1) Find the ***running time* T(n)** and the asymptotic running time (using Θ-notation and O-notation) of the following piece of code:

```
n = int(input("Enter an integer number greater than 2:"))

for i in range(n):
     print(i)

for j in range(n):
     print(j)
```

T(n) =

T(n) = Θ( )

2) Find the ***running time* T(n)** and the asymptotic running time (using Θ-notation and O-notation) of the following piece of code:

```
n = int(input("Enter an integer number greater than 10:"))

for i in range(n):
     for j in range(n):
          print(i,"\t",j)
```

T(n) =

T(n) = Θ( )

3) Find the ***running time* T(n)** and the asymptotic running time (using Θ-notation and O-notation) of the following piece of code:

```
n = int(input("Enter an integer number greater than 12:"))

while n>1:
     print(n)
     n=n//2

print(n)
```

T(n) =

T(n) = Θ( )

**CSI33** **Algorithm Analysis handout** **v1**

1) Copy the following program (you may omit the docstring):

```
def summation1(n):
    """ finds the sum (n+i)^2/i, where i runs from 1 to n

    pre: n in positive integer
    post: returns a positive integer number."""
    sum = 0
    for elem in list(range(n)):
        sum += (n+1+elem)**2/(elem+1)
    return sum
```

2) run the defined procedure on different inputs, for example *n = 1, 2, 10*.
Write down the results.

3) Write the sum of fractions that the program calculates on inputs *n = 1, 2, 10*
don't calculate it! keep them as fractions, but feel free to simplify!

4) find the ***running time*** **T(n)** of the procedure (depending on n), assuming that it takes one unit of time for each of math operations; the assignment operator and `range` function also take one time unit, and function `list` takes ***n*** time units.

5) What is the order of growth (in terms of O and Θ)?