

## Define the following 10 functions in your python source file.

[Tip] `str.upper()` 함수는 해당 문자열을 대문자로 변환하여 반환한다.

[Tip] `str.lower()` 함수는 해당 문자열을 소문자로 변환하여 반환한다.

[Tip] `help()` 함수를 사용하면 해당 타입에서 제공하는 메뉴얼을 확인할 수 있다.

(VS Code에서 `help()`를 사용할 때, 'q'를 누르면 메뉴얼 확인모드에서 빠져나올 수 있음)

[Tip] VS Code에서, 코드가 무한루프에 빠지면 **Ctrl+C** (커맨드+C on Mac)로 종료할 수 있다.

---

### function01 (1 point)

- The function name: **function01**
  - This function takes two arguments
    - First argument: int type
    - Second argument: int type
  - The return value
    - 두 인자의 합을 반환
  - 함수는 아래와 같이 **positional arguments**를 사용하여 테스트
    - `function01(1, 2)` returns 3
    - `function01(0, 0)` returns 0
    - `function01(-1, -2)` returns -3
- 

### function02 (1 point)

- The function name: **function02**
  - This function takes one argument
    - int type의 값을 요소로 갖는 tuple 또는 list
  - The return value
    - 주어진 list 또는 tuple의 모든 요소의 합을 반환. 아무 요소가 없다면 0 반환
  - 함수는 아래와 같이 **positional arguments**를 사용하여 테스트
    - `function02([1, 2, 3, 4, 5])` returns 15
    - `function02((1, 2, -3, 4))` returns 4
    - `function02([])` returns 0
- 

### function03 (2 points)

- The function name: **function03**
- This function takes two arguments
  - First argument: int type
  - Second argument: int type (두 번째 인자는 첫 번째 인자보다 큼)
- The return value
  - 첫 번째 인자와 두 번째 인자 사이에 해당하는 정수의 합을 반환  
(첫 번째 인자와 두 번째 인자에 해당하는 숫자는 합산에 포함하지 않음)
- 함수는 아래와 같이 **positional arguments**를 사용하여 테스트
  - `function03(1, 4)` returns 5
  - `function03(0, 3)` returns 3

- function03(-3, -1) returns -2
- 

#### function04 (2 points)

- The function name: **function04**
  - This function takes one argument
    - int type의 값을 요소로 갖는 tuple 또는 list
  - The return value
    - 주어진 list 또는 tuple의 요소 중 짝수의 합을 반환. 아무 요소가 없다면 0 반환
  - 함수는 아래와 같이 positional arguments를 사용하여 테스트
    - function04([1, 2, 3, 4, 5]) returns 6
    - function04((1, 3, 2)) returns 2
    - function04([]) returns 0
    - function04([-2, 0, 1]) returns -2
- 

#### function05 (2 points)

- The function name: **function05**
  - This function takes one argument
    - str type (두 글자 이상이며, 알파벳으로만 구성됨)
  - The return value
    - 주어진 문자열을 첫 번째 글자만 대문자, 나머지 글자는 소문자로 변환하여 반환
  - 함수는 아래와 같이 positional arguments를 사용하여 테스트
    - function05("hello") returns "Hello"
    - function05("heLLO") returns "Hello"
    - function05("pythoN") returns "Python"
    - function05("AA") returns "Aa"
- 

#### function06 (2 points)

- The function name: **function06**
- This function takes one argument
  - str type (두 글자 이상이며, 알파벳으로만 구성됨. 공백 없음)
- The return value (bool type)
  - 대소문자 상관없이, 주어진 문자열이 팰린드롬(palindrome)이라면 *True*, 팰린드롬이 아니라면 *False*를 반환
  - [Tip] 팰린드롬이란 "우영우", "기러기", "인도인", "level", "mygym", "ABCcba" 등과 같이 거꾸로 뒤집어도 원래의 단어와 동일한 문자열을 의미.
- 함수는 아래와 같이 positional arguments를 사용하여 테스트
  - function06("level") returns True
  - function06("levEL") returns True
  - function06("Aa") returns True
  - function06("hello") returns False (because "hello" != "olleh")
  - function06("pythoN") return False (because "pythoN" != "Nohtyp")

