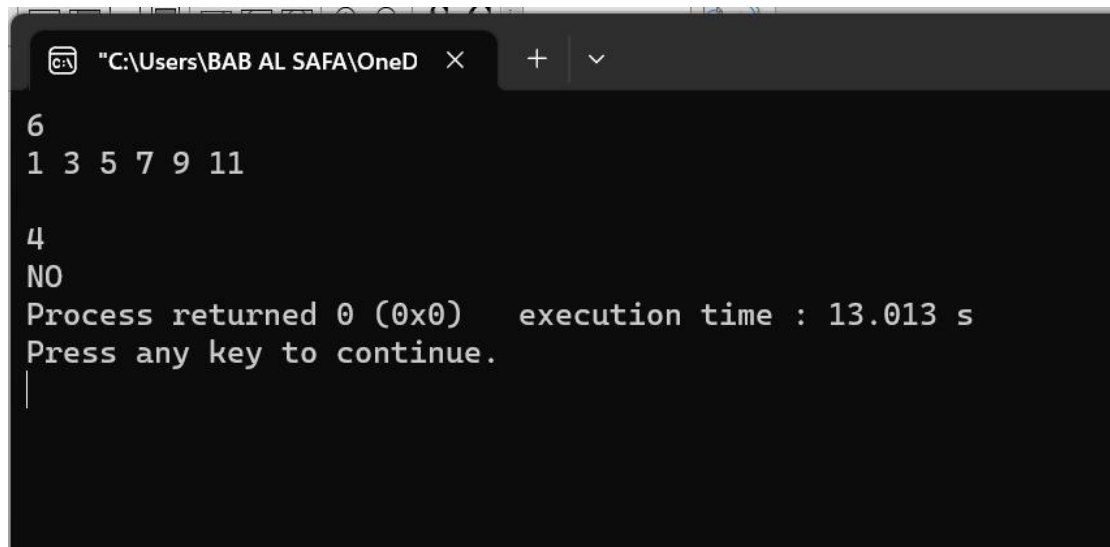
Fig 2.1: Output on console for case 1.



```
C:\Users\BAB AL SAFA\OneD
5
1 2 3 4 5
4 2
6
NO

Process returned 0 (0x0)   execution time : 12.780 s
Press any key to continue.
|
```

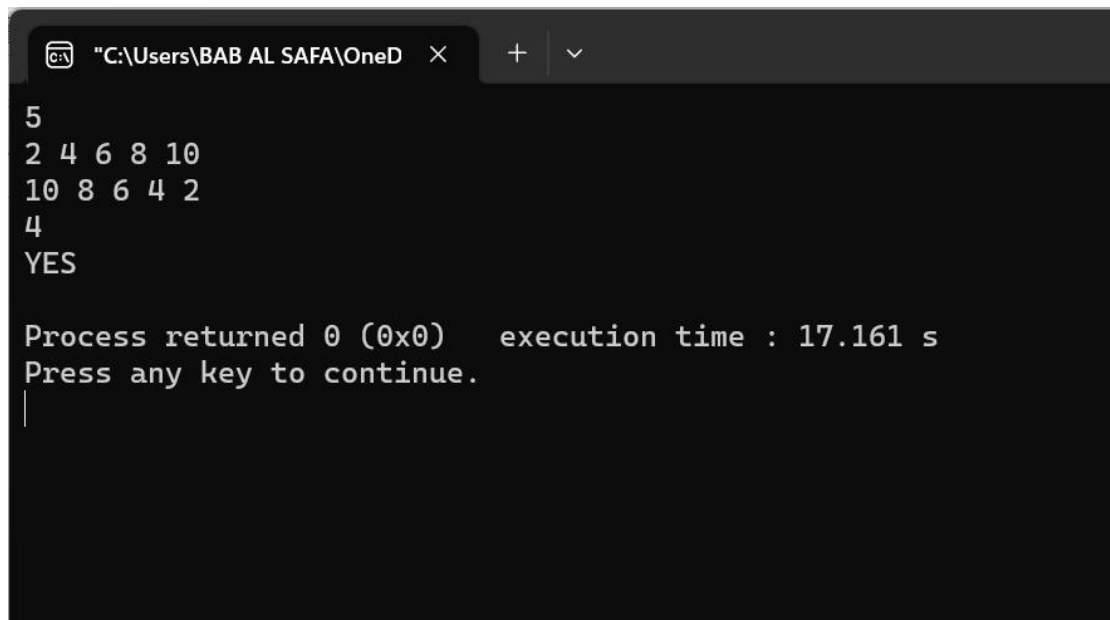
Fig 2.2: Output on console for case 2.



```
"C:\Users\BAB AL SAFA\OneD"
6
1 3 5 7 9 11

4
NO
Process returned 0 (0x0)   execution time : 13.013 s
Press any key to continue.
|
```

