Worksheet 1.2

Student Name: Garv Khurana **UID:** 21BCS6615

Branch: AIT - CSE - AIML **Section/Group: 21AML - 9 - "A"**

Semester: 3rd

Subject Name: Data Structures **Subject Code:** 21CSH-241

Program 1

1. **Aim/Overview of the practical:** To create a program to remove an element from an array

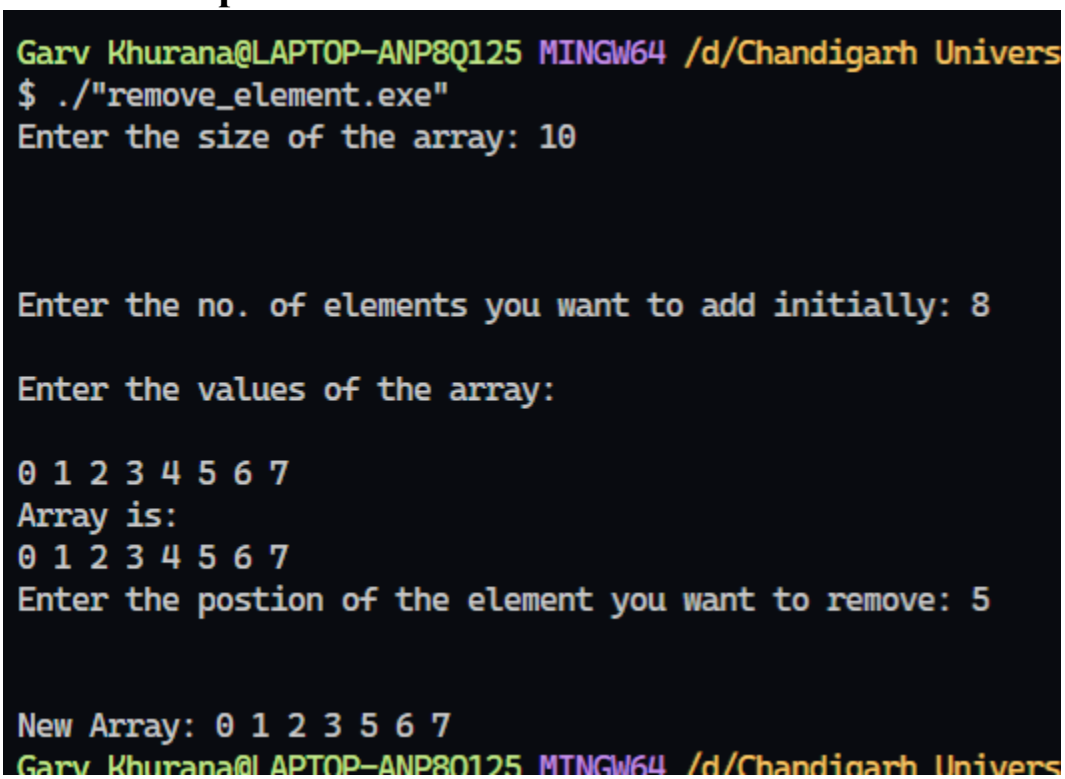
2. **Code:**

```
#include <stdio.h>

#include <stdlib.h>

void arrayScanner(int *arr, int length){

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

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

    }

}

void arrayPrinter(int *arr, int length){

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

        printf("%d ", arr[i]);

    }}
```

```
void arrayRemove(int *arr, int position, int *length){
    for (int i = position; i < *length; i++) {
        arr[i] = arr[i + 1];
    }
    (*length)--;
}
int main(){
    int size;
    int length;
    int *arr;
    printf("Enter the size of the array: ");
    scanf("%d", &size);
    printf("\n");
    arr = (int *)malloc(size * sizeof(int));
    int position;
    printf("\n");
    printf("\nEnter the no. of elements you want to add initially: ");
    scanf("%d", &length);
    printf("\n");
    printf("Enter the values of the array: \n\n");
    arrayScanner(arr, length);
    printf("Array is: \n");
    arrayPrinter(arr, length);
    printf("\nEnter the postion of the element you want to remove: ");
```

```
    scanf("%d", &position);

    arrayRemove(arr, position - 1, &length);

    printf("\n\nNew Array: ");

    arrayPrinter(arr, length);

}
```

