STACK OPERATIONS

CODE:

```
#include<stdio.h>
#include<conio.h>
#define max 5
int top=-1;
int st[max],i,ele;
void push(int st[],int ele);
int pop();
void display();
void main()
{
 int ch,i,n;

 do
 {
 printf("\t\n***menu***\n1.push\n2.pop\n3.display\n4.exit\n");
 printf("enter your choice");
 scanf("%d",&ch);
 switch(ch)
 {
 case 1:
  {
     printf("enter the elements\t");
     scanf("%d",&n);
```

```
        push(st,n);
        break;
    }
  case 2:
    {
        n=pop();
        break;
    }
  case 3:
        display();
        break;
    case 4:
        exit(0);
        break;
  }
  }
  while(ch!=5);
  getch();
}
void push(int st[],int ele)
{
    if(top==max-1)
    {
        printf("\t stack is full");
    }
    else
```

```
        top++;
        st[top]=ele;
}
int pop()
{
    int ele;
    if(top==-1)
    {
        printf("\t stack is empty");
    }
    else
        {
            ele=st[top];
            top--;
            printf("element deleted is %d",ele);
            return ele;

        }
}
void display()
{
    int i;
    if(top==-1)
    {
        printf("\t stack is empty");
    }
```

```
    else
        {
            for(i=top;i>=0;i--)
            {
                printf("%d\t",st[i]);


            }


        }


}
```

OUTPUT:


menu
1.push
2.pop
3.display
4.exit
enter your choice1
enter the elements 3
menu
1.push
2.pop
3.display
4.exit
enter your choice2
element deleted is 3
menu
1.push
2.pop
3.display
4.exit
enter your choice3
stack is empty
menu
1.push
2.pop
3.display
4.exit
enter your choice4
Process returned 0 (0x0) execution time : 25.107 s
Press any key to continue.