

Session 3: Loop Logic (Solutions Only)

Case 4. Referral Marketing

Solve the discussion question using for or while loops. In each case, denote the numerical inputs (12, 100, and 100) by a variable so that your code would work with any other input. In particular, find the monthly sale in month 36, and number of months before monthly sale exceeds 1 million. **Hint:** Use a list to store the sales for each month.

```
[7]: # Solution to Case 4a)
n=36
sales=[1,1]
for month in range(2,n+1):
    curSale=sales[-1]+sales[-2]
    sales.append(curSale)
print(f'The number of items sold in month {n} is {sales[n]}')
```

The number of items sold in month 36 is 24157817.

```
[8]: # Illustration of the above
n=12
sales=[1,1]
print('month\tsales')
for month in range(2,n+1):
    curSale=sales[-1]+sales[-2]
    sales.append(curSale)
    print(f'{month}\t{sales}')
print(f'The number of items sold in month {n} is {sales[n]}')
```

month	sales
2	[1, 1, 2]
3	[1, 1, 2, 3]
4	[1, 1, 2, 3, 5]
5	[1, 1, 2, 3, 5, 8]
6	[1, 1, 2, 3, 5, 8, 13]
7	[1, 1, 2, 3, 5, 8, 13, 21]
8	[1, 1, 2, 3, 5, 8, 13, 21, 34]
9	[1, 1, 2, 3, 5, 8, 13, 21, 34, 55]
10	[1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
11	[1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144]
12	[1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233]

The number of items sold in month 12 is 233.

```
[9]: # Alternative solution to Case 4a) that only stores last two months
n=36
oldSale=0
sale=1
for month in range(n):
    newSale=sale+oldSale
    oldSale=sale
    sale=newSale
print(f'The number of items sold in month {n} is {sale}')
```

The number of items sold in month 36 is 24157817.

```
[10]: # Illustration of the above
n=12
oldSale=0
sale=1
print('month\tsale\toldSale')
print(f'0\t{sale}\t{oldSale}')
for month in range(n):
    newSale=sale+oldSale
    oldSale=sale
    sale=newSale
    print(f'{month+1}\t{sale}\t{oldSale}')
print(f'The number of items sold in month {n} is {sale}.')
```

month	sale	oldSale
0	1	0
1	1	1
2	2	1
3	3	2
4	5	3
5	8	5
6	13	8
7	21	13
8	34	21
9	55	34
10	89	55
11	144	89
12	233	144

The number of items sold in month 12 is 233.

```
[11]: # Solution to Case 4b)
      threshold=1000000
      sales=[1,1]
      while sales[-1]<=threshold:
          sales.append(sales[-1]+sales[-2])
      n=len(sales)-1
      print(f'In month {n}, the sales is {sales[n]}, which exceeds {threshold}.')
```

In month 30, the sales is 1346269, which exceeds 1000000.

```
[12]: # Illustration of the above
      threshold=100
      sales=[1,1]
      print('sales[-1]<=threshold?\tsales')
      while sales[-1]<=threshold:
          sales.append(sales[-1]+sales[-2])
          print(f'{sales[-1]<=threshold}\t\t\t{sales}')
      n=len(sales)-1
      print(f'In month {n}, the sales is {sales[n]}, which exceeds {threshold}.')
```

sales[-1]<=threshold?	sales
True	[1, 1, 2]
True	[1, 1, 2, 3]
True	[1, 1, 2, 3, 5]
True	[1, 1, 2, 3, 5, 8]
True	[1, 1, 2, 3, 5, 8, 13]
True	[1, 1, 2, 3, 5, 8, 13, 21]
True	[1, 1, 2, 3, 5, 8, 13, 21, 34]
True	[1, 1, 2, 3, 5, 8, 13, 21, 34, 55]
True	[1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
False	[1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144]

In month 11, the sales is 144, which exceeds 100.

```
[13]: # Alternative Solution to Case 4b) that only stores last two months
threshold=100
oldSale=0
n=0
sale=1
while sale<=threshold:
    newSale=sale+oldSale
    oldSale=sale
    sale=newSale
    n+=1
print(f'In month {n}, the sales is {sale}, which exceeds {threshold}.')
```

In month 11, the sales is 144, which exceeds 100.

```
[14]: # Illustration of the above
threshold=100
oldSale=0
n=0
sale=1
print('month\tsale\toldSale\tsale<=threshold?')
while sale<=threshold:
    newSale=sale+oldSale
    oldSale=sale
    sale=newSale
    n+=1
    print(f'{n}\t{sale}\t{oldSale}\t{sale<=threshold?}')
print(f'In month {n}, the sales is {sale}, which exceeds {threshold}.')
```

month	sale	oldSale	sale<=threshold?
1	1	1	True
2	2	1	True
3	3	2	True
4	5	3	True
5	8	5	True
6	13	8	True
7	21	13	True
8	34	21	True
9	55	34	True
10	89	55	True
11	144	89	False

In month 11, the sales is 144, which exceeds 100.

Case 5. Hospital Capacity Planning

A hospital is applying for funding to expand its number of beds. Given a list of estimated number of arrivals each day, and assuming each patient stays 3 days, how many beds does the hospital need so as to be able to not turn away anyone?

```
demand=[5,8,3,10,7,4,9,5,8]
```

```
[16]: demand=[5,8,3,10,7,4,9,5,8]
      worstNeed=0
      for day in range(2,len(demand)):
          recentDemand=demand[day-2:day+1]
          need=sum(recentDemand)
          if need>worstNeed:
              worstNeed=need
      print(f'The number of beds needed to accommodate demand is {worstNeed}.')
```

The number of beds needed to accommodate demand is 22.

Here's the same code with intermediate outputs to show what is going on each iteration.

```
[15]: demand=[5,8,3,10,7,4,9,5,8]
      print('day\tarrivals last 3 days\tneed\t{worstNeed}')
      worstNeed=0
      for day in range(2,len(demand)):
          recentDemand=demand[day-2:day+1]
          need=sum(recentDemand)
          if need>worstNeed:
              worstNeed=need
          print(f'{day}\t{recentDemand}\t\t{need}\t{worstNeed}')
      print(f'The number of beds needed to accommodate demand is {worstNeed}.')
```

day	arrivals last 3 days	need	{worstNeed}
2	[5, 8, 3]	16	16
3	[8, 3, 10]	21	21
4	[3, 10, 7]	20	21
5	[10, 7, 4]	21	21
6	[7, 4, 9]	20	21
7	[4, 9, 5]	18	21
8	[9, 5, 8]	22	22

The number of beds needed to accommodate demand is 22.