MODALITATEA DE DESFĂȘURARE A EXAMENULUI LA DISCIPLINA "PROGRAMAREA ALGORITMILOR" DIN SESIUNEA DE RESTANȚE 31 MAI – 6 IUNIE 2021

- Examenul la disciplina "Programarea algoritmilor" se va desfășura în ziua de 03.06.2021, între orele 9^{00} și 14^{00}, astfel:
  - $09^{00} – 9^{15}$: efectuarea prezenței studenților la testul de laborator
  - $09^{15} – 10^{45}$: desfășurarea testului de laborator
  - $10^{45} – 11^{00}$: verificarea faptului că sursele trimise de către studenți au fost salvate pe platformă
  - $11^{00} – 11^{30}$: pauza
  - $11^{30} – 11^{45}$: efectuarea prezenței studenților la examenul scris
  - $11^{45} – 13^{45}$: desfășurarea examenului scris
  - $13^{45} – 14^{00}$: 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, următoarele informații: numele și prenumele studentului, grupa sa și enunțul subiectului rezolvat în fișierul sursă respectiv. 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).

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) Scrieți o funcție *vocale* care primește un număr variabil de cuvinte formate doar din litere mici și grupează aceste cuvinte după numărul de vocale, returnând un dicționar de perechi de forma {*numar_vocale* : *lista_de_cuvinte*}, unde cheia *numar_vocale* reprezintă un număr, iar *lista_de_cuvinte* este o listă nevidă care conține toate cuvintele primite ca parametru care au numărul de vocale egal cu *numar_vocale*.
De exemplu, pentru apelul *vocale*("acasa", "masa", "scaun", "oaie", "oare") funcția trebuie să furnizeze dicționarul {3: ["acasa", "oare"], 2: ["masa", "scaun"], 4: ["oaie"]}, deoarece cuvintele "acasa" și "oare" au 3 vocale, cuvintele "masa" și "scaun" au 2 vocale, iar cuvântul "oaie" are 4 vocale. **(1.5 p.)**

b) Știind că *lista_cuvinte* este o listă cu elementele cuvinte (de tip *str*), înlocuiți punctele de suspensie din instrucțiunea *lista_rez* = […] cu o secvență de inițializare (*list comprehension*) astfel încât, după executarea sa, lista *lista_rez* să conțină cuvintele din *lista_cuvinte* care încep și se termină cu aceeași literă. De exemplu, dacă *lista_cuvinte* = ["acasa", "masa", "este", "scaun"] atunci *lista_rez* va avea ca elemente cuvintele "acasa" și "este" **(0.5 p.)**

c) Considerăm următoarea funcție recursivă (se presupune importat modulul `random`):

```
def f(v, p, u):
    if u == p:
        return v[p]
    else:
        m = (p + u) // 2
        x = random.randint(0,2)
        if x==1:
            return f(v, m + 1, u)
        else:
            return f(v, p, m)
```

Determinați complexitatea funcției 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: $\mathcal{O}(n \log_2 n)$

Gigel tocmai a învățat la școală înmulțirea numerelor întregi. Pentru a-l ajuta pe Gigel să-și fixeze cunoștințele proaspăt dobândite, inclusiv cele algoritmice, bunicul său a scris numere întregi pe n cartonașe ($1 \leq n \leq 100000$), iar Gigel trebuie să selecteze o parte dintre ele (cel puțin unul și inclusiv toate!) astfel încât produsul numerelor scrise pe cartonașele respective să fie maxim. Scrieți un program Python care citește de la tastatură valorile scrise pe cele n cartonașe și afișează pe ecran cel mai mare produs pe care îl poate obține Gigel, precum și valorile cartonașelor selectate de el.

Exemplu:
Dacă valorile scrise pe cartonașe sunt $-10, -1, 10, 7, -10, -2, 4, -1, 3$, atunci produsul maxim pe care îl poate obține Gigel este 168000, folosind cartonașele -10, $10, 7, -10, -2, 4, -1, 3$.