---

### function07 (2 points)

- The function name: **function07**
  - This function takes one argument
    - int type의 값을 요소로 갖는 tuple 또는 list. 아무 요소가 없다면 0 반환
  - The return value
    - 주어진 list 또는 tuple의 요소 중 중복되지 않는 고유값만의 합을 구하여 반환 (예: [1, 2, 3, 3, 3] 을 인자로 넘기면 중복되는 3 두 개를 제외하고, 1, 2, 3의 합을 반환)
  - 함수는 아래와 같이 positional arguments를 사용하여 테스트
    - function07([1, 2, 2, 3, 3, 3]) returns 6
    - function07((2, 2, 1)) returns 3
    - function07([-1]) returns -1
    - function07((-1, -1, 0, 1)) returns 0
- 

### function08 (2 points)

- The function name: **function08**
  - This function takes the unknown number of positional arguments
    - 몇 개의 인자가 넘어올지 알 수 없지만, 모든 인자는 int type
  - The return value
    - 모든 인자 중 짝수의 합을 계산하여 반환. 아무 인자가 없다면 0 반환
  - 함수는 아래와 같이 positional arguments를 사용하여 테스트
    - function08() returns 0
    - function08(1) returns 0
    - function08(1, 2, 3) returns 2
    - function08(1, -2, 3, 6) returns 4
    - function08(1, 2, 3, 4, 5) returns 6
- 

### function09 (3 points)

- The function name: **function09**
- This function takes the unknown number of positional or keyword arguments
  - 몇 개의 인자가 넘어올지 알 수 없지만, 모든 인자는 int type
- The return value
  - keyword arguments의 값(value) 중 가장 큰 값을 반환
  - keyword arguments가 없다면 -1을 반환
- 함수는 아래와 같이 positional and keyword arguments 모두 사용하여 테스트
  - function09() returns -1
  - function09(1, 2, 3) returns -1
  - function09(a=1, b=2) returns 2
  - function09(1, 2, 7, n1=5, n2=4) returns 5
  - function09(5, k=2) returns 2

---

### function10 (3 points)

- The function name: **function10**
- This function takes one argument
  - int type (1 이상의 정수)
- The return value
  - 피보나치 수열에서 주어진 인자의 순서에 해당하는 숫자를 반환
  - ※ 피보나치 수열이란 2개의 1로 시작하고, 그 뒤의 숫자는 앞의 두 숫자의 합으로 이뤄진 수열을 말한다.
    - 1, 1, 2, 3, 5, 8, 13, 21, 34 .....
- 함수는 아래와 같이 positional arguments를 사용하여 테스트
  - function10(1) returns 1
  - function10(2) returns 1
  - function10(3) returns 2
  - function10(4) returns 3
  - function10(5) returns 5
  - function10(6) returns 8

---

### 제출 전 확인 사항

- 본인의 파일 이름이 **\_학번.py** 형식을 갖추고 있는지 확인 → 틀리면 감점 -1
- 테스트용 함수 호출 코드는 모두 삭제 → 모든 함수가 잘 동작하더라도 함수 외 다른 코드로 인해 에러 발생시 감점 -1
- **print()** 삭제 또는 주석처리
- 과제게시판에 제출을 완료했으면 시험 시간이 종료될 때까지 노트북을 덮고 본인의 준비한 책이나 프린트물로 다른 시험 공부 가능

**Define the following 10 functions in your python source file.**

[Tip] `str.upper()` returns a string where all characters are in upper case.

[Tip] `str.lower()` returns a string where all characters are in lower case.

[Tip] use `help()` if necessary (press 'q' button to quit from help() menial on the VS Code)

[Tip] press Ctrl+C (or Command+C on Mac) to escape from the infinite loop on VS Code.

---

**function01 (1 point)**

- The function name: **function01**
  - This function takes two positional arguments
    - First argument: int type
    - Second argument: int type
  - The return value
    - sum of the two arguments
  - For example, this function will be tested with positional arguments as follows:
    - `function01(1, 2)` returns 3
    - `function01(0, 0)` returns 0
    - `function01(-1, -2)` returns -3
- 

**function02 (1 point)**

- The function name: **function02**
  - This function takes one positional argument
    - a *list* or *tuple* which has only int type elements
  - The return value
    - Sum of all elements. Return 0 if there is no element.
  - For example, this function will be tested with positional arguments as follows:
    - `function02([1, 2, 3, 4, 5])` returns 15
    - `function02((1, 2, -3, 4))` returns 4
    - `function02([])` returns 0
- 

**function03 (2 points)**

- The function name: **function03**
- This function takes two positional arguments
  - First argument: int type
  - Second argument: int type (this argument is bigger than the first argument)
- The return value
  - sum of integers which is between the first argument and the second argument (not including the first and second arguments)
- For example, this function will be tested with positional arguments as follows:
  - `function03(1, 4)` returns 5
  - `function03(0, 3)` returns 3

