

- (1) Use a for-loop to compute the sum of all numbers from 1 to 33 (both included) and print the result to the screen.

Solution:

```
total = 0
for i in range(1,34):
    total += i
print('The total is', total)
```

- (2) Use a while loop to add up the numbers 1,2,3,... until the sum has become larger than 100. Print the resulting sum and the last number added to the screen.

Solution:

```
total = 0
count = 0
while total <= 100:
    count += 1
    total += count
print('total =', total, ' , last added number =', count)
```

(3) Write a program that uses a for-loop to simulate 20 coin tosses by using random numbers between 1 and 2. Random numbers can be generated using the randint(...) function, e.g. randint(1,2) to generate a random integer number between 1 and 2 (both included). To use the randint(...) function, the statement

```
from random import randint
```

is needed at the beginning of the program. Within the loop, print out "Head" if the number is 1 and "Tail" if the number is 2. In addition, count the number of heads and tails and print out the percentages of heads and tails after the loop, e.g. as

Percentage heads: ...%

Percentage tails: ...%

You will have to make sure that the computation of the percentages is done using floating point division (not floor division), for instance by using an float value for the total number of throws.

Solution:

```
from random import randint
```

```
countHeads = 0
```

```
countTails = 0
```

```
count = 20
```

```
for i in range(1,count+1):
```

```
    r = randint(1,2)
```

```
    if r == 1:
```

```
        countHeads += 1
```

```
        print('Head')
```

```
    else:
```

```
        countTails += 1
```

```
        print('Tail')
```

```
print('Percentage heads:', (countHeads / count * 100), '%')
```

```
print('Percentage tails:', (countTails / count * 100), '%')
```

(4) Write a program that goes through a list of exam scores given as integer numbers between 0 and 100. Scores from 90 to 100 correspond to an A, from 80 to 89 to a B, 70 to 79 to C, 60 to 69 to D, and 0 to 59 to F. Negative numbers or numbers above 100 in the list should be ignored. The goal is to print the total number of valid scores (= scores between 0 and 100) and the number of scores in each letter-grade category (A,B,C,D, and F) to the screen.

Example: For the list

```
scores = [98, 87, 86, 85, 85, 78, 73, 72, 72, 72, 70, 66, 63, 50]
```

the output should be

Total number of scores: 14

Number of A's = 1

Number of B's = 4

Number of C's = 6

Number of D's = 2

Number of F's = 1

Solution:

```
scores = [98, 87, 86, 85, 85, 78, 73, 72, 72, 72, 70, 66, 63, 50]
countA = 0
countB = 0
countC = 0
countD = 0
countF = 0
validScores = 0
```

```
for number in scores:
    if number >= 0 and number <= 100:
        validScores += 1

        if number >= 90 and number <= 100:
            countA += 1
        elif number >=80 and number < 90:
            countB +=1
        elif number >= 70 and number < 80:
            countC += 1
        elif number >= 60 and number < 70:
            countD += 1
        elif number >= 0 and number < 60:
            countF += 1
```

```
print('Total number of valid scores:',validScores)
print("Number of A's =",countA)
print("Number of B's =",countB)
print("Number of C's =",countC)
print("Number of D's =",countD)
print("Number of F's =",countF)
```

(5) Write a program which takes the integer number stored in variable `n`, let's say `n=5`, and then generates a triangle of numbers as shown below:

```
1
12
123
1234
12345
```

The first line always contains just the number 1, while the last line contains the numbers from 1 to `n`. One solution to this problem uses two nested for-loops. You can use `print` with a with the additional parameter `end=' '` (e.g., `print(x, end=' ')`) to avoid the line break after the output. `print('\n')` on the other hand can be used to generate just a line break.

Solution:

```
n = 5
for i in range(1,n+1):
    for j in range(1,i+1):
        print(j, end=' ')
    print('\n')
```