Question –

1. Write a menu-driven program in C to perform following operations in an integer array allocated memory dynamically. The list may grow or shrink compared to the initial allocation as and when required. Multiple operations may be performed on the existing list without recompiling/re-executing the program. The individual operation should be implemented using a single function.

List of operations:

a) Create,

b) Count,

c) Indexed element,

d) Insert,

e) Delete,

f) Merge,

g) Split,

h) Sort,

i) Search

Program –

```c
#include<stdio.h>
#include<stdlib.h>
#include<stdbool.h>
int *ptr;
int len = 0;
void swap(int *a,int *b){
```

```
    int temp = *a;
    *a = *b;
    *b = temp;
}
void create(int size){
    ptr = (int*)malloc(size * sizeof(int));
    for(int i=0;i<size;i++){
        scanf("%d",&ptr[i]);
    }
    len = size;
    printf("Enter the next operation.");
}
void insert(int n,int ind){
    for(int i=len;i>ind;i--){
        ptr[i] = ptr[i-1];
    }
    ptr[ind] = n;
    len++;
    printf("Enter the next operation.");
}
int count(){
    return len;
}
void indexed_element(int ind){
    printf("The %dth index is: ",ind);
    printf("%d\n",ptr[ind]);
    printf("Enter the next operation.");
}
void delete(int ind){
    for(int i=ind;i<len-1;i++){
        ptr[i] = ptr[i+1];
    }
    len--;
    printf("Enter the next operation.");
}
void sort(){
    int i,j;
    bool swapped;
    for(i = 0;i < len - 1; i++){
        for(j=0;j<len-i-1;j++){
            if(ptr[j] > ptr[j + 1]){
                swap(&ptr[j],&ptr[j + 1]);
            }
        }
    }
}
void search(int x){
    for(int i=0;i<len;i++){
```

```
        if(ptr[i]==x){
            printf("Element found at index %d",i);
        }
    }
}
void merge(){
    int n,i;
    printf("Enter the number of element for the 2nd array\n");
    scanf("%d",&n);
    int arr[n];
    int arr2[len + n];
    printf("Enter the elements\n");
    for(i=0;i<n;i++){
        scanf("%d",&arr[i]);
    }
    for(i=0;i<(len+n);i++){
        if(i < len){
            arr2[i] = ptr[i];
        }
        else{
            arr2[i] = arr[i - len];
        }
    }
    printf("The merged array is \n");
    for(i=0;i<(len + n);i++){
        printf("%d ",arr2[i]);
    }
    printf("\n");
    printf("Enter the next operation.");
}
void split(int ind){
    int i,arr1[ind + 1],arr2[len - (ind + 1)];
    for(i=0;i<ind + 1;i++){
        arr1[i] = ptr[i];
    }
    for(i=0;i<(len - (ind + 1));i++){
        arr2[i] = ptr[ind + 1 + i];
    }
    printf("After splitting, the 1st array is: \n");
    for(i=0;i<ind + 1;i++){
        printf("%d ",arr1[i]);
    }
    printf("\nThe 2nd array is: \n");
    for(i=0;i<(len - (ind + 1));i++){
        printf("%d ",arr2[i]);
    }
}
void main(){
```

```
    printf("Enter the opertions...\n1 for creating the array\n2 for inserting
values\n3 for getting the number of lements in the array\n 4 for getting an el
ement\n5 to delete an element\n6 for sorting the array\n7 for displaying the w
hole array\n8 for searching a particular value in the array\n9 for merging two
 arrays.\n10 for splitting the array.\n");
    while(1){
        int n,x,ind;
        scanf("%d",&n);
        switch (n)
        {
        case 1:
            printf("enter the size of the array: ");
            scanf("%d",&x);
            create(x);
            break;
        case 2:
            printf("enter the value to be entered: \n");
            scanf("%d",&x);
            printf("Enter the index to be inserted: \n");
            scanf("%d",&ind);
            insert(x,ind);
            break;
        case 3:
            printf("The number of elements of the array is %d",count());
            break;
        case 4:
            printf("enter the index: ");
            scanf("%d",&x);
            indexed_element(x);
            break;
        case 5:
            printf("Enter the index to be deleted: \n");
            scanf("%d",&ind);
            delete(ind);
            printf("element deleted...");
            break;
        case 6:
            sort();
            printf("element sorted...");
            break;
        case 7:
            printf("The elements are...");
            for(int i=0;i<len;i++){
                printf("%d ",ptr[i]);
            }
            break;
        case 8:
            printf("Enter the element to be searched: ");
```

```
            scanf("%d",&x);
            search(x);
            break;
        case 9:
            merge();
            break;
        case 10:
            printf("Enter the index for the split operation: ");
            scanf("%d",&ind);
            split(ind);
            break;
        default:
            exit(0);
        }
    }
}
```

Ouput –

```
Enter the opertions...
1 for creating the array
2 for inserting values
3 for getting the number of lements in the array
 4 for getting an element
5 to delete an element
6 for sorting the array
7 for displaying the whole array
8 for searching a particular value in the array
9 for merging two arrays.
10 for splitting the array.
1
enter the size of the array: 5
1
2
3
4
5
Enter the next operation.2
enter the value to be entered:
10
Enter the index to be inserted:
2
Enter the next operation.7
The elements are...1 2 10 3 4 5
3
The number of elements of the array is 6
4
enter the index: 2
The 2th index is: 10
Enter the next operation.5
```

```
The 2th index is: 10
Enter the next operation.5
Enter the index to be deleted:
2
Enter the next operation.element deleted...
7
The elements are...1 2 3 4 5
6
element sorted...
7
The elements are...1 2 3 4 5
8
Enter the element to be searched: 5
Element found at index 4
9
Enter the number of element for the 2nd array
6
Enter the elements
10
11
12
13
14
15
The merged array is
1 2 3 4 5 10 11 12 13 14 15
Enter the next operation.10
Enter the index for the split operation: 2
After splitting, the 1st array is:
1 2 3
The 2nd array is:
4 5
```