

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare Informatică și Microelectronică**

**Departamentul Informatică și Ingineria Sistemelor**

**Disciplina: Programarea intereactivă**

**Lucrare de laborator №1**

**Tema: Introducere în limbajul Python. Expresii, variabile, funcţii. Instrucţiunile logice şi decizionale.**

A efectuat:

Pleșca Virgiliu,

*student*

IA-201

A controlat:

Strună Vadim,

*lector univ.*

**Chișinău 2021**

# Task

## Obiective:

* Instalare interpretor / IDE python
* Familiarizare cu limbajul python
  + Execuţie instrucţiuni simple în interpretor
  + Implementare/executare programe simple folosind Python IDLE

## Cerinţe:

* Rezolvaţi în timpul laboratorului problemele de la punctul I și II.

## I Probleme

1. Scrieţi o funcţie mile\_kilometri care ia un parametru *mile* şi returnează numărul de *kilometri*.
2. Scrieţi o funcţie seconde\_totale care ia trei parametri *ore, minute şi secunde* şi returnează numărul total de secunde pentru variabilele ore, minute şi secunde.
3. Scrieţi o funcţie suprafata\_dreptunghi care are doi parametri *lăţimea* şi *înălţimea* corespunzătoare lungimilor laturilor unui dreptunghi şi returnează perimetrul și aria dreptunghiului în cm.
4. Scrieţi o funcţie circumferinta\_cerc care are un parametru *raza* ce corespunde razei unui cerc în cm şi returnează circumferinţa și aria circumferinței unui cerc în cm. Pentru a furniza o aproximare mai mare la precizie pentru variabila π utilizaţi funcţia pi din modulul math. (ex. import math math.pi)
5. Scrieţi o funcţie depozit\_valoare care are trei parametri *valoare\_curenta*, *rata\_anuala* şi *ani\_depozit* şi returnează valoarea\_sumei\_viitoare\_de\_bani a valoare\_curenta investită la dobânda rata\_anuala în procente, formată anual.
6. Scrieţi o funcţie virsta\_persoana care are parametrul ziua de naştere (zi/luna/an), şi va returna vîrsta persoanei în zile.

## II. Probleme

1. Scrieţi o funcţie numar\_par care are ca intrare un număr întreg şi returnează True dacă numărul este par şi False dacă numărul este impar.

2. Scrieţi o funcţie gaseste\_nume care are ca input un nume şi returnează True dacă numele este fie "Ion", "Elena" sau "Maria" şi se întoarce False în celelalte cazuri.

3. Scrieţi o funcţie an\_bisect care ia ca date de intrare anul şi returnează True dacă anul (un număr întreg) este un an bisect, conform calendarului gregorian şi False în caz contrar. (Sfat Intră pe Wikipedia: este un algoritm simplu pentru a stabili dacă un an este un an bisect.)

4. Scrieţi o funcţie intersectare\_interval care ia parametrii a, b, c şi d, şi returnează True dacă intervalele [a, b] şi [c, d] se intersectează şi False în caz contrar. În timp ce acest test poate părea complicat, soluţia este de fapt foarte simplă.

5. Scrieţi o funcţie numele\_si\_virsta care ia ca date de intrare numele (un şir de caractere) şi vîrsta (un număr) şi returnează un şir de forma " % este % de ani." unde procentele sunt numele şi vîrsta sub forma de şir de caractere. Funcţia trebuie să includă o verificare de eroare pentru cazul în care vîrsta este mai mică decât zero. În acest caz, funcţia ar trebui să returneze şirul "Eroare: vârstă invalid".

6. Scrieţi o funcţie afiseaza\_numerele care ia un număr întreg din intervalul [0,100) şi afişează mesajul "Numarul consta din % zeci si % unitati." unde procentele ar trebui să fie înlocuite cu valorile corespunzătoare. Funcţia trebuie să includă o verificare de eroare pentru cazul în care numărul este negativ sau mai mare sau egal cu 100. În aceste cazuri, funcţia ar trebui să afişeze un mesaj de erare "Eroare: Numarul introdus nu este un număr de două cifre.".

7. Scrieţi o funcţie cauta\_nume care are ca parametru prenume, care corespunde unuia dintre ("Vasile", "Mihail", "Ion" sau "Stefan") şi apoi returnează numele lor corespunzătoare ("Moraru", "Perebinos", "Sirghi" sau "Buzurniuc"). Dacă prenume nu se potriveşte cu niciuna dintre aceste şiruri de caractere, returnează eroare "Eroare: Nu este profesor".

8. Pig Latin este un joc de limbaj care implică modificarea cuvintelor printr-un simplu set de reguli. Scrieţi o funcţie pig\_latin care are ca parametru un cuvînt şi aplică următoarele reguli pentru a genera un nou cuvînt în limbajul pig latin.

