

Programiranje 1 - 2. pisni izpit

16. 8. 2022

Ime in priimek: _____

Vpisna številka: _____

Število preostalih Jollyjev: _____

Točke: _____ / 50

- Za reševanje je na voljo 60 minut.

1. NALOGA [12 T]

Sestavi izraze (ne stavke), ki imajo naslednje vrednosti: (če ne znaš sestaviti izraza, lahko za polovično število točk napišeš zaporedje ukazov, ki poskrbijo, da je v spremenljivki `rezultat` potem ustrezna vrednost)

- (1) [3 T] Logično vrednost, ki pove, ali največji element tabele števil `stevila` presega vrednost v spremenljivki `naj_velikost`.
- (2) [3 T] Naključni element tabele `tabela`.
- (3) [3 T] Za podana slovarja `slovar1` in `slovar2` število ključev, ki se pojavijo v obeh slovarjih.
- (4) [3 T] Vsoto vseh števk v nizu `niz`. Za `niz = "I FEEL 570v3n1j4"` naj ima torej vrednost 20, za `niz = "3.14"` pa 8.

2. NALOGA [6 T]

Dana je funkcija f:

```
def f(s, t):  
    if s == '':  
        return True  
    if t == '':  
        return False  
    if s[0] == t[0]:  
        return f(s[1:], t[1:])  
    return f(s, t[1:])
```

Kakšne so vrednosti naslednjih izrazov:

- a) [1 T] f('', ''): _____
- b) [1 T] f('beseda', ''): _____
- c) [1 T] f('a', 'tabla'): _____
- d) [1 T] f('mentor', 'monter'): _____
- e) [1 T] f('osa', 'postaja'): _____
- f) [1 T] f('med', 'čebela'): _____

3. NALOGA [4 T]

Kakšne bodo tabele tab1, tab2, tab3, tab4 po tem, ko izvedemo naslednji program?

```
def funkcija2(tab):  
    nova = tab  
    for i in range(len(nova)):  
        if nova[i] < 0:  
            nova[i] = 0  
    return nova  
  
tab1 = [3, -2, 12, 5, -1, 10]  
tab2 = tab1  
tab3 = tab1[1:4]  
tab4 = funkcija2(tab1)  
print(tab1, tab2, tab3, tab4)
```

tab1: _____

tab2: _____

tab3: _____

tab4: _____

Sestavi naslednje preproste **funkcije**.

-
- A diagram of a periodic wave, specifically a sawtooth or triangular wave. The wave consists of a series of connected line segments forming a continuous zigzag pattern. The vertical distance from the horizontal baseline to the peaks is labeled 'd', representing the amplitude. The angle between the horizontal baseline and the upward-sloping segments is labeled 'a'.

5. NALOGA [8 T]

Dan je razred Razred.

```
class Razred:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    @property
    def x(self):
        return self._x

    @x.setter
    def x(self, novi):
        self._x = novi

    @property
    def y(self):
        return self._y

    @y.setter
    def y(self, novi):
        self._y = novi

    def __str__(self):
        return f"({self.x}, {self.y})"

    def __repr__(self):
        return f"Razred({self.x}, {self.y})"
```

[5 T] Kaj bodo izpisale vrstice (a), (b), (c) in (d) če v konzolo vpišemo naslednje stavke:

```
>>> objekt = Razred(10, -20)
>>> print(objekt)                                # (a)
>>> objekt                                         # (b)
>>> tab1 = [(1, 2), (-3, 1), (-2, -2), (0, 0), (-2, 1), (1, 1)]
>>> tab2 = [Razred(xi, yi) for xi, yi in tab1]
>>> print('[' + ', '.join(str(x) for x in tab2) + ']')    # (c)
>>> tab2.sort(key=lambda el: el.x**2 + el.y**2)
>>> tab2                                             # (d)
```

(a) >>> print(objekt):

(b) >>> objekt

(c) >>> print('[' + ', '.join(str(x) for x in tab2) + ']')

(d) >>> tab2

[3 T] Tabele objektov lahko urejamo s klicem metode `sort` ali funkcije `sorted` brez podajanja argumenta `key`, če definiramo, kako naj Python te objekte med seboj primerja po velikosti. Definiraj ustrezno (magično) metodo razredu `Razred`, da bo klic `tab2.sort()` tabelo uredil enako kot vrstica (d) iz prvega dela naloge, torej po oddaljenosti točke od izhodišča.

6. NALOGA [5 T]

Podano imamo tabelo kovancev različnih vrednosti. Zanima nas, kakšna je največja vsota, ki jo lahko nabereмо, če ne smemo pobrati dveh zaporednih kovancev. Dopolni funkcijo `najvecja_vsota(i, tab)`, da bo funkcija `najvec` delovala kot kaže zgled.

Zgled:

```
>>> najvec([2, 4, 1, 5, 2, 7, 8, 3, 1, 6])
25
>>> najvec([])
0
>>> najvec([10])
10
>>> najvec([4, 5])
5
```

```
def najvec(stevila):
    '''koliko je najvecja vsota dobljena iz
       tabele stevila, ce ne smemo izbrati dveh
       zaporednih elementov'''
    return najvecja_vsota(len(stevila), stevila)

def najvecja_vsota(i, tab):
    '''Vrne najvecjo vsoto iz prvih i elementov
       tabele tab, ce ne smemo vzeti dveh sosednjih'''
    if i == 0: # ni nobenih števil

        return _____

    if i == 1: # imamo le eno število

        return _____

    vzamemo = _____

    ne_vzamemo = _____

    return _____
```