

```
##### PROGRAM TO EVALUATE EXPRESSION USING STACK #####
```

```
class Evaluate:
```

```
    def __init__(self,capacity):
        self.array=[]
        self.top=-1
        self.capacity=capacity

    def isempty(self):
        return len(self.array)==0

    def isfull(self):
        return len(self.array)==self.capacity

    def push(self,ele):
        if self.isfull():
            print("Stack is Full")
        else:
            self.array.append(ele)
            print("element pushed into the stack",ele)

    def pop(self):
        if self.isempty():
            print("Stack is Empty")
        else:
            ele=self.array.pop()
            print("element popped from the stack",ele)
            return ele

    def evaluatePostfix(self,exp):
        for i in exp:
            if i.isdigit():
                self.push(i)
            else:
                val1=self.pop()
                val2=self.pop()
                self.push(str(eval(val2 + i + val1)))
        return int(self.pop())
```

```
exp="231*+9-"
print("expression to evaluate is =",exp)
obj=Evaluate(len(exp))
result=obj.evaluatePostfix(exp)
print("value of {0} is {1} ".format(exp,result))
```

```
#####OUTPUT#####3
...
```

```
expression to evaluate is = 231*+9-
element pushed into the stack 2
element pushed into the stack 3
element pushed into the stack 1
element popped from the stack 1
```

```
element popped from the stack 3
element pushed into the stack 3
element popped from the stack 3
element popped from the stack 2
element pushed into the stack 5
element pushed into the stack 9
element popped from the stack 9
element popped from the stack 5
element pushed into the stack -4
element popped from the stack -4
value of 231*+9- is -4
'''
```