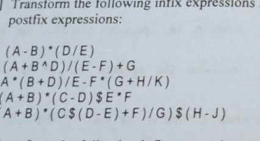
Q1.



Ans 1)

AB-DE/\*

AB+D^EF-G+/

ABD+\*EF-GH+K/\*/

AB+CD-E$F\*\*

AB+CDE-$F+G/HJ-$\*

Q2.Sort the array using selection sort

arr[] = 64 25 12 22 11

Ans2)

First we will find the minimum in the array from arr[0] to arr[4] and replace it with the starting of this array that is on 1st element

11 25 12 22 64

Then find the minimum in the array from arr[1] to arr[4] and replace it with the starting of this array that is on 2st element

11 12 25 22 64

Then find the minimum in the array from arr[2] to arr[4] and replace it with the starting of this array that is on 3st element

11 12 22 25 64

The array is sorted.

Q3. Sort the following array using selection sort algorithm



Ans3)

Arr[]= 4 3 2 10 12 1 5 6

First we will find the minimum in the array from arr[0] to arr[7] and replace it with the starting of this array that is on 1st element

1 3 2 10 12 4 5 6

Then find the minimum in the array from arr[1] to arr[7] and replace it with the starting of this array that is on 2st element

1 2 3 10 12 4 5 6

Then find the minimum in the array from arr[2] to arr[7] and replace it with the starting of this array that is on 3st element

123 10 12 4 5 6

Then find the minimum in the array from arr[3] to arr[7] and replace it with the starting of this array that is on 4st element

1 2 3 4 12 10 5 6

Then find the minimum in the array from arr[4] to arr[7] and replace it with the starting of this array that is on 5st element

1 2 3 4 5 10 12 6

Then find the minimum in the array from arr[5] to arr[7] and replace it with the starting of this array that is on 6st element

1 2 3 4 5 6 12 10

Then find the minimum in the array from arr[6] to arr[7] and replace it with the starting of this array that is on 7st element

1 2 3 4 5 6 10 12

Q4.Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in wrong order.

Sort this array using bubble sort demonstrate all the steps 5 1 4 2 8.

Ans4)

*First Pass:*  
( *5 1* 4 2 8 ) –> ( **1***5* 4 2 8 ) compares the first two elements, swaps 5 and 1.  
( 1 *5 4* 2 8 ) –>  ( 1 *4 5* 2 8 ), Swaps 5 and 4  
( 1 4 *5 2* 8 ) –>  ( 1 4 *2 5* 8 ), Swaps 5 and 2  
( 1 4 2 *5 8* ) –> ( 1 4 2 *5 8* ),

*Second Pass:*  
( *1 4* 2 5 8 ) –> ( *1 4* 2 5 8 )  
( 1 *4 2* 5 8 ) –> ( 1 *2 4* 5 8 ), Swaps 4 and 2  
( 1 2 *4 5* 8 ) –> ( 1 2 *4 5* 8 )  
( 1 2 4 *5 8* ) –>  ( 1 2 4 *5 8* )

array is sorted, but the algorithm needs one whole pass without any swap to know it is sorted.

*Third Pass:*

( 1 2 4 5 8 ) –> ( 1 2 4 5 8 )  
( 1 2 4 5 8 ) –> ( 1 2 4 5 8 )  
( 1 2 4 5 8 ) –> ( 1 2 4 5 8 )  
( 1 2 4 5 8 ) –> ( 1 2 4 5 8 )