Exp - 3

Q1) Sort elements is in ascending order using bubble sort.

Ans:

```
#include <stdio.h>
int main()
{
    int i,j,n;
    int temp=0;
   printf("Enter number of array elements: ");
   scanf("%d", &n);

   int arr[n];
  printf("Enter array elements: \n");

  for (i=0; i<n; i++)
  {
     printf("Enter element %d: ", i);
     scanf("%d", &arr[i]);
  }
   for(i=0; i<n; i++)
  {
     for (j=0; j<n-1; j++)
     {
       if (arr[j] < arr[j+1])
      {
          temp = arr[j];
          arr[j]= arr[j+1];
          arr[j+1] = temp;
      }
      }
      }
printf("Array sorted in ascending order:\n");
for ( i = 0 ; i<n ; i++)
 {
    printf("\n %d", arr[i]);
  }
  return 0;
 }
```

Output:-

```
Enter number of array elements :5
 Enter array elements:
  Enter element 0 : 10
 Enter element 1 : 5
 Enter element 2 : 0
 Enter element 3 : 200
 Enter element 4 : 42
Array sorted in ascending order:
 0
 5
 10
 42
 200
```

Q 2) Sort elements in descending order using selection sort.

Ans

```
#include<stdio.h>
int main()
{
    int i, j, n, temp = 0;
   printf("Enter number of array elements:");
   scanf("%d", &n);
   int arr[n];
printf("Enter array elements:\n");
for (i=0; i<n; i++)
{
    printf("Enter elements:&%d:", i);
    scanf("%d",&arr[i]);
}
for (i=0; i<n; i++)
{
    for (i=i+1; i<n; i++)
    {
        if (a[i] <a[i])
        {
            r= a[i];
            a[i] = a[i];
            a[i] = r;
        }
    }
}
printf("\n Sorted Array Element in descending order :\n");
for (i=0; i<n; i++)
{
    printf("%d\n", a[i]);
}
return 0;
}
```

* Output

```
Enter no. of elements in array : 5
Enter array elements:
1
2
3
4
5
```

Sorted array element in descending order

5

4

3

2

1

3)

Bubble sort:

S:- 100, 200, 300, 400, 500

we know no. of pass = n(-1) (n=5)

$\underbrace{\boxed{100}\quad\boxed{200}}_{x}\quad 300\quad 400\quad 500$

$100\quad\underbrace{\boxed{200}\quad\boxed{300}}_{x}\quad 400\quad 500$

$100\quad 200\quad\underbrace{\boxed{300}\quad\boxed{400}}_{x}\quad 500$

$100\quad 200\quad 300\quad\underbrace{\boxed{400}\quad\boxed{500}}_{x}$

* Algorithm.

1) Start with the list of no.

2) Compare each element and swap if the first element is greater ~~to~~ than second element.

3) After each full pass the largest unsorted element moves to the array.

4) Repeat the process of 2 step on the unsorted Array.

5) Stop the program.

4) Selection sort.

Ans. 5 elements:- 500, -20, 30, 14, 50

$\boxed{500}$ $\boxed{-20}$ 30 14 50

x

$\boxed{-20}$ 500 $\boxed{30}$ 14 50

x

-20 ~~500~~ 500 30 $\boxed{14}$ 50

x

$\boxed{-20}$ 500 30 14 $\boxed{50}$

x

Pass. 2) -20 $\boxed{500}$ $\boxed{30}$ 14 50

x

-20 $\boxed{30}$ 500 $\boxed{14}$ 50

x

-20 $\boxed{14}$ 500 30 $\boxed{50}$

x

Pass- 3) -20 14 $\boxed{500}$ $\boxed{30}$ 50

x

-20 14 $\boxed{30}$ 500 $\boxed{50}$

x

Pass-4) -20 14 30 $\boxed{500}$ $\boxed{50}$

x

-20 14 30 50 500

* Algorithm.

1) Start
2) Search min element of array.
3) Swap value of the selection of ~~a~~ main
4) Increment min to print the next element location
5) Repeat . till sorted
6) Stop

13/11