

Tutoriat 5 – Fișiere. Funcții - rezolvări -

1. a) Scrieți o funcție care primește ca parametru un număr natural și stabilește dacă numărul este prim.
- b) Scrieți o funcție care primește ca parametru o listă de numere naturale și returnează două liste: una cu numere prime și alta cu numere compuse, folosind funcția de la punctul a).
- c) Scrieți un program care citește din fișierul "numere.txt" o listă de numere și afișează în fișierul "prime.txt" lista numerelor prime, folosind funcția de la punctul b).

```
def prim( n ):
    if n < 2:
        return False
    for i in range( 2, int( n ** ( 1 / 2 ) ) + 1 ):
        if n % i == 0:
            return False
    return True

def primcomp( lista ):
    prime = []
    compuse = []
    for i in lista:
        if prim(i) == True:
            prime.append(i)
        else:
            compuse.append(i)
    return prime, compuse

def primcomp2( lista ):
    prime = [ x for x in lista if prim(x) ]
    compuse = [ x for x in lista if not prim(x) ]
    return prime, compuse

f = open( "numere.txt", 'r' )
lista = [ int(x) for x in f.readline().split() ]
f.close()
with open( "prime.txt", 'w' ) as g:
    g.write( str(primcomp(lista)[0]) )
```

2. Definiți o funcție care primește ca parametru un număr variabil de numere naturale și întoarce lista formată doar din numerele pare.

```
def pare( *numere ):
    pare = []
    for i in numere:
        if i % 2 == 0:
            pare.append(i)
    return pare
```

3. a) Scrieți o funcție care primește un număr variabil de parametri, numai liste de numere întregi, și returnează rezultatul concatenării listelor.
- b) Scrieți un program care citește din fișierul "liste.txt", de pe fiecare linie, câte o listă de numere întregi și scrie în același fișier rezultatul concatenării listelor, folosind funcția de la punctul a).

```
def concat( *args ):
    rez = []
    for ls in args:
        rez = rez + ls
    return rez

with open("liste.txt", 'r') as f:
    liste = [ [int(x) for x in lista.split()] for lista in f.readlines() ]
with open("liste.txt", 'w') as g:
    g.write( str( concat(*liste) ) )
```

4. a) Scrieți o funcție care primește ca parametri informații despre un angajat: numele, salariul și, opțional, un procent. Să se returneze numele și salariul angajatului, după ce a fost crescut cu procentul respectiv. Dacă procentul lipsește, salariul nu se va modifica.
- b) Scrieți un program care citește din fișierul "initial.txt", de pe fiecare linie, informații despre câte un angajat și scrie în fișierul "final.txt", pe câte o linie a fișierului, rezultatul apelării funcției de la punctul a) cu informațiile despre angajatul respectiv date ca parametru.

```
def crestere(nume, salariu, procent = 0):
    return nume, salariu + procent / 100 * salariu

f = open("initial.txt", 'r')
g = open("final.txt", 'w')
for linie in f:
    angajat = linie.split()
    if len( angajat ) == 2:
        g.write( str(crestere( angajat[0], int(angajat[1]) )) )
    else:
        g.write( str(crestere( angajat[0], int(angajat[1]), int(angajat[2]) )) )
f.close()
g.close()
```