a) Write a c program to implement selection Sort & Bubble sort

selection sort

```
1. #include <stdio.h>
2. void selection(int arr[], int n)
3. {
4.     int i, j, small;
5.
6.     for (i = 0; i < n-1; i++)    // One by one move boundary of unso
   rted subarray
7.     {
8.         small = i; //minimum element in unsorted array
9.
10.             for (j = i+1; j < n; j++)
11.             if (arr[j] < arr[small])
12.                 small = j;
13.       // Swap the minimum element with the first element
14.          int temp = arr[small];
15.          arr[small] = arr[i];
16.          arr[i] = temp;
17.          }
18.       }
19.
20.       void printArr(int a[], int n) /* function to print the array */
21.       {  din
22.          int i;
23.          for (i = 0; i < n; i++)
24.              printf("%d ", a[i]);
25.       }
26.
27.       int main()
28.       {
29.          int a[] = { 12, 31, 25, 8, 32, 17 };
30.          int n = sizeof(a) / sizeof(a[0]);
```

```
31.         printf("Before sorting array elements are - \n");
32.         printArr(a, n);
33.         selection(a, n);
34.         printf("\nAfter sorting array elements are - \n");
35.         printArr(a, n);
36.         return 0;
        }
```

Bubble sort

```
1. #include<stdio.h>
2.  void print(int a[], int n) //function to print array elements
3.     {
4.     int i;
5.     for(i = 0; i < n; i++)
6.     {
7.        printf("%d ",a[i]);
8.     }
9.     }
10.       void bubble(int a[], int n) // function to implement bubble s
   ort
11.       {
12.         int i, j, temp;
13.         for(i = 0; i < n; i++)
14.          {
15.           for(j = i+1; j < n; j++)
16.             {
17.                if(a[j] < a[i])
18.                {
19.                   temp = a[i];
20.                   a[i] = a[j];
21.                   a[j] = temp;
22.                }
23.             }
24.          }
```

```
25.       }
26.       void main ()
27.       {
28.          int i, j,temp;
29.          int a[5] = { 10, 35, 32, 13, 26};
30.          int n = sizeof(a)/sizeof(a[0]);
31.          printf("Before sorting array elements are - \n");
32.          print(a, n);
33.          bubble(a, n);
34.          printf("\nAfter sorting array elements are - \n");
35.          print(a, n);
    }
```

b) Write a C program to reverse the elements within a given range in a sorted list.

Example :
input : 10
c 9 1 2 4 3 4 6 7 8 10
3 8
output: 1 2 8 7 6 4 4 3 9 10

```
#include <stdio.h>

// Function to reverse elements within a given range in a sorted list

void reverseRange(int arr[], int start, int end) {
    while (start < end) {
        int temp = arr[start];
        arr[start] = arr[end];
        arr[end] = temp;
        start++;
        end--;
    }
```

```
}

// Function to display array elements
void displayArray(int arr[], int n) {
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}
int main() {
    int n;
    printf("Enter the number of elements in the list: ");
    scanf("%d", &n);
    char type;
    int arr[n];
    printf("Enter the elements of the list in sorted order: ");
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    printf("Enter the range (start and end indices) to reverse: ");
    int start, end;
    scanf("%d %d", &start, &end);
    // Check if the range is valid
    if (start >= 0 && start < n && end >= 0 && end < n && start
<= end) {
```

```
        reverseRange(arr, start, end);
        printf("Output: ");
        displayArray(arr, n);
    } else {
        printf("Invalid range.\n");
    }
    return 0;
}
```