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

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): niterations
 for j in range(n): niterations
 print(i,"\t",j) 1 step

 $T(n) = n^*n^*1 = n^2$ $T(n) = \Theta(n^2)$

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) 1 step
    n = n//2 each time n becomes "twice less", until eventually it is <= 1, 2 steps
print(n) 1 step</pre>
```

 $T(n) = 3\log_2 n + 1$ $T(n) = \Theta(\log n)$

CSI33 Algorithm Analysis handout

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

<u>pre</u>: 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
```

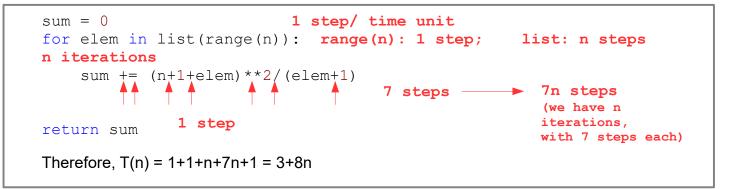
2) run the defined procedure on different inputs, for example n = 1, 2, 10. Write down the results.

4.0 17.0 547.8968253968254

3) Write, following the code of the program, each call of this procedure on inputs n = 1, 2, 10 as a sum of fractions, i.e. write which sum finds for procedure for each of these calls, but don't calculate it.

n=1 list: 0	n=2 list = 0,1	n=10 list = 0,1,2,3,4,5,6,7,8,9
2^{2}	$3^2 \cdot 4^2$	sum: $0 + \frac{11^2}{1} + \frac{12^2}{2} + \frac{13^2}{3} + \frac{14^2}{4} + \frac{15^2}{5} + \frac{16^2}{6} + \frac{17^2}{7} + \frac{18^2}{8} + \frac{19^2}{9} + \frac{20^2}{10}$
sum: $0 + \frac{2}{1}$	sum: $0 + \frac{3}{1} + \frac{4}{2}$	sum: $0 + \frac{11}{1} + \frac{12}{2} + \frac{13}{3} + \frac{14}{4} + \frac{15}{5} + \frac{16}{6} + \frac{17}{7} + \frac{18}{8} + \frac{19}{9} + \frac{20}{10}$
as expected	as expected	as expected

4) find the running time 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 take also one time unit, and function list takes n time units.



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

O(n), Θ(n)