MODALITATEA DE DESFĂȘURARE A EXAMENULUI LA DISCIPLINA "PROGRAMAREA ALGORITMILOR" DIN SESIUNEA DE REEXAMINĂRI ȘI MĂRIRI 01.09.2022 – 07.09.2022

- Examenul la disciplina "Programarea algoritmilor" se va desfășura în ziua de 02.09.2022, între orele 10^{00} și 15^{00}, astfel:
  - 10^{00} – 10^{15}: efectuarea prezenței studenților la testul de laborator
  - 10^{15} – 11^{45}: desfășurarea testului de laborator
  - 11^{45} – 12^{00}: verificarea faptului că sursele trimise de către studenți au fost salvate pe platformă
  - 12^{00} – 12^{30}: pauza
  - 12^{30}– 12^{45}: efectuarea prezenței studenților la examenul scris
  - 12^{45} – 14^{45}: desfășurarea examenului scris
  - 14^{45} – 15^{00}: verificarea faptului că fișierele trimise de către studenți au fost salvate pe platforma MS Teams

- Ambele probe se vor desfășura pe platforma MS Teams, iar pe tot parcursul desfășurării lor studenții trebuie să fie conectați pe canalul dedicat cursului de "Programarea algoritmilor" corespunzător seriei lor.
- În momentul efectuării prezenței, fiecare student trebuie să aibă pornită camera video în MS Teams și să prezinte buletinul sau cartea de identitate. Dacă dorește să-și protejeze datele personale, studentul poate să acopere codul numeric personal și/sau adresa!
- În timpul desfășurării testului studenții pot să închidă camera video, dar trebuie să o deschidă dacă li se solicită acest lucru de către un cadru didactic!

Precizări privind desfășurarea testului de laborator:

- Testul va conține 3 subiecte, iar un subiect poate să aibă mai multe cerințe.
- Rezolvarea unui subiect se va realiza într-un singur fișier sursă Python (.py), indiferent de numărul de cerințe, care va fi încărcat/atașat ca răspuns pentru subiectul respectiv.
- Numele fișierului sursă Python trebuie să respecte următorul șablon: *grupa_nume_prenume_subiect.py*. De exemplu, un student cu numele Popescu Ion Mihai din grupa 131 trebuie să denumească fișierul care conține rezolvarea primului subiect astfel: 131_Popescu_Ion_Mihai_1.py.
- La începutul fiecărui fișier sursă Python se vor scrie, sub forma unor comentarii, numele și prenumele studentului, precum și grupa sa. Dacă un student nu reușește să rezolve deloc un anumit subiect, totuși va trebui să încarce/atașeze un fișier sursă Python cu informațiile menționate anterior!
- Toate rezolvările (fișierele sursă Python) trimise de către studenți vor fi verificate din punct de vedere al similarității folosind un software specializat, iar eventualele fraude vor fi sancționate conform Regulamentului de etică și profesionalism al FMI (http://old.fmi.unibuc.ro/ro/pdf/2015/consiliu/Regulament_etica_FMI.pdf).
- Operațiile de sortare se vor efectua folosind funcții sau metode predefinite din limbajul Python.

Precizări privind desfășurarea examenului scris:

- Toate subiectele se vor rezolva folosind limbajul Python.
- Subiectul 1 este obligatoriu, iar dintre subiectele 2, 3 și 4 se vor rezolva CEL MULT DOUĂ, la alegere.
- Citirea datelor de intrare se va realiza de la tastatură, iar rezultatele vor fi afișate pe ecran.
- Se garantează faptul că datele de intrare sunt corecte.
- Operațiile de sortare se vor efectua folosind funcții sau metode predefinite din limbajul Python.
- Rezolvările subiectelor alese dintre subiectele 2, 3 și 4 trebuie să conțină:
  - o scurtă descriere a algoritmului și o argumentare a faptului că acesta se încadrează într-o anumită tehnică de programare;
  - în cazul problemelor rezolvate folosind metoda Greedy sau metoda programării dinamice se va argumenta corectitudinea criteriului de selecție sau a relațiilor de calcul;
  - în cazul subiectelor unde se precizează complexitatea maximă pe care trebuie să o aibă soluția, se va argumenta complexitatea soluției propuse și vor primi punctaj maxim doar soluțiile corecte care se încadrează în complexitatea cerută;
  - în fiecare program Python se va preciza, pe scurt, sub forma unor comentarii, semnificația variabilelor utilizate.
- Pentru subiectele 1 nu contează complexitățile soluțiilor propuse.
- Rezolvările corecte care nu respectă restricțiile indicate vor primi punctaje parțiale.
- Se acordă 1 punct din oficiu.