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

O furnicuță vrea să se plimbe pe un teren reprezentat printr-un tablou cu m linii și n coloane, elementele tabloului fiind numere naturale, reprezentând înălțimi (altitudini). Furnicuța este mică și nu se poate deplasa decât câte un pas la **dreapta** sau un pas în **jos**, în celula alăturată celulei în care se află. De asemenea, dacă înălțimea la care se află (valoarea din celula curentă) este x, ea se poate deplasa doar în celule alăturate cu înălțime cu valoarea $x-1$ sau $x+1$, altfel efortul ei ar fi prea mare (și nici nu vrea să rămână după un pas la aceeași înălțime). Furnicuța ar vrea însă să se plimbe cât mai mult și poate începe traseul din orice punct al terenului, așa că vă roagă pe voi să scrieți un program care, date m, n și elementele tabloului, să afișeze un traseu de lungime maximă pe care îl poate parcurge; lungimea traseului este dată de numărul de celule de pe traseu. Se vor afișa coordonatele celulelor din traseu (liniile și coloanele tabloului se presupun numerotate de la 1).

Intrare de la tastatură	Ieșire pe ecran (soluția nu este unică)
5 5 9 1 2 4 5 8 7 8 9 11 7 6 7 1 12 3 8 6 7 2 1 9 8 7 6	1 1 2 1 2 2 2 3 3 3 4 3 4 4
4 6 1 2 1 1 4 5 0 2 3 4 7 2 5 1 6 3 4 4 5 6 7 8 7 5	2 2 2 3 2 4 3 4 3 5

Subiectul 4 – metoda Backtracking (3 p.)

a) Loteria Română a extins jocul *Loto 6/49* la *Loto 6/n*, unde n este un număr natural cuprins între 13 şi 100. Jucătorii împătimiţi, care au observat cu atenţie extragerile efectuate de-a lungul mai multor ani, au denumit *extrageri norocoase* acele extrageri formate din 6 numere care nu conţin numere consecutive, iar numerele nu au toate aceeaşi paritate. Scrieţi un program Python care să citească de la tastatură numărul natural n, după care să afişeze toate extragerile norocoase. **(2.5 p.)**

Exemplu:
Pentru $n = 14$ există 70 de extrageri norocoase:

1,3,5,7,9,12	1,3,6,8,10,14	1,4,6,9,11,13	2,4,6,9,11,14
1,3,5,7,9,14	1,3,6,8,11,13	1,4,6,9,11,14	2,4,6,9,12,14
1,3,5,7,10,12	1,3,6,8,11,14	1,4,6,9,12,14	2,4,7,9,11,13
1,3,5,7,10,13	1,3,6,8,12,14	1,4,6,10,12,14	2,4,7,9,11,14
1,3,5,7,10,14	1,3,6,9,11,13	1,4,7,9,11,13	2,4,7,9,12,14
1,3,5,7,11,14	1,3,6,9,11,14	1,4,7,9,11,14	2,4,7,10,12,14
1,3,5,7,12,14	1,3,6,9,12,14	1,4,7,9,12,14	2,5,7,9,11,13
1,3,5,8,10,12	1,3,6,10,12,14	1,4,7,10,12,14	2,5,7,9,11,14
1,3,5,8,10,13	1,3,7,9,11,14	1,4,8,10,12,14	2,5,7,9,12,14
1,3,5,8,10,14	1,3,7,9,12,14	1,5,7,9,11,14	2,5,7,10,12,14
1,3,5,8,11,13	1,3,7,10,12,14	1,5,7,9,12,14	2,5,8,10,12,14
1,3,5,8,11,14	1,3,8,10,12,14	1,5,7,10,12,14	3,5,7,9,11,14
1,3,5,8,12,14	1,4,6,8,10,12	1,5,8,10,12,14	3,5,7,9,12,14
1,3,5,9,11,14	1,4,6,8,10,13	1,6,8,10,12,14	3,5,7,10,12,14
1,3,5,9,12,14	1,4,6,8,10,14	2,4,6,8,10,13	3,5,8,10,12,14
1,3,5,10,12,14	1,4,6,8,11,13	2,4,6,8,11,13	3,6,8,10,12,14
1,3,6,8,10,12	1,4,6,8,11,14	2,4,6,8,11,14	
1,3,6,8,10,13	1,4,6,8,12,14	2,4,6,9,11,13	

b) Precizaţi cum ar trebui modificată o singură instrucţiune din program astfel încât să fie afişate doar *extragerile meganorocoase*, adică extragerile norocoase care nu conţin numărul 13. Pentru exemplul anterior, aceste soluţii sunt cele scrise cu roşu. **(0.5 p.)**