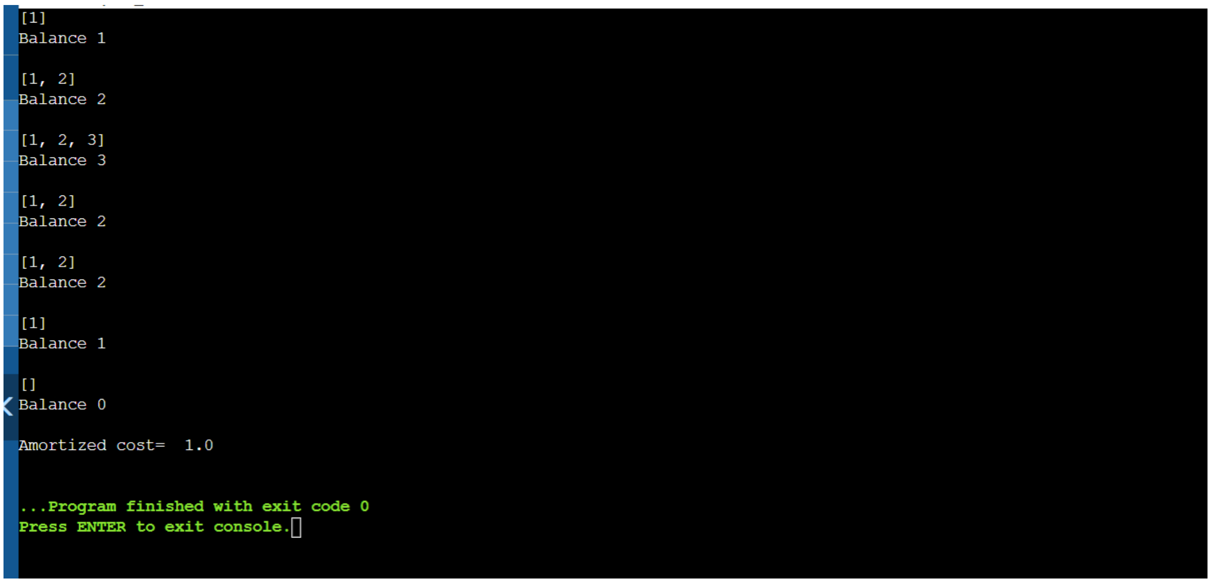
```
def accounting(n):
    size = 1
    total = 0
    dcost = 0
    icost = 0
    bank = 0

    print("Elements\tDoubling Copying Cost\tInsertion Cost\tTotal Cost\tBank\t\tSize")

    for i in range(1, n + 1):
        icost = 1
        if i > size:
            size *= 2
            dcost = i - 1
        total = icost + dcost
        bank += (3 - total)

        print(i, "\t\t\t", dcost, "\t\t\t", icost, "\t", total, "\t\t", bank, "\t\t", size)

        icost = 0
        dcost = 0

    n = int(input("Enter number of elements:"))
    print("Accounting method")
    accounting(n)


class AccountingStack:
    def __init__(self):
        self.stack = []
        self.cost = 0
        self.balance = 0

    def push(self, item):
        self.stack.append(item)
        self.cost += 1
        self.balance += 1
        self.printstack()

    def pop(self):
        self.stack.pop()
        self.cost += 1
        self.balance -= 1
        self.printstack()

    def multipop(self, k):
        for i in range(k):
            self.pop()

    def printstack(self):
        print(self.stack, "\nBalance", self.balance, "\n")


s = AccountingStack()
s.push(1)
s.push(2)
s.push(3)
s.pop()
s.printstack()
s.multipop(2)
print("Amortized cost= ", s.cost / 6)
```

Output :



```
[1]
Balance 1

[1, 2]
Balance 2

[1, 2, 3]
Balance 3

[1, 2]
Balance 2

[1, 2]
Balance 2

[1]
Balance 1

[]
Balance 0

Amortized cost=  1.0


...Program finished with exit code 0
Press ENTER to exit console.
```