- Rezolvările tuturor subiectelor se vor scrie de mână, folosind pix/stilou cu culoarea pastei/cernelii albastră sau neagră. Pe fiecare pagina studentul își va scrie numele și grupa, iar paginile trebuie să fie numerotate.
- Înainte de expirarea timpului alocat examenului, toate paginile vor fi fotografiate/scanate clar, în ordinea corectă, și transformate într-un singur fișier PDF care va fi încărcat pe platforma MS Teams folosind un anumit formular.

- Numele fișierului PDF **trebuie să respecte șablonul** *grupa_nume_prenume.pdf*. De exemplu, un student cu numele Popescu Ion Mihai din grupa 131 trebuie să denumească fișierul care conține rezolvarea primului subiect astfel: *131_Popescu_Ion_Mihai.pdf*.

Subiectul 1 – limbajul Python – 3 p.

a) Scrieţi o funcţie **numere** care primeşte un număr variabil de numere naturale şi returnează informaţii despre numere sub forma unui dicţionar de perechi **{medie: lista de numere}**. *Lista de numere* este formată din numerele naturale primite ca parametru care au aceeaşi medie aritmetică a cifrelor, iar cheia *medie* este un număr real având valoarea respectivei medii aritmetice. Listele de numere nu conţin elemente duplicate şi sunt sortate în ordine descrescătoare.
De exemplu, pentru apelul **numere (82375, 201, 51, 73, 3456, 2855, 1021, 90, 153)** funcţia va returna dicţionarul {1.0: [1021, 201], 3.0: [153, 51], 4.5: [3456, 90], 5.0: [82375, 2855, 73]}. **(1.5 p)**

b) Înlocuiţi punctele de suspensie din instrucţiunea **`L = [...]`** cu o secvenţă de iniţializare (*list comprehension*) astfel încât, după executarea sa, lista **`L`** să conţină numerele naturale formate din exact 3 cifre aflate în ordine strict crescătoare. **(0.5 p.)**

c) Considerăm următoarea funcţie recursivă:

```
def f(lista, p, u):
    if u - p <= 1:
        return lista[p]
    k = (p + u) // 2
    if k % 2 == 0:
        return f(lista, p, k-2) + f(lista, k+2, u)
    else:
        return f(lista, p, k-1) + f(lista, k+1, u)
```

Determinaţi complexitatea funcţiei apelată pentru o listă **`L`** formată din **`n`** numere întregi astfel: **`f(L, 0, n-1)`. (1 p.)**

Subiectul 2 – metoda Greedy (3 p.)
Complexitatea maximă a soluției: $\mathcal{O}(mn)$

În vitrina magazinului *CheapLuxury* bijuteriile din aur sunt așezate pe $m \geq 3$ rânduri, iar pe fiecare rând sunt câte $n \geq 3$ bijuterii. Hoțul Gicuță vrea să spargă vitrina magazinului și să fure bijuteriile, dar, deoarece sistemul de alarmă al magazinului este foarte performant, el își dă seama că are nevoie de un plan bine pus la punct. În acest scop, Gicuță se gândește să fure de pe fiecare din cele m rânduri câte o singură bijuterie astfel încât valoarea totală a acestora să fie maximă și, fiind lacom, vrea ca valoarea fiecărei bijuterii să fie strict mai mare decât valoarea bijuteriei furate de pe rândul precedent.
Scrieți un program Python care citește de la tastatură două numere naturale nenule m și n, o matrice cu m linii și n coloane care conține pe linia i valorile bijuteriilor de pe rândul i exprimate în euro, după care afișează pe ecran, în forma indicată în exemplu, valoarea totală a bijuteriilor pe care trebuie să le fure Gicuță, precum și pozițiile acestora (rândurile sunt considerate ca fiind numerotate de la 1, la fel și pozițiile bijuteriilor în cadrul unui rând). Dacă nu există nicio modalitate de a fura bijuteriile conform restricțiilor indicate, atunci programul va afișa mesajul "Imposibil".