Fig 2.3: Output on console for case 3.



```
"C:\Users\BAB AL SAFA\OneD"
5
2 4 6 8 10
10 8 6 4 2
4
YES

Process returned 0 (0x0)   execution time : 17.161 s
Press any key to continue.
|
```

Fig 2.4: Output on console for case 4.

**Problem No.: 03****Problem Statement:**

Write a program that reads n numbers from the user, but does not allow the user to enter duplicates. This means that if a number has been entered already, the program will not accept it as input again and instead ask the user to enter a different number.

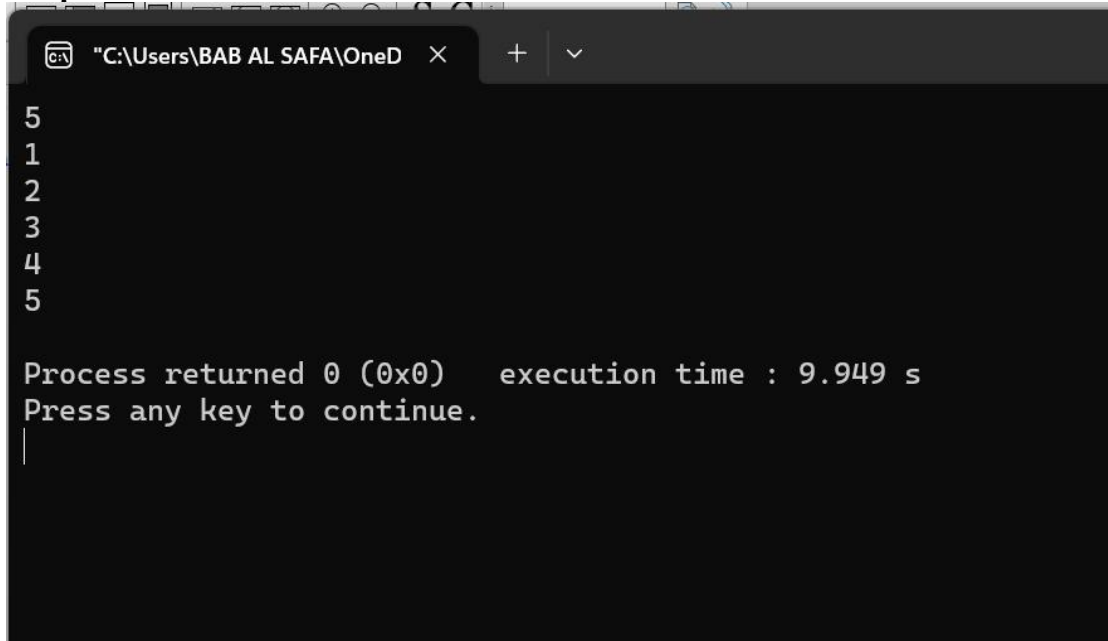
**Code:**

```
#include <stdio.h>

int main() {
    int n, x, i=0;
    scanf("%d", &n);
    int a[n];
    for(int i=0; i<n; i++){
        scanf("%d", &a[i]);
        for(int j=0; j<i; j++){
            if(a[j]==a[i]){
                printf("Enter a different number\n");
                i--;
            }
        }
    }
    return 0;
}
```

