ASSIGNMENT NO:4 **DATE:24/02/2016**

PROGRAM TITLE:Implement a stack.

THEORY:

We implement a stack using an array. Thus the max size of the stack is already predefined and cannot be changed after the start of the program.

ALGORITHM:

```
Algo_push(item)
{
      if(top<maxsize)
      {
            top=top+1;
            stack[top]=item;
      }
      else
            print "Overflow"
}

Algo-pop()
{
      if(top>-1)
      {
            delete stack[top];
            top=top-1;
      }
      else
            underflow;
}
```

PROGRAM CODE:

```
#Shell Program to implement Stack Operations
read -p "Enter max size of stack::" m
top=-1
push()
{
      if [ $top -ne `expr $m - 1` ]
      then
            top=`expr $top + 1`
            stack[$top]=$1
      else
            echo "The Stack is Full."
      fi
}
pop()
{
      if [ $top -ne -1 ]
      then
            echo "The element deleted is::" ${stack[$top]}
            top=`expr $top - 1`
      else
            echo "The Stack is Empty."
      fi
```

```
}
disp()
{
      echo "The Stack is now"
      if [ $top -eq -1 ]
      then
            echo "empty."
      else
            i=0
            while [ $i -le $top ]
            do
                  echo ${stack[$i]}
                  i=`expr $i + 1`
            done
      fi
}
ch=1
while [ $ch -ne 3 ]
do
      read -p "Menu::1.Push 2.Pop 3.Exit.Enter your Choice::" ch
      case $ch in
            1) read -p "Enter the number::" n
                  push $n
                  ;;
            2) pop
                  ;;
            3) echo "Program terminated."
      esac
      disp
done
```

OUTPUT:

```
Enter max size of stack::5
Menu::1.Push 2.Pop 3.Exit.Enter your Choice::1
Enter the number::10
The Stack is now
10
Menu::1.Push 2.Pop 3.Exit.Enter your Choice::1
Enter the number::20
The Stack is now
10
20
Menu::1.Push 2.Pop 3.Exit.Enter your Choice::1
Enter the number::30
The Stack is now
10
20
30
Menu::1.Push 2.Pop 3.Exit.Enter your Choice::2
The element deleted is:: 30
The Stack is now
10
20
Menu::1.Push 2.Pop 3.Exit.Enter your Choice::1
Enter the number::40
The Stack is now
```

```
10
20
40
Menu::1.Push 2.Pop 3.Exit.Enter your Choice::1
Enter the number::50
The Stack is now
10
20
40
50
Menu::1.Push 2.Pop 3.Exit.Enter your Choice::1
Enter the number::60
The Stack is now
10
20
40
50
60
Menu::1.Push 2.Pop 3.Exit.Enter your Choice::1
Enter the number::70
The Stack is Full.
The Stack is now
10
20
40
50
60
Menu::1.Push 2.Pop 3.Exit.Enter your Choice::2
The element deleted is:: 60
The Stack is now
10
20
40
50
Menu::1.Push 2.Pop 3.Exit.Enter your Choice::2
The element deleted is:: 50
The Stack is now
10
20
40
Menu::1.Push 2.Pop 3.Exit.Enter your Choice::2
The element deleted is:: 40
The Stack is now
10
20
Menu::1.Push 2.Pop 3.Exit.Enter your Choice::2
The element deleted is:: 20
The Stack is now
10
Menu::1.Push 2.Pop 3.Exit.Enter your Choice::2
The element deleted is:: 10
The Stack is now
empty.
Menu::1.Push 2.Pop 3.Exit.Enter your Choice::3
Program terminated.
The Stack is now
empty.
```

DISCUSSION:

1. We are using commandline arguments to send data to the functions.
2. The same set of variables are being used throughout the program.
3. No size for the array stack need be pre-defined. We have to ensure that the size doesn't exceed the specified size by using another variable 'max'.