Exemplu:

Date de intrare	Date de ieșire
4 3 515.99 350.79 731.25 299.99 515.88 766.10 566.25 271.99 444.89 865.99 918.55 799.99	2351.47 1 2 2 2 3 1 4 2

Subiectul 3 – metoda Programării Dinamice (3 p.)
Complexitatea maximă a soluției: O(n²)

Pe o tablă de șah de dimensiuni $n \times n$ se află pioni albi și ture albe. Un cal negru urmează să fie așezat pe tabla de șah cu misiunea de a elimina de pe tablă pioni și ture. Pentru un pion eliminat calul adună 1 punct, iar pentru o tură 2 puncte. Calul trebuie așezat pe prima coloană a tablei și se poate deplasa către ultima coloană respectând regulile după care se deplasează un cal la șah, dar cu restricția de a se deplasa doar la dreapta, adică pe o coloană care are indicele mai mare decât cea pe care se află. Astfel, mișcările posibile ale calului sunt indicate cu gri în figura următoare:

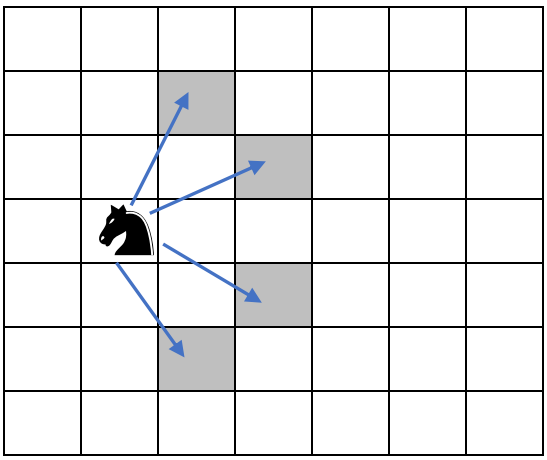

Tabla de șah se reprezintă ca o matrice de dimensiune $n \times n$ cu elementele în mulțimea {0,1,2} cu semnificația:

- 0 = pătrățică liberă
- 1 = pătrățică în care se află un pion alb
- 2 = pătrățică în care se află o tură albă

Scrieți un program Python care citește de la tastatură dimensiunea tablei $n > 4$, matricea care reprezintă tabla și care determină un traseu al calului pe care acesta adună un număr maxim de puncte (pornind de pe prima coloană și ajungând pe ultima). Se vor afișa punctajul și coordonatele celulelor din traseu. Liniile și coloanele matricei se presupun numerotate de la 1.

Intrare de la tastatură	Ieșire pe ecran (nu este unica)
6 0 0 1 0 0 0 0 0 2 **2** 0 1 **0** 0 0 2 0 **2** 0 0 **2** 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0	Punctaj maxim 6 1 3 3 4 4 2 6 3

Subiectul 4 – metoda Backtracking (3 p.)

a) Norocel a moştenit G kilograme de aur. Deoarece Norocel este şi mărinimos, el s-a gândit să împartă cele G kilograme de aur în n saci cu proprietatea că valoarea absolută a diferenţei dintre greutăţile oricăror doi saci să fie cel mult egală cu o valoare k şi apoi să doneze $n-1$ dintre sacii obţinuţi unor cazuri sociale. Ajutaţi-l pe Norocel să-şi împartă aurul conform dorinţelor sale scriind un program Python care să citească de la tastatură numerele G, n şi k, după care să afişeze toate modalităţile în care Norocel poate să-şi împartă aurul moştenit în saci. Dacă nu există nicio modalitate de împărţire a aurului care să respecte cerinţele lui Norocel, atunci programul va afişa mesajul "*Imposibil*".

Exemplu: Pentru $G = 10$, $n = 3$ şi $k = 4$ trebuie afişate următoarele 22 de variante pentru împărţirea aurului (nu neapărat în această ordine):

```
1 + 4 + 5
1 + 5 + 4
2 + 2 + 6
2 + 3 + 5
2 + 4 + 4
2 + 5 + 3
2 + 6 + 2
3 + 2 + 5
3 + 3 + 4
3 + 4 + 3
3 + 5 + 2
3 + 6 + 1
4 + 1 + 5
4 + 2 + 4
4 + 3 + 3
4 + 4 + 2
4 + 5 + 1
5 + 1 + 4
5 + 2 + 3
5 + 3 + 2
5 + 4 + 1
6 + 2 + 2
```

b) Modificaţi o singură instrucţiune din program astfel încât să fie afişate doar soluţiile care conţin un sac cu greutatea de 1 kg. Pentru exemplul anterior, aceste soluţii sunt cele scrise cu roşu. **(0.5 p.)**