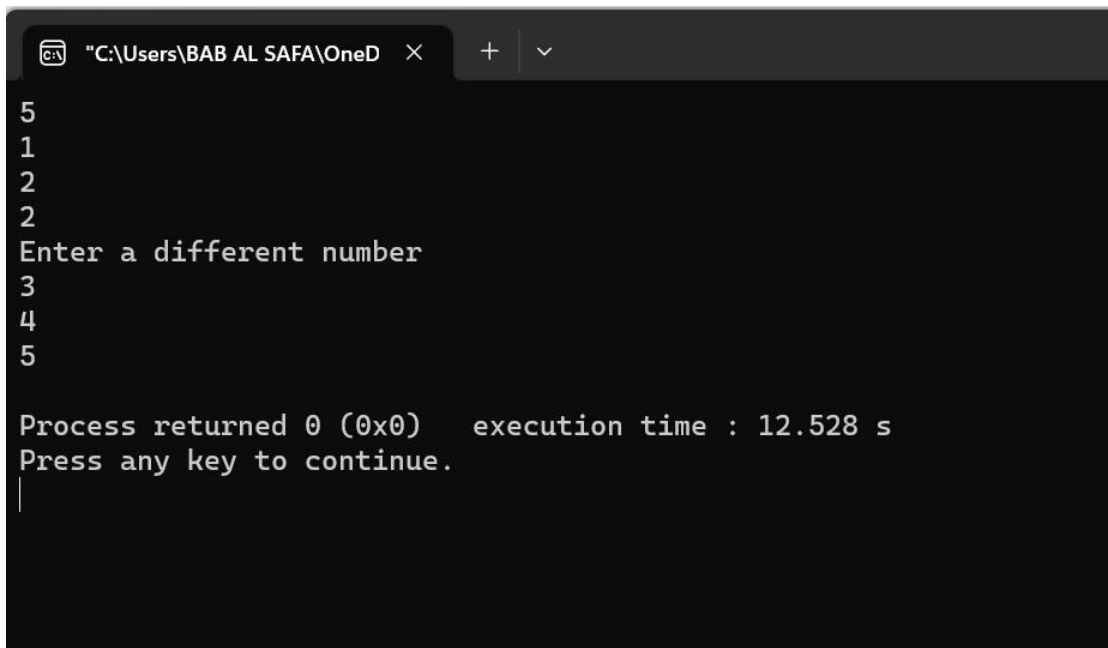


### Output:



```
"C:\Users\BAB AL SAFA\OneD" X + v
5
1
2
3
4
5
Process returned 0 (0x0) execution time : 9.949 s
Press any key to continue.
|
```

Fig 3.1: Output on console for case 1.



```
"C:\Users\BAB AL SAFA\OneD" X + v
5
1
2
2
Enter a different number
3
4
5
Process returned 0 (0x0) execution time : 12.528 s
Press any key to continue.
|
```

Fig 3.2: Output on console for case 2.

## Problem No.: 04

### Problem Statement:

Given two integer arrays of same size, “arr[]” and “index[]”, reorder elements in “arr[]” according to given index array. It is not allowed to given array arr’s length.

