

1

Se consideră (în python) variabila:

`l=[22,7,10,3,15,100]`

Care dintre următoarele afirmații sunt adevărate? (pentru fiecare afirmație se consideră valoarea inițială a variabilei `l` ca fiind cea din întrebare și nu valoarea schimbată de la vreun alt subpunct (15 puncte)

- ☐ Nu există nicio funcție `f` astfel încât `l.sort(key=f)` să schimbe lista `l` în `[10, 100, 15, 22, 3, 7]`
- ☐ Putem defini o funcție `expr(x)` astfel încât folosind expresia `l2=[expr(x) for x in l]` să obținem în `l2` lista `[10,100]`
- ☐ Există un sir `s`, astfel încât expresia `s.join(l)` să aibă ca rezultat `"22+7+10+3+15+100"`, fără a realiza alte operații asupra lui `l` în prealabil.
- ☐ Fie o listă `l1`. Expresia `sorted([(x,y) for x in l+l for y in l1])[:-1]==sorted([(y,x) for x in l1 for y in 2*l], reverse=True)` este adevărată pentru orice listă `l1`, nevidă, de numere. ✓
- ☐ Nu există nicio funcție `cond(x)` și nicio funcție `expr(x)` astfel încât în urma atribuirii `l1=[expr(x) for x in l if cond(x)]`, `l1` să fie `[100, 15, 10]`
- ☐ Pentru o listă `l`, de numere, nu neapărat cea din enunț expresia `l.sort()` poate modifica variabila `l`, pe când expresia `sorted(l)` nu îl modifică niciodată pe `l`. ✓

2

Vom considera reprezentarea unei matrici de dimensiune $n \times m$ ca fiind o listă de n liste diferite ca obiecte (nu avem două referințe de liste care să fie egale). Fiecare din cele n liste are m elemente.

Considerăm matricea `matr=[[1,4,8],[2,8,0],[5,1,9],[0,3,0]]`

Care dintre următoarele fraze sunt adevărate? (15 puncte)

- ☐ Pentru a obține numărul de zerouri din matricea `matr` putem scrie `matr.count(0)`
- ☐ `[[0]*n]*m` creează o matrice validă cu n linii și m coloane și toate elementele nule.
- ☐ Să presupunem că `mn` este valoarea minimă dintre numărul de linii și numărul de coloane ale matricii `matr`, atunci expresia `[[matr[j][i] if matr[j][i]>matr[i][j] else matr[i][j] for i in range(mn)] for j in range(mn)]` va rezulta întotdeauna într-o matrice pătratică, simetrică față de diagonală principală. ✓
- ☐ Dacă executăm instrucțiunea `matr.append(matr[0])`, iar apoi calculăm `matr[0][0]=max([matr[i+1][i] for i in range(len(matr[0]))])` și `matr[-1][0]=max([matr[i] for i in range(len(matr))])`, atunci prima și ultima linie din matricea `matr` vor fi mereu egale. ✓
- ☐ Putem verifica egalitatea (adică au aceleași dimensiuni și aceleași elemente) a două matrici de numere, `m1` și `m2` prin `m1==m2` ✓

3

Care dintre următoarele fraze sunt adevărate? (15 puncte)

- ☐ Pentru orice listă l , nevidă, există două perechi de valori (i, j) astfel încât $l[:i] + l[:j]$ să fie o listă palindrom (simetrică față de mijlocul său) ✓
- ☐ Considerăm o listă l (de numere) pentru care facem operațiile: $\text{copie}_l = \text{list}(l)$, $l^* = n$, unde n e număr natural nenul și apoi $l.\text{sort}()$. Există un număr k astfel încât $l[:k] == \text{copie}_l$ ✓
- ☐ Fie o listă l nevidă și o variabilă $l2 = l[a:b:c]$. Dacă $l2$ e nevidă atunci putem spune cu certitudine că $a < b$.
- ☐ Fie o listă l nevidă și o variabilă $l2 = l[a:b:c]$. Dacă $l2$ are două elemente, în această ordine: $e1$ și $e2$, și c este negativ, atunci putem spune cu certitudine că, în lista l , $e1$ se află după $e2$. ✓
- ☐ Pentru o listă $l2 = l[a:b:c]$, cu c pozitiv, nenul, avem în $l2$ garantat parte întreagă din $|b-a|/c$ elemente, unde notația $|x|$ înseamnă modulul (valoarea absolută) lui x

4

Care din următoarele fraze sunt adevărate? (15 puncte)

- ☐ Pentru șirul $\text{sir} = \text{"cotcodac si cip-cip-cip-cirip"}$ exista 6 numere: a, b, c, d, e, f astfel încât $\text{sir}[a:b:c] + \text{sir}[d:e:f]$ să fie egal cu "doctor-cip" ✓
- ☐ Fie un șir de caractere nevid, s . Atunci putem spune că oricare ar fi i nenul, întreg, avem că $s[:i]$ e mereu egal cu $s[::i]$
- ☐ Pentru șirul $s = \text{"abcd"}$, dacă scriem $s[0] = s[1]$, s va avea valoarea "aacd"
- ☐ Pentru șirul $\text{sir} = \text{"pisica prinde soareci"}$ exista 6 numere: a, b, c, d, e, f astfel încât $\text{sir}[a:b:c] + \text{sir}[d:e:f]$ să fie egal cu "pisici aici"

Se dă clasa de mai jos. Care dintre fraze sunt adevărate?