- Dacă prima literă din cuvînt este o consoană, adăugaţi consoana la sfîrşit şi adăugaţi "ay" la restul cuvîntului. De exemplu, pig\_latin ("pig") va rezulta "igpay".

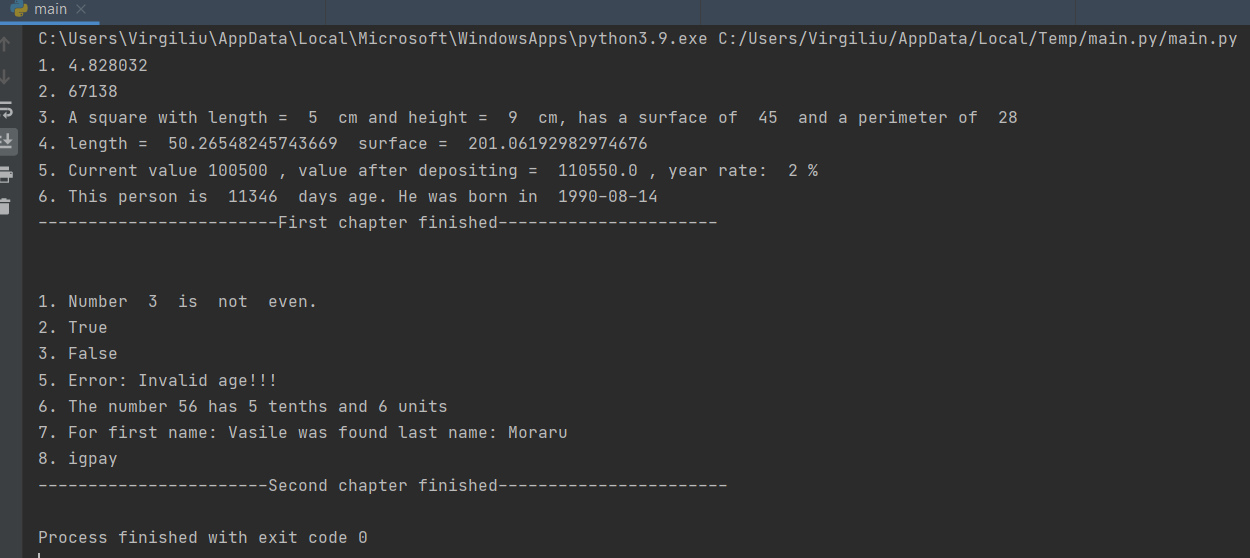
- Dacă prima literă din cuvînt este o vocală, adăugaţi "way" la sfîrşitul cuvîntului. De exemplu, pig\_latin ("awl") returnează "owlway".

# Solving the problems

## Source code

import math  
from datetime import date  
  
# -----------------------------First chapter------------------------------------  
  
  
# 1. Miles into km--------------------------  
def mile\_km(mile):  
 return (mile \* 1.609344)  
  
  
print("1.", mile\_km(9))  
  
  
# 2. Time in seconds-------------------------  
def time\_converter(h, m, s):  
 return (h \* 3600) + (m \* 60) + s  
  
  
print("2.", time\_converter(1, 1, 1))  
  
  
# 3. Square surface & perimeter---------------  
def surface\_perimeter(length, width):  
 s = length \* width #suprafata patrulaterului  
 p = (2 \* length) + (2 \* width) #perimetrul  
 return [s, p]  
  
  
lt = 5  
h = 9  
square\_data = surface\_perimeter(lt, h)  
print("3. A square with length = ", lt, " cm and height = ", h, " cm, has a surface of ", square\_data[0], " and a perimeter of ", square\_data[1])  
  
  
# 4. Circle length and surface---------------------  
def circumferinta\_cerc(r):  
 lt = 2 \* math.pi \* r # Lungimea circumferintei  
 s = math.pi \* r\*\*2 # Suprafata cercului  
 return [lt, s]  
  
  
print("4. length = ", circumferinta\_cerc(8)[0], " surface = ", circumferinta\_cerc(8)[1])  
  
  
# 5. Deposit-----------------------------------------------------------------------------------  
def depozit\_valoare(valoare\_curenta, rata\_anuala, n\_ani):  
 total\_gain = 0 # Initializam cu 0 pentru a putea efectua in continuare operatii  
  
 for i in range(n\_ani): # Calculeaza dobanda pentru numarul de ani pentru depozitare (se repeta de "n\_ani" ori)  
 total\_gain = total\_gain + (valoare\_curenta \* ((rata\_anuala + 100) / 100) - valoare\_curenta)  
  
 return valoare\_curenta, valoare\_curenta + total\_gain, rata\_anuala # total\_gain este dobanda castigata in anii de depozitare  
  
  
print("5. Current value", depozit\_valoare(100500, 8, 5)[0], ", value after depositing = ", depozit\_valoare(100500, 2, 5)[1], ", year rate: ", depozit\_valoare(100500, 2, 5)[2], "%")  
  
