

Laborator 4

Fișiere, Generatori, Excepții

1. În fișierul text “test.in” se află testul unui elev de clasa a II-a la matematică, conținând 9 înmulțiri scrise pe rânduri distincte. Un calcul corect este notat cu un punct, iar unul incorect cu 0 puncte. Să se realizeze un program care să evalueze testul dat, astfel: în dreptul fiecărui calcul corect se va scrie mesajul ‘Corect’, iar în dreptul fiecărui calcul greșit se va scrie mesajul ‘Greșit’ și rezultatul corect, iar la final se va scrie nota (un punct se acordă din oficiu!). Rezultatul evaluării testului se va scrie în fișierul text “test.out”.

Exemplu:

test.in 3*4=11 2*10=20 5*5=24 7*4=28 2*6=12 10*10=100 3*9=27 6*7=33 0*9=1	test.out 3*4=11 Greșit 12 2*10=20 Corect 5*5=24 Greșit 25 7*4=28 Corect 2*6=12 Corect 10*10=100 Corect 3*9=27 Corect 6*7=33 Greșit 42 0*9=1 Greșit 0 Nota 6
---	--

2. Departajare

La facultate se ține un concurs de programare. Organizatorii au aranjat ca punctajele să fie calculate automat, din codul scris. Totuși, ei nu au implementat vreo metodă de a departaja punctajele egale, așa că în ultimul moment s-au decis ca persoana care a trimis codul prima să fie clasată mai sus decât cel de-al doilea ca timp, dar cu punctaj identic, ș.a.m.d. Până la introducerea lui “-1”, să se citească de la tastatură perechi de două elemente: (punctaj, nume_student), cu punctajul între 0 și 100. Perechile se consideră a fi introduse în ordinea predării codului de către participanți. a) Salvați datele de intrare într-o listă de tuple (punctaj, nume_student, nr_ordine), astfel încât să se poată deduce și numărul de ordine pentru fiecare participant (al câtelea a predat codul). b) Să se afișeze lista tuturor punctajelor distincte obținute de participanți (folosiți un set). c) Într-un dicționar, pentru fiecare punctaj să se asocieze o listă de tuple ce conțin numele participantului și numărul său de ordine. Apoi, să se afișeze clasamentul

concursului (descrescător după punctaje, pe câte un rând: punctaj, nume_participant și nr_ordine).

De exemplu, pentru următorul input:

```
64 Danil Marius
70 Derek Alexandru
100 Pirpiric Claudiu
18 Alexandrescu Matias
64 Popescu Catalin
100 Cozia Daniel
82 Stefan Dinca
-1
```

Vom obtine: a) Lista [(64, Danil Marius, 1), (70, Derek Alexandru, 2), (100, Pirpiric Claudiu, 3), (18, Alexandrescu Matias, 4), (64, Popescu Catalin, 5), (100, Cozia Daniel, 6), (82, Stefan Dinca, 7)]

b) Mulțimea 100, 82, 70, 64, 18 (nu neapărat în această ordine)

c) Și dicționarul (având perechile nu neapărat în această ordine):

```
{
100: [(Pirpiric Claudiu, 3), (Cozia Daniel, 6)],
82: [(Dinca Stefan, 7)],
70: [(Derek Alexandru, 2)],
64: [(Danil Marius, 1), (Popescu Catalin, 5)],
18: [(Alexandrescu Matias, 4)]
}
```

3. Generator

Implementați un generator infinit de numere prime. Folosind acest generator, scrieți un program care citește de la tastatură un număr natural nenul $n \leq 2$ și afișează pe ecran:

1. numerele prime cel mult egale cu n ;
2. primele n numere prime.

4. Tratarea excepțiilor

1. Să se scrie o funcție “min_max” care primește un număr variabil de parametri (numere naturale) și returnează cel mai mic și cel mai mare număr dintre cele primite ca parametri, dacă există cel puțin un parametru și dacă toți parametrii sunt numere naturale, sau returnează None altfel.

2. Să se citească tot conținutul fișierului text “numere.txt” și apoi să se afișeze pe ecran rezultatele obținute aplicând funcția “min_max” asupra sa. Dacă valoarea returnată de funcția min_max este diferită de None, se va scrie în fișierul text “impartire.txt” rezultatul împărțirii valorii maxime din fișierul text la cea minimă. Să se trateze excepțiile care pot să apară: nu există fișierul text de intrare, nu există drept de scriere pentru fișierul de ieșire, fișierul de intrare conține valori care nu sunt numere naturale, împărțire la zero etc.

Exemplu: Dacă fișierul text “numere.txt” conține: 11 9 31 7 145 5 101 4 80, atunci funcția “min_max” va returna (4, 145), iar în fișierul “impartire.txt” se va scrie: 36.25

5. Sortare folosind comparatori Din fișierul “cuvinte.txt” se citesc cuvinte care se salvează într-o listă L. În fișierul “cuv_sortate.txt” să se scrie, pe câte o linie, lista L sortată astfel:

1. descrescător, folosind compararea implicită
2. crescător, comparând după lungimea cuvintelor și apoi după ordinea alfabetică
3. crescător după lungimea cuvintelor, dar pentru cuvintele de aceeași lungime păstrând ordinea din lista originală.

Exemplu:

Dacă $L = ['zi', 'ana', 'sac', 'acadea', 'bac', 'nori', 'zar', 'mi', 'abur']$, fișierul cuv_sortate.txt va avea următorul conținut:

- a) ['zi', 'zar', 'sac', 'nori', 'mi', 'bac', 'ana', 'acadea', 'abur']
- b) ['mi', 'zi', 'ana', 'bac', 'sac', 'zar', 'abur', 'nori', 'acadea']
- c) ['zi', 'mi', 'ana', 'sac', 'bac', 'zar', 'nori', 'abur', 'acadea']