Exercise 1-6: Big-O

Fill the T(*n*) and Big-O for Exercise 1 to 6 in the table below

Input/Method	Ex 1	Ex 2	Ex 3	Ex 4	Ex 5	Ex 6
T(*n*)						
Big-O						

Exercise 1

Python Code

```
def example1(s):
  n = len(s)
  total = 0
  for i in range(n):     # loop from 0 to n-1
    total += s[i]
  return total
```

T(n) = ?

Complexity: *O*(?)

Exercise 2

Python Code

```
def example2(s):
  n = len(s)
  total = 0
  for i in range(0,n,2):  # Increment of 2
    total += s[i]
  return total
```

T(n) = ?

Complexity: *O*(?)

Exercise 3

Python Code

```
def example3(s):
  n = len(s)
  total = 0
  for i in range(n):        # loop from 0 to n-1
    for k in range(1+i):    # loop from 0 to i
      total += s[k]
  return total
```

T(n) = ?

Complexity: *O*(?)

Exercise 4

Python Code

```
def example4(s):
  n = len(s)
  prefix = 0
  total = 0
  for i in range(n):
    prefix += s[i]
    total += prefix
  return total
```

T(n) = ?

Complexity: *O*(?)

Exercise 5

Python Code

```
# Assume that A and B have equal length of n
def example5(A,B):
  n = len(A)
  count = 0
  for i in range(n):          # loop from 0 to n-1
    total = 0
    for j in range(n):        # loop from 0 to n-1
      for k in range(1+j):    # loop from 0 to j
        total += A[k]
    if B[i] == total:
      count += 1
  return count
```

T(n) = ?

Complexity: *O*(?)

Exercise 6

Python Code

```
# Assume that A and B have equal length of n
def example5(A,B):
  n = len(A)
  count = 0
  for i in range(n):          # loop from 0 to n-1
    total = 0
    for j in range(n):        # loop from 0 to n-1
      for k in range(1+j):    # loop from 0 to j
        total += A[k]
    if B[i] == total:
      count += 1
  return count
```

T(n) = ?

Complexity: *O*(?)

Exercise 7.1: Summation

Write a Python function named summation_v1(*n*) where *n* is a positive integer (n > 0)

- Calculate the sum of an integer from 1 to *n*
- Return the total sum
- Use a loop

Exercise 7.2: Summation

Write a Python function named summation_v2(*n*) where *n* is a positive integer (n > 0)

- Calculate the sum of an integer from 1 to *n*
- Return the total sum
- A loop is not allowed, use the mathematical formula instead.

Exercise 7.3: Summation

Record both 7.1 and 7.2 running time (in seconds) for each of the following inputs:

- 100
- 10,000
- 1,000,000
- 100,000,000
- 1,000,000,000

Exercise 7.3: Summation

1) Submit the code
2) Fill the running time records in the table below

Input/Method	100	10,000	1,000,000	100,000,000	1,000,000,000	Big-O
7.1						
7.2						