

PYTHON TEST

1. Definiti in python3 clasa Laptop, care sa aiba urmatoarele campuri: cpus, capacity, hasSSD. Obiectele de tip Laptop trebuie sa aiba urmatorul comportament:

```
laptop1 = Laptop()
laptop2 = Laptop(4, 512, True)
```

```
print(laptop1) sa afiseze:
(cpus = 0, capacity = 0, hasSSD = False)
```

```
print(laptop2) sa afiseze:
(cpus = 4, capacity = 512, hasSSD = True)
```

```
l = [laptop1, laptop2]
print(l) sa afiseze:
[(cpus = 0, capacity = 0, hasSSD = False), (cpus = 4, capacity = 512, hasSSD = True)]
```

2. Sa se implementeze in python3 functia zipLongest, cu urmatorul comportament:

```
zipLongest(l1, l2)

print(list(zipLongest([1,2,3], [4,5,6,7,8]))) sa afiseze
[(1, 4), (2, 5), (3, 6), (0, 7), (0, 8)]
```

Obs: Se poate folosi functia zip. Comportamentul este asemanator, doar ca nu se mai opreste la lista cea mai scurta, ci pune 0 pe pozitiile care nu exista fata de lista lunga

3. Sa se implementeze functia palindromeMap, care primeste o lista de cuvinte si intoarce o lista de string-uri "Yes"/"No" . Este obligatorie folosirea de list comprehension, nu este permisa folosirea functiei map.

Exemplu de utilizare:

```
palindromeMap(["ana", "are", "mere", "aerisirea", "rever", "pere"]) intoarce
["Yes", "No", "No", "Yes", "Yes", "No"]
```

Hint: Se recomanda definirea predicatului isPalindrome(word), care verifica daca un cuvant primit ca parametru este sau nu palindrom

Pentru a verifica ca un cuvant este palindrome, se poate folosi functia reversed, care intoarce o lista continand caracterele cuvantului in ordine inversa. Poate fi utila si functia str.join.

```
reversed("are") => ['e', 'r', 'a']
str.join("_", ["a", "b", "c"]) => 'a_b_c'
```

4. Fie urmatorul cod in python3.

```
digits = [ i for i in range(10) ]
```

```
v1 = { i:i**2 for i in digits }
```

```
v2 = [ [i, i**2] for i in digits ]
```

Precizati valorile variabilelor v1 si v2.

5. Fie variabila:

```
my_string = "protectMe"
```

Aceasta se simte "in siguranta" daca este precedata de 1000 de caractere '#' si urmata de 500 de caractere '+'. Suprascrieti variabila my_string, astfel incat aceasta sa se simta "in siguranta".

6. Lista este un tip de date fundamental in python.

a) Scrieti 3 moduri prin care se poate crea o lista vida in python3.

b) Fie l = [1, 2, 3]. La ce se evalueaza l[: -1] (eroare sau o valoare, cu precizarea acesteia).

c) Precizati ce afiseaza urmatorul cod:

```
alphabet = map(chr, range(97, 123))
```

```
l = list(alphabet)
```

```
s = str.join(", l[:5])
```

```
l = list(s)
```

```
print(l)
```

7. Fie urmatorul cod:

```
class Father():
```

```
    def __init__(self):
```

```
        print("Father constructor")
```

```
    def function(self):
```

```
        print("Father function")
```

```
class Child(Father):
```

```
    def __init__(self):
```

```
        print("Child constructor")
```

```
    def function(self):
```

```
        print("Child function")
```

```
        self.function()
```

```
child = Child()
```

```
child.function()
```

a) La rularea intr-un interpretor python3, primim urmatoarea eroare: `RecursionError: maximum recursion depth exceeded while calling a Python object`
Corectati eroarea modificand doar codul din clasa `Child` si precizati ce se afiseaza.

b) Realizati modificarile necesare in clasa `Child` astfel incat codul anterior sa afiseze urmatoarele:

Child constructor
Father constructor
Child function
Father function

8. Implementati in python3 functia `find` cu urmatorul comportament:

`find(start_path, file_to_search)`

- cauta recursiv incepand cu `start_path` fisierul cu numele `file_to_search`
- se opreste la primul fisier gasit in ierarhia de cai care incep cu `start_path`
- daca gaseste fisierul afiseaza "File found" si calea catre fisier, in caz contrar nu afiseaza nimic

exemplu de utilizare:

`find("tmp", "my_file")` afiseaza "File found ./tmp/my_folder/secret/my_file"
avand urmatoarea ierarhie de fisiere

```
tmp/  
    my_folder/  
    another_file  
    another_folder  
        secret/  
            top_secret  
            my_file
```

Hint: Se pot folosi urmatoarele functii din modulul `os`:

`os.listdir(path)`
`os.path.join(path1, path2)`
`os.path.isdir(path)`

Bonus:

9. Explicati cum functioneaza MVC Pattern (Model-View-Controller).