Input:            arr[] = [10, 11, 12]  
                  index[] = [1, 0, 2]

Output:          arr[] = [11, 10, 12]  
                  index[] = [0, 1, 2]

Input:            arr[] = [50, 40, 70, 60, 90]  
                  index[] = [3, 0, 4, 1, 2]

Output:          arr[] = [40, 60, 90, 50, 70]  
                  index[] = [0, 1, 2, 3, 4]

### Code:

```
#include <stdio.h>

void bubble_sort(int *arr, int *index, int n){
    for(int i=0; i<n-1; i++){
        for(int j=0; j<n-1-i; j++){
            if(index[j]>index[j+1]){
                int tmp1 = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = tmp1;
                int tmp2 = index[j];
                index[j] = index[j+1];
                index[j+1] = tmp2;
            }
        }
    }
}

void print_array(int *a, int *indx, int n){
    printf("arr[] = [");
    for(int i=0; i<n; i++){
        if(i<n-1)
            printf("%d, ", a[i]);
        else
            printf("%d", a[i]);
    }
    printf("]\n");
}
```

```

printf("index[] = [");
for(int i=0; i<n; i++){
    if(i<n-1)
        printf("%d, ", indx[i]);
    else
        printf("%d", indx[i]);
}
printf("]\n");
}

int main() {
    int arr1[]={10, 11, 12};
    int index1[]={1, 0, 2};

    int arr2[]={50, 40, 70, 60, 90};
    int index2[]={3, 0, 4, 1, 2};

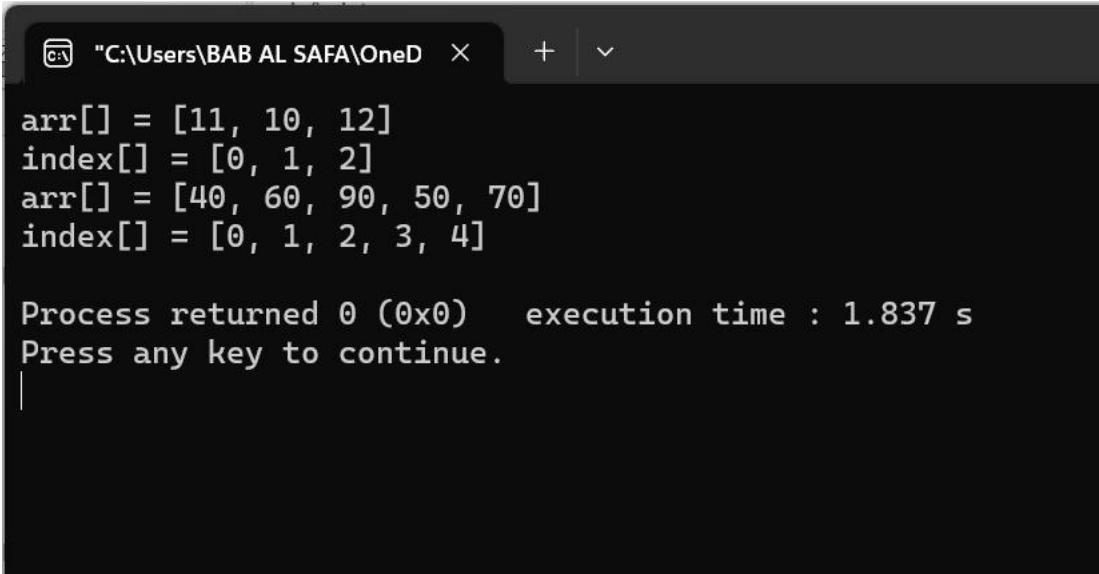
    int n1=sizeof(arr1)/sizeof(arr1[0]);
    int n2=sizeof(arr2)/sizeof(arr2[0]);
    bubble_sort(arr1, index1, n1);
    bubble_sort(arr2, index2, n2);

    print_array(arr1, index1, n1);
    print_array(arr2, index2, n2);

    return 0;
}

```

## Output:



```
"C:\Users\BAB AL SAFA\OneD  ×  +  v
arr[] = [11, 10, 12]
index[] = [0, 1, 2]
arr[] = [40, 60, 90, 50, 70]
index[] = [0, 1, 2, 3, 4]

Process returned 0 (0x0)    execution time : 1.837 s
Press any key to continue.
|
```

Fig 4.1: Output on console.