  
# 6. Person age----------------------------------------------------------------  
def varsta\_persoana(day, month, year):  
 curr\_date = date.today()  
 birth\_date = date(year, month, day)  
 age = curr\_date - birth\_date  
  
 return age.days  
  
  
print("6. This person is ", varsta\_persoana(14, 4, 2001), " days age. He was born in ", date(1990, 8, 14))  
print("------------------------First chapter finished----------------------\n\n")  
  
# -------------------------------------Second chapter-------------------------------------  
  
  
# 1. Numar par-----------  
def is\_even(n):  
 if n % 2 == 0:  
 return True  
 else:  
 return False  
  
  
n = 3  
a = "not" if is\_even(n) else ""  
print("1. Number ", n, " is ", a, " odd.")  
  
  
# 2. Gaseste nume----------------------------  
def gaseste\_nume(name):  
 if name == "Ion":  
 return True  
 elif name == "Elena":  
 return True  
 elif name == "Maria":  
 return True  
 else:  
 return False  
  
  
print("2.", gaseste\_nume("Maria"))  
  
  
# 3. Leap year------------------------------  
def an\_bisect(year):  
 if year % 4 != 0:  
 return False  
 elif year % 100 != 0:  
 return True  
 elif year % 400 != 0:  
 return False  
 else:  
 return True  
  
  
print("3.", an\_bisect(2100))  
  
  
# 4. Intersection interval  
def intersectare\_interval(a, b, c, d):  
 for i in range(a, b):  
 if i in range(c, d):  
 return True  
 else:  
 return False  
  
print("Intervalele [1, 4] si [2, 5] se intersecteaza = ", intersectare\_interval(1, 4, 2, 5))  
  
# 5. Name and age  
def numele\_si\_varsta(name, age):  
 if age < 0:  
 return "Error: Invalid age!!!"  
 else:  
 return name + age.\_\_str\_\_()  
  
  
print("5.", numele\_si\_varsta("Gingis Han, ", -5))  
  
   
# 6. Afiseaza numere  
def afiseaza\_numere(n):  
 if n < 0 or n >= 100:  
 print("6. Error: Number out of the range: [0, 100)")  
 return  
 else:  
 n2 = n  
 u = n % 10  
 n = math.trunc(n / 10)  
 z = n % 10  
  
 print("6. The number", n2, "has", z, "tenths and", u, "units")  
  
  
afiseaza\_numere(56)  
  
  
# 7. Find names.  
# Am schimbat din array de tupluri in dictionar si am folosit Try - Except  
teacher\_data = {  
 "Vasile": "Moraru",  
 "Mihail": "Perebinos",  
 "Ion": "Sirghi",  
 "Stefan": "Buzurniuc"  
}  
  
  
def find\_lastname(first\_name):  
 try:  
 print("7. For the first name:", first\_name, "has been found last name:", teacher\_data[first\_name])  
 except:  
 print("7. Error: No such teacher")  
  
  
find\_lastname("Vasile")  
  
  
# 8. Pig Latin  
def is\_vocala(c): # Verifies if the parameter is a vocal  
 if c == 'a' or c == 'e' or c == 'i' or c == 'o' or c == 'u':  
 return True  
 else:  
 return False  
  
  
def move\_fst\_2\_end(s): # Moves first char at the end of the string  
 aux = s[0]  
 s = s[1:]  
 s = s + aux  
  
 return s  
  
  
def pig\_latin(s):  
 if is\_vocala(s[0]):  
 return s + "way"  
 else:  
 return move\_fst\_2\_end(s) + "ay"  
  
  
print("8.", pig\_latin("pig"))  
print("-----------------------Second chapter finished-----------------------")

## Output



# Conclusion

After doing this laboratory work i have familiarised with Python programming language. Now i am more confident using Python. I think that i understood perfectly basic statements, variable declaration and that stuff which is nearly the same in each of the programming languages. The only thing i had to get used with was the sintax. Also I’ve made a conclusion about Python that it is a language for “lazy people”, because there are lots and lots of librarys thar can do the work for you, the only thing you have to do is to type *pip install <package>* in console and then use it in your projects. In this way people are developing projects in Python with a speed close to the speed of light :). There is even a quote about this: “Sloth is the mother of invention”.