Day 6

Evaluation of postfin Expression

Posttin notation is used to represent algebraic expression. The expressio written in postfin from are evaluated faster Compared to infin notation as prarthsis are not required in postfin.

Evalulation 1:- (Single digit Expression)

```
Class Evaluate :

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

    def . isempty(self):
        return True if self.top == -1
                    else False
```

```
def peek(self):
    return self.stack[-1]

def pop(self):
    If not self.isempty():
        self.top -= 1
        return self.stack.pop()
    else:
        return "$"

def push(self, i):
    self.top += 1
    self.stack.append(i)

def evalpostfin(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())
```

input :-

exp = "231*+9-"

obj = Evaluate (len(exp))

print (obj. Evalposttfin (exp))

output :

-4

Expression 3

Indeger Contains with multiple digits.

```
class evaluate :
    def __init__ (self):
        self. top = -1
        self. stack = []
    def isempty (self)
        return true if self. top == -1
                    else False
    def peek(self):
        return self. stack [-1]
```

```
def pop(self):
    if not self.isempty():
        self.top -= 1
        return self.stack.pop()
    else:
        return "$"

def push(self, i):
    self.top += 1
    self.stack.append(i)

def evalpostfix(self, exp):
    for i in exp:
        try:
            self.push(int(i))
        except ValueError:
            val1 = self.pop()
            val2 = self.pop()
            switcher = {'+': val2 + val1,
                        '-': val2 - val1,
                        '*': val2 * val1,
                        '/': val2 / val1,
                        '^': val2 ** val1}
```

```
        self.push(switcher.get(i))
    return int(self.pop())
```

Input :

```
Str = "100 200 + 2 / 5 * 7"

Strconv = Str.split(' ')

Obj = Evaluate()

print(Obj.Evaluatepostfin(Strconv))
```

Output :

757