# Mock1 Test3

NAME, SURNAME, STUDENT ID (IN CAPITAL LETTERS): ……………………………………………………………………………………………

**Create programs in files with the names given in parentheses at the beginning of each task. If you use other names, you will not receive points.**

**Each function has a name f(). If you use a different name, you will not receive points.**

(p1.py) Create a function f(word) that, for a given word, returns a string of characters in which the subsequent letters of the word form the so-called Mexican Wave. Initially, the first letter of the word is capitalized and the remaining letters are lowercase. Then, the second letter of the word is capitalized and the remaining letters are lowercase, etc. Separate words with a minus sign. Example:

f("book") 🡪 "Book-bOok-boOk-booK"  
f("water") 🡪 "Water-wAter-waTer-watEr-wateR"  
f("ok") 🡪 "Ok-oK"  
f("a") 🡪 "A"  
f("") 🡪 ""

(p2.py) Create a function f(x,y,digit) that returns how many times the given digit appears in numbers in the range from x to y. Example:

f(10,15,1) 🡪 7  
f(28,32,2) 🡪 3  
f(100,105,6) 🡪 0  
f(100,101,0) 🡪 3

(p3.py) The uid array contains user IDs in one of the popular websites. Identifiers should be unique. Create a function f(uid) that returns true if all ids are unique. Otherwise, the function returns false. Example:

f(["john5","ann123","JOHN5","xxx","abc333","a10"]) 🡪 True  
f(["abc123","ann","abc123","a10"]) 🡪 False  
f(["bob2","bob2"]) 🡪 False  
f(["bob2","BOB2"]) 🡪 True

(p4.py) The prods array contains the names of the products in stock. Create a function f(fnc,prods) that maps product names to their IDs, according to the fnc function. The f function returns a comma-separated list of product IDs, with no spaces. Example:

prods = ["water","cheese","tomato"]   
fnc1 = lambda x: "id:"+x[:2]   
f(fnc1,prods) 🡪 " id:wa,id:ch,id:to"  
fnc2 = lambda x: (x[0]+x[-1]).upper()   
f(fnc2,prods) 🡪 "WR,CE,TO"

(p5.py) A counter allows you to count any elements. Define a class C to create counters. The initial value of the counter is assigned when the object is created. The class contains the following methods:

* m1() - returns the counter value
* m2() - increases the counter value by 1
* m3() - decreases the counter value by 1
* m4(n) - increases/decreases the counter value by n
* \_\_str\_\_ - returns a string representation of the counter value

Example:

c=C(5)  
c.m1() 🡪 5  
c.m2()  
c.m1() 🡪 6  
c.m4(-8)  
c.m1() 🡪 -2  
c.m3()  
c.m1() 🡪 -3  
c.m4(10)  
c.m1() 🡪 7  
c.\_\_str\_\_() 🡪 "7"

(p6.py) Assume that a valid variable name in a computer program consists of one to five characters. The first character of a variable name must be a lowercase or uppercase letter or an underscore. The remaining characters in the variable name can be uppercase or lowercase letters, digits or the underscore character. Create a function f(vname) that returns true if the variable name vname is valid. Otherwise, the function returns false. Example:

f("abc") 🡪 True  
f("Abc") 🡪 True  
f("aBC") 🡪 True  
f("\_ab\_c") 🡪 True  
f("abcdef") 🡪 False  
f("8abc") 🡪 False  
f("\_aB8\_") 🡪 True  
f("\_4x") 🡪 True

(p7.py) Create a function f(arr2D) that returns true when the sum of the values in at least two columns is the same. Otherwise, the function returns false. Example:

arr = [[3,4,2],[2,2,2,0],[5,0,0,5],[4,7,0,2],[0,2,0,0]]  
f([[3,4,2],[5,1,6]]) 🡪 True  
f([[3,4,2],[5,1,7]]) 🡪 False  
f([[3,4],[5,1],[4,7]]) 🡪 True  
f([[3,4],[5,9],[4,7]]) 🡪 False