Observație: sunt 2 exerciții care folosesc aceeași clasă, deci nu e nevoie să recitiți codul dacă ați mai făcut un exercițiu cu clasa curentă. (15 puncte)

```

77 class Pisica:
78     numar=0
79     soareciPrinsi=[]
80
81     def __init__(self, _nume, _varsta=0, _greutate=0):
82         self.nume=_nume
83         self.varsta=_varsta
84         self.greutate=_greutate
85         self.nrSoareci=0
86         self.__class__.numar+=1
87
88     def prindeSoareci(self, numeSoarece, greutateSoarece=0):
89         self.nrSoareci+=1
90         self.greutate+=greutateSoarece
91         self.soareciPrinsi.append(numeSoarece)
92
93     def __str__(self):
94         return "Pisica "+self.nume
95
96     def __repr__(self):
97         return "{}, {} a prins {} soareci".format(self.nume, self.varsta, self.nrSoareci)
98
99     @classmethod
100     def reseteazaSoareci(cls):
101         cls.soareciPrinsi=[]

```

- ☐ Creăm un obiect `p1=Pisica("Mitzi", 10)`. În acest caz, în constructor, în parametrul `self` va intra valoarea Mitzi, în `_nume` 10, iar ceilalți 2 parametri vor avea valorile implicite 0 și 0.
- ☐ Constructorul clasei se poate apela fără argumente
- ☐ Numărul maxim de argumente pentru constructor este 4.
- ☐ Pentru `p1=Pisica("Mitzi", _greutate=10)` dacă afișăm, fără modificări intermediare, proprietatea `varsta`, aceasta va fi 0. ✓
- ☐ pentru `p1=Pisica("Mitzi", 10)` un apel corect de metodă este `p1.prindeSoareci(p1,"Chitzi",0.3)`

Se dă clasa de mai jos. Care dintre fraze sunt adevărate?

Observație: sunt 2 exerciții care folosesc aceeași clasă, deci nu e nevoie să recitiți codul dacă ați mai făcut un exercițiu cu clasa curentă. (15 puncte)

```

77 class Pisica:
78     numar=0
79     soareciPrinsi=[]
80
81     def __init__(self, _nume, _varsta=0, _greutate=0):
82         self.nume=_nume
83         self.varsta=_varsta
84         self.greutate=_greutate
85         self.nrSoareci=0
86         self.__class__.numar+=1
87
88     def prindeSoareci(self, numeSoarece, greutateSoarece=0):
89         self.nrSoareci+=1
90         self.greutate+=greutateSoarece
91         self.soareciPrinsi.append(numeSoarece)
92
93     def __str__(self):
94         return "Pisica "+self.nume
95
96     def __repr__(self):
97         return "{}, {} a prins {} soareci".format(self.nume, self.varsta, self.nrSoareci)
98
99     @classmethod
100     def reseteazaSoareci(cls):
101         cls.soareciPrinsi=[]

```

- ☐ pentru `p1=Pisica("Mitzi", 5, 10)`, presupunând că nu s-au făcut modificări asupra variabilei, instrucțiunea `print(p1)` va afișa **Pisica Mitzi** ✓
- ☐ pentru `p1=Pisica("Mitzi", 5, 10)`, presupunând că nu s-au făcut modificări asupra variabilei, instrucțiunea `print(p1)` va afișa **Mitzi, 5 a prins 0 soareci**
- ☐ Dacă avem în program printre alte instrucțiuni, codul `p2=Pisica("Pisi", 1, 5); p2.prindeSoareci("Chitzi")`, imediat după aceste instrucțiuni avem garanția că `p2.soareciPrinsi` este lista `["Chitzi"]`
- ☐ Dacă avem în program printre alte instrucțiuni, codul `p2=Pisica("Pisi", 1, 5); p2.soareciPrinsi=[]; p2.prindeSoareci("Chitzi")`, iar după aceste instrucțiuni apelăm `Pisica.reseteazaSoareci()`, atunci în `p2.soareciPrinsi` nu vom mai avea elemente.
- ☐ Dacă încercăm să comparăm două instanțe ale clasei `Pisica`, `p1=Pisica("Mitzi", 5, 10)` și `p2=Pisica("Pisi", 1, 5)` prin `p1<p2`, se va arunca o eroare, deoarece nu este definită metoda `__lt__()` în cadrul clasei `Pisica` ✓

7

Care dintre următoarele afirmații sunt adevărate? (15 puncte)