3. Code Output:



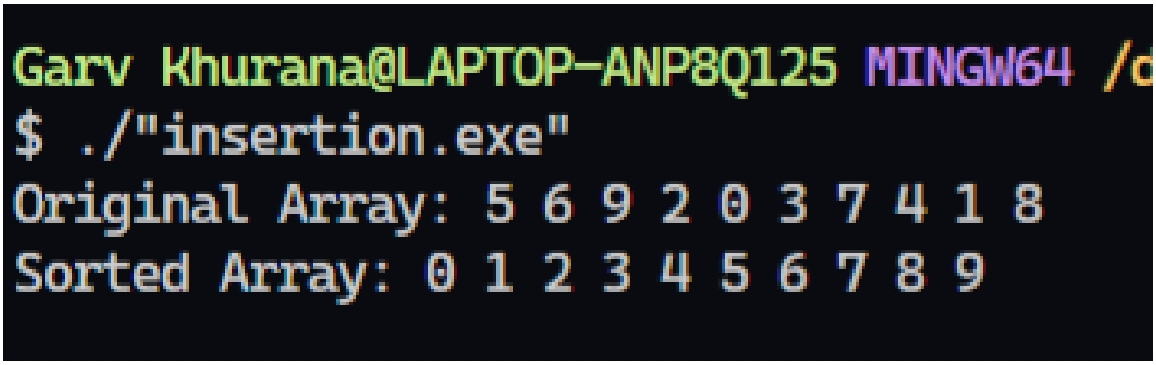
Program 2

1. **Aim/Overview of the practical:** WAP to sort an array using Insertion sort.
2. **Code:**

```
#include <stdio.h>

void insertion(int *arr, int size){

    int i, j, key;

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

        j = i - 1;

        key = arr[i];

        while ((j >= 0) && (key < arr[j])){

            arr[j + 1] = arr[j];

            j--;

        }

        arr[j + 1] = key;

    }

}

int main(){

    int arr[10] = {5, 6, 9, 2, 0, 3, 7, 4, 1, 8};

    printf("Original Array: ");

    for (int i = 0; i < 10; i++) {

        printf("%d ", arr[i]);

    }

    printf("\n");

    insertion(arr, 10);
```

```
    printf("Sorted Array: ");
    for (int i = 0; i < 10; i++){
        printf("%d ", arr[i]);
    }
    printf("\n");
}
```

3. Code Output:



Program 3

1. **Aim/Overview of the practical:** WAP to sort an array using Selection Sort
2. **Code:**

```
#include <stdio.h>

void selection(int *arr, int size){
    for (int i = 0; i < size; i++){
        for (int j = i + 1; j < size; j++){
            if (arr[i] > arr[j]) {
                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
}
int main(){
    int arr[10] = {40, 30, 50, 60, 90, 80, 10, 70, 20, 0};
    printf("Original Array: ");
    for (int i = 0; i < 10; i++){
        printf("%d ", arr[i]);
    }
    printf("\n");
    selection(arr, 10);
    printf("Sorted Array: ");
    for (int i = 0; i < 10; i++){
```

printf("%d ", arr[i]);

3. Code Output:

```
Garv Khurana@LAPTOP-ANP8Q125 MINGW64 /d/Chandigarh
$ ./"selection.exe"
Original Array: 40 30 50 60 90 80 10 70 20 0
Sorted Array: 0 10 20 30 40 50 60 70 80 90
```

Learning outcomes (What I have learned):

1. How to remove elements from an array
2. How to sort an array using Insertion Sort
3. How to sort and array using Quick Sort

Evaluation Grid (To be created per the faculty's SOP and Assessment guideline):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			