Data Structure and Algorithm Essentials

DAY 5 | ASSIGNMENT **EMAIL:** pavanibhavya77@gmail.com

1.)Write the function for insertion sort.
Sol.)

```
#include<stdio.h>
void main()
{
int i, j, num, temp, arr[20];
printf("Enter size of array: ");
scanf("%d", &num);
printf("Enter %d elements in array: \n", num);
for(i=0; i<num; i++)
{
scanf("%d", &arr[i]);
}
for(i=1; i<num; i++) {
temp=arr[i];
j=i-1;
while((temp<arr[j])&&(j>=0))
{
arr[j+1]=arr[j];
j=j-1;
}
arr[j+1]=temp;
}
printf("After Sorting elements: ");
for(i=0; i<num; i++)
{
printf("%d ", arr[i]);
}
}
```

2.)Write a function to find the maximum element in the stack.
Sol.)

```
#include<stdio.h>
#define SIZE 20
void push(int);
void display_max();
int stack[SIZE],top=-1;
int main(){
```

```
    int value,ch;
    while(1){
        printf("1.push\n2.maximum element\n");
        printf("enter choice");
        scanf("%d",&ch);
        switch(ch){
            case 1:printf("enter num to be pushed\n");
            scanf("%d",&value);
            push(value);
            break;
            case 2:display_max();
            break;
            default:printf("wrong choice\n");
        }
    }
    return 0;
}
void push(int value){
    if(top==SIZE-1)
    {
        printf("\nStack is full!!");

    }
    else
    {
        top=top+1;
        stack[top]=value;
    }
}
void display_max(){
    int i,j,temp;
    for(i=1; i<=top; i++) {
    temp=stack[i];
    j=i-1;
    while((temp>stack[j])&&(j>=0))
    {
        stack[j+1]=stack[j];
        j=j-1;
    }
    stack[j+1]=temp;
}
printf("maximum element is %d\n",stack[0]);
}
```

3.)Write a function to find the minimum element in the stack.
Sol.)

```
#include<stdio.h>
#define SIZE 20
void push(int);
void display_min();
int stack[SIZE],top=-1;
int main(){
   int value,ch;
   while(1){
      printf("1.push\n2.minimum element\n");
      printf("enter choice");
      scanf("%d",&ch);
      switch(ch){
         case 1:printf("enter num to be pushed\n");
         scanf("%d",&value);
         push(value);
         break;
         case 2:display_min();
         break;
         default:printf("wrong choice\n");
      }
   }
   return 0;
}
void push(int value){
   if(top==SIZE-1)
   {
      printf("\nStack is full!!");

   }
   else
   {
      top=top+1;
      stack[top]=value;
   }
}
void display_min(){
   int i,j,temp;
   for(i=1; i<=top; i++) {
   temp=stack[i];
   j=i-1;
   while((temp<stack[j])&&(j>=0))
   {
```

```
        stack[j+1]=stack[j];
        j=j-1;
    }
    stack[j+1]=temp;
}
printf("minimum element is %d\n",stack[0]);
}
```