- function03(-3, -1) returns -2
- 

#### function04 (2 points)

- The function name: **function04**
  - This function takes one positional argument
    - a *list* or *tuple* which has only int type elements
  - The return value
    - Sum of elements which are even. Return 0 if there is no element.
  - For example,
    - function04([1, 2, 3, 4, 5]) returns 6
    - function04((1, 3, 2)) returns 2
    - function04([]) returns 0
    - function04([-2, 0, 1]) returns -2
- 

#### function05 (2 points)

- The function name: **function05**
  - This function takes one positional argument
    - str type (a length of the argument is greater than 1. All the characters are alphabet letters.)
  - The return value
    - The string with the uppercase first character and lowercase remaining characters
  - For example, this function will be tested with positional arguments as follows:
    - function05("hello") returns "Hello"
    - function05("heLLO") returns "Hello"
    - function05("pythoN") returns "Python"
    - function05("AA") returns "Aa"
- 

#### function06 (2 points)

- The function name: **function06**
- This function takes one positional argument
  - str type (the length of this argument is greater than 1. All the characters are alphabet letters, no spaces)
- The return value (bool type)
  - *True* if the argument is palindrome, *False* otherwise. (ignore cases)
  - [Tip] The palindrome is a word that reads the same backwards as forwards, such as "우영우", "기러기", "인도인", "level", "mygym", and "ABCcba".
- For example, this function will be tested with positional arguments as follows:
  - function06("level") returns True
  - function06("levEL") returns True
  - function06("Aa") returns True
  - function06("hello") returns False (because "hello" != "olleh")

- function06("pythoN") return False (because "pythoN" != "Nohtyp")
- 

### function07 (2 points)

- The function name: **function07**
  - This function takes one positional argument
    - a *list* or *tuple* which has only int type elements
  - The return value
    - Sum of only unique elements from the given argument (for example, if [1, 2, 3, 3, 3] is given, return sum of 1, 2, 3 after removing two 3 which are duplicated.). Return 0 if there is no element.
  - For example, this function will be tested with positional arguments as follows:
    - function07([1, 2, 2, 3, 3, 3]) returns 6
    - function07((2, 2, 1)) returns 3
    - function07([-1]) returns -1
    - function07((-1, -1, 0, 1)) returns 0
- 

### function08 (2 points)

- The function name: **function08**
  - This function takes the unknown number of positional arguments
    - all arguments are int type
  - The return value
    - Sum of the arguments which are even. Return 0 if there is no argument.
  - For example, this function will be tested with positional arguments as follows:
    - function08() returns 0
    - function08(1) returns 0
    - function08(1, 2, 3) returns 2
    - function08(1, -2, 3, 6) returns 4
    - function08(1, 2, 3, 4, 5) returns 6
- 

### function09 (3 points)

- The function name: **function09**
- This function takes the unknown number of positional or keyword arguments
  - All arguments are int type
- The return value
  - A maximum value of the keyword arguments' values (not keys)
  - If there are no keyword arguments, return -1
- For example, this function will be tested with positional and keyword arguments as follows:
  - function09() returns -1
  - function09(1, 2, 3) returns -1
  - function09(a=1, b=2) returns 2

- `function09(1, 2, 7, n1=5, n2=4)` returns 5
  - `function09(5, k=2)` returns 2
- 

### **function10 (3 points)**

- The function name: **function10**
  - This function takes one positional argument
    - int type (positive integer)
  - The return value
    - $N$  th (the argument) number from Fibonacci sequence
    - ✖ The Fibonacci sequence is a set of integers that starts with two 1. Then, the sequence follows the rule that each number is equal to the sum of the preceding two numbers.
      - 1, 1, 2, 3, 5, 8, 13, 21, 34 .....
  - For example, this function will be tested with positional arguments as follows:
    - `function10(1)` returns 1
    - `function10(2)` returns 1
    - `function10(3)` returns 2
    - `function10(4)` returns 3
    - `function10(5)` returns 5
    - `function10(6)` returns 8
- 

### **Checklist**

- Check if your file name is **\_STUDENTNUMBER.py** → -1 point if it is wrong
- Remove all unnecessary code such as function calls to test your code → -1 point if it raises errors.
- Remove or comment your *print()*
- After submitting your work, close your laptop. Then, you can see other books or paper-materials until the mid-term ends.