In [1]:

```
# A
class ArrayStack:
    def __init__(self,size):
        self.size=size
        self.data=[0 for i in range(size)]
        self.top=0
    def isEmpty(self):
        if self.top==0:
            return True
        else:
            return False
    def Push(self,value):
        if self.top==self.size:
            print("Stack Overflow !")
        else:
            self.data[self.top]=value
            self.top+=1
    def Pop (self):
        if self.isEmpty():
            print("Stack Underflow !")
        else:
            x=self.data[self.top-1]
            self.top-=1
            self.data[self.top]=0
            return x
    def Check(self):
        if self.isEmpty:
            True
        else:
            False
    def Peek(self):
        return self.data[self.top-1]
    def Count(self):
        return self.top
    def Printt(self):
        for i in self.data:
            print(i)
ob=ArrayStack(3)
ob.Push(7)
ob.Push(6)
ob.Push(5)
ob.Push(77)
#ob.Pop()
print(ob.Peek())
print(ob.Count())
ob.Printt()
```

```
Stack Overflow !
5
3
7
6
5
```

In [4]:

```
# B
class ArrayQueue:
    def __init__(self,size):
        self.size=size
        self.data=[0 for i in range(size)]
        self.f=-1
        self.r=0
    def enQueue(self,value):
        self.data[self.r]=value
        self.r=(self.r+1)%self.size
    def deQueue(self):
        self.f=(self.f+1)%self.size
        return self.data[self.f]
    def isEmpty(self):
        if self.f==-1 and self.r==0:
            return True
        else:
            False
    def Count(self):
        return len(self.data)
    def Printt(self):
        print(self.Count())
ob=ArrayQueue(4)
ob.enQueue(1)
ob.enQueue(2)
ob.enQueue(3)
ob.enQueue(4)
ob.enQueue(5)
print(ob.deQueue())
print(ob.data)
ob.Printt()
```

```
5
2
[5, 77, 3, 4]
4
```

In [6]:

```
# C
class ArrayStack:
    def __init__(self,lst):
        self.lst=lst
        self.size=len(self.lst)
        self.data=[0 for i in range(self.size)]
        self.top=0
    def StringExp(self):
        for i in self.lst:
            #print(self.data)
            if i=="{" or i=="(" or i=="[":
                self.Push(i)
            if i=="}" or i==")" or i=="]":
                self.Pop()
        return self.isEmpty()
    def isEmpty(self):
        if self.top==0:
            return True
        else:
            return False
    def Push(self,value):
        if self.top==self.size:
            print("Stack Overflow !")
        else:
            self.data[self.top]=value
            self.top+=1
    def Pop(self):
        if self.isEmpty():
            print("Stack Underflow !")
        else:
            x=self.data[self.top]
            self.top-=1
            self.data[self.top]=0
            return x
str1="{()}[()]{}"
ob=ArrayStack(str1)
ob.Pop()
print(ob.StringExp())
```

```
Stack Underflow !
True
```