- ☐ Pentru o matrice `m`, salvată ca listă de liste, putem evalua dacă în interiorul său se află un element nul prin expresia `all([all(linie) for linie in m])`. Dacă expresia e `True`, înseamnă că nu avem elemente nule, și `False` în caz contrar. ✓
- ☐ Pentru orice listă `L` nevidă, conținând valori ce pot fi evaluate ca booleene, expresia `not any([not x for x in L]) == all(L)` ✓
- ☐ Pentru o listă oarecare `L`, dacă e adevărat `all(L)` atunci sigur e adevărat și `any(L)` ✓
- ☐ Fie o variabilă `sirDat` care conține un șir de caractere și un alt șir mai mic, numit separator. Expresia `all(sirDat.split(separator))` este adevărată dacă și numai dacă șirul separator nu apare în mod consecutiv ca subșir în șirul `sirDat` (de exemplu, șirul separator `"#"` apare în mod consecutiv în `"a##*bc"` dar nu și în `"a#*bc#*def"`)
- ☐ Considerăm o matrice `m`, de numere, reprezentată ca listă de liste. Presupunând că am importat funcția `reduce` din `functools`, și presupunând că avem într-o variabilă `x` o valoare numerică, atunci `reduce(lambda x,b:x+b, [lin.count(x) for lin in m])` oferă numărul de apariții al valorii lui `x` în matricea `m` ✓

8

Care din următoarele afirmații sunt adevărate (15 puncte)

- ☐ Un bloc `try` poate avea asociate mai multe blocuri `except` ✓
- ☐ Pentru o funcție definită cu `def f(x:int, y:str) -> int`, apelul `f(2.5, 10)` va arunca o eroare
- ☐ Există o funcție `f` astfel încât `min([10,417,22,0,99], key=f)` pentru lista dată să returneze 417 ✓
- ☐ Pentru o matrice `m` definită ca listă de liste, `max(max(m))` returnează cel mai mare element din matrice
- ☐ Pentru o matrice `m` definită ca listă de liste, expresia `[(lin, linie.index(0)) for lin, linie in enumerate(m)]` calculează o listă cu toate coordonatele (linie,coloană) în care se află elementul 0 în matrice.

9

Care dintre următoarele afirmații sunt adevărate, cu privire la tehnicile de căutare?
Considerăm o soluție ca fiind un drum de la nodul start la un nod scop. (15 puncte)

- ☐ Algoritmul BreadthFirst va întoarce întotdeauna o soluție, indiferent care e nodul start și care sunt nodurile scop din graf.
- ☐ Într-un graf neconex, pot exista noduri scop pentru care nu avem un drum soluție (pornind de la nodul start) ✓
- ☐ Într-un graf orientat, pe care aplicăm o problemă de căutare, avem măcar un drum soluție **dacă și numai dacă** nodul start și minim un nod scop se găsesc în aceeași componentă tare conexă
- ☐ Dacă am folosi algoritmul BF ca să obținem toate soluțiile pentru o problemă de căutare, ultima soluție returnată ar fi de lungime maximă ✓
- ☐ Dacă nodul start este nod izolat în graf, mulțimea soluțiilor este vidă, indiferent care este mulțimea nodurilor scop din graf.

10

Considerăm un graf orientat cu $n > 2$ noduri numerotate cu numere de k cifre, $k \geq 2$ (numărul va fi informația nodului), unde prima cifră nu poate fi 0. Vom nota informația unui nod cu $\text{info}(\text{nod})$. Considerând că mulțimea arcelor A este mulțimea tuturor perechilor de noduri (i, j) (arcul este de la i la j) cu proprietatea că între $\text{info}(i)$ și $\text{info}(j)$ diferența e de o singură cifră (de exemplu, pentru $k=3$, există arc de la 247 la 347, deoarece între ele diferă doar prima cifră). Vom considera scopurile ca fiind nodurile cu numere având toate cifrele egale (de exemplu, pentru $k=3$, 777 este un nod scop). Considerăm lungimea unui drum ca fiind numărul de muchii din drum. Care dintre următoarele afirmații sunt adevărate? (15 puncte)

- ☐ Pentru o valoare k oarecare, numărul de succesori ai unui nod este mereu 9^k (9 la puterea k)
- ☐ Pentru orice k și orice nod din graf ales ca fiind nodul start, lungimea celui mai scurt drum este $k-1$
- ☐ Vom presupune ordinea de generare a succesorilor următoarea: cifrele se schimbă de la stânga la dreapta și valorile cu care se schimbă se iau în ordine crescătoare. De exemplu, pentru $k=3$ și nodul 205, succesorii în ordine ar fi: 105, 305, 405, ..., 905, 215, 225, ..., 295, 201, 202, 203, 204, 206, 207, 208, 209. Atunci pentru orice k , dacă setăm nodul de start 10^{k-1} (10 la puterea $k-1$) drumul returnat de BreadthFirst este același cu drumul returnat de DepthFirst
- ☐ Pentru un k oarecare, dacă alegem drept nod start un nod scop, lungimea primului drum returnat de BreadthFirst este mereu k .
- ☐ Pentru un nod start cu $k-1$ cifre identice și o cifră diferită, primul drum returnat de BreadthFirst are în mod cert lungime 1, indiferent de ordinea evaluării succesorilor. ✓
- ☐ Pentru un nod start cu $k-1$ cifre identice și o cifră diferită, primul drum returnat de DepthFirst are în mod cert lungime 1, indiferent de ordinea evaluării succesorilor.