1	2	3	4	5	6	7	8	9	10

Neprajem si zverejniť moje výsledky testu na webe:

_____

Záverečný test z Programovania (1) pre 1. ročník

1. Čo presne vypíše tento program?

```
def rek(n):
    vysl = 1
    for i in range(1, n):
        vysl += rek(i)
    return vysl

for i in range(6):
    print(i, rek(i))
```

riadky výpisu:

2. Zmenili sme metódu **`fd()`** pre korytnačku:

```
class MojaTurtle(turtle.Turtle):
    def fd(self, d):
        if d <= 10:
            super().fd(d)
        else:
            self.fd(d//2-5)
            self.pu()
            self.fd(10)
            self.pd()
            self.fd(d-d//2-5)
```

Zistite, aká je prejdená dĺžka čiar so spusteným perom pre tieto volania:

a) `MojaTurtle().fd(30)`
b) `MojaTurtle().fd(40)`
c) `MojaTurtle().fd(50)`

Uvedomte si, že táto funkcia je rekurzívna.

3. Zistite, čo vráti táto funkcia:

```
def test(pole):
    vysl, n = 0, len(pole)
    for i in range(n):
        for j in range(n):
            vysl += abs(pole[i][j]-pole[j][i])
    return vysl
```

pre volanie:

```
>>> test([[1, 2, 3], [2, 1, 0], [3, 2, 1]])
```

4. Zadefinovali sme triedu **`Zoznam`**, pomocou ktorej si vieme udržiavať zoznam svojich záväzkov. Trieda obsahuje tieto metódy:
   - **`pridaj(prvok)`**, ak sa tam takýto záväzok ešte nenachádza, pridá ho na koniec
   - **`vyhod(prvok)`**, ak sa tam takýto záväzok nachádzal, vyhodí ho
   - **`__contains__(prvok)`** vráti **`True`** alebo **`False`** podľa toho, či sa tam tento záväzok nachádzal
   - **`vypis()`** vypíše všetky záväzky v tvare `záväzok, záväzok, záväzok`

   Opravte všetky chyby (to, čo nie je chyba, neopravujte):

```
class Zoznam:
    pole = []

    def pridaj(self, prvok):
        if prvok in self:
            pole.insert(prvok)

    def vypis(self):
        print(*self.pole, sep=', ')

    def __contains__(self, prvok):
        return prvok in pole

    def vyhod(self, prvok):
        if prvok not in self:
            pole.delete(prvok)
```

5. V utriedenom poli máme tieto hodnoty:

```
pole = [10, 25, 30, 32, 43, 45, 51, 53, 58, 59, 63, 65, 70, 81, 82]
```

   Pomocou algoritmu binárneho vyhľadávania zisťujeme, či sa v ňom nachádza nejaká hodnota. Do základného cyklu algoritmu sme vložili príkaz na vypísanie indexu aj hodnoty stredného prvku v hľadanom úseku:

```
def hladaj(hodnota):
    zac, kon = 0, len(pole)-1
    while zac <= kon:
        stred = (zac + kon) // 2
        print(stred, pole[stred])   # pridali sme vypis
        if pole[stred] < hodnota:
            zac = stred + 1
        elif pole[stred] > hodnota:
            kon = stred - 1
        else:
            return True
    return False
```

   a) Vypíšte všetky tieto dvojice stredný index a hodnota, keď budeme hľadať číslo 63, t.j. volanie **`hladaj(63)`**.
   b) Vypíšte tieto dvojice pre hľadanú hodnotu 15, t.j. volanie **`hladaj(15)`**.

6. Napíšte funkciu **`pocet_riadkov(meno_suboru)`**, ktorá vráti počet riadkov zadaného súboru. Ak daný súbor neexistuje (nepodaril sa **`open()`**), funkcia vráti -1.

```
def pocet_riadkov(meno_suboru):
```

7. Napíšte funkciu **farba(retazec)**, ktorá z daného reťazca - mena farby v slovenčine vráti správny názov farby pre **tkinter**. Ak danú farbu nerozpozná, vráti farbu **pink**. Funkcia by mala akceptovať tieto slovenské mená farieb: **'biela'**, **'cervena'**, **'modra'**, **'zlta'**, **'zelena'**. Vo funkcii nepoužite príkaz **if**.

```
def farba(retazec):

    return ...
```

8. Mali sme napísať funkciu **viac(pole)**, ktorá vráti množinu tých prvkov poľa **pole**, ktoré sa v ňom vyskytujú viac ako raz. Napr.

```
>>> viac(['prvy', 2, (3, 4), 'dva', 3, 4, 'prvy', 3])
{'prvy', 3}
```

Lenže v našom programe, je niekoľko chýb. Opravte ich. Nepridávajte nové príkazy.

```
def viac(pole):
    mnozina1, mnozina2 = {}, {}
    for prvok1 in pole:
        for prvok2 in mnozina1:
            if prvok1 == prvok2:
                mnozina2.add(prvok2)
            else:
                mnozina1.add(prvok1)
    return mnozina1
```

9. Funkcia.

```
def pocitaj(a=0, b=0, c=0, d=0, e=0, f=0):
    return a - b + c - d + e - f
```

funguje pre maximálne 6 parametrov (postupne k výsledku pripočítava a odpočítava číselné parametre). Prerobte túto funkciu tak, aby mohla mať ľubovoľný počet parametrov. Napr..

```
>>> pocitaj(17, 15, 13, 11, 9, 7, 5, 3, 1)
9
```

Zapíšte

```
def pocitaj(.....................):

    return ...
```

10. Zapíšte funkciu **vyber()**, ktorá dostáva dva parametre: nejakú postupnosť hodnôt a postupnosť indexov. Funkcia vráti pole hodnôt (z prvého parametra), ktoré sú na príslušných indexoch (druhého parametra). Funkciu zapíšte generátorovou notáciou:

```
def vyber(pole, indexy):

    return ...
```

Napr.

```
>>> vyber('python', (1, 3, 5, 4, 2, 0))
['y', 'h', 'n', 'o', 't', 'p']
>>> vyber([11,12,13,14], (2,1,2,0,1,0,0,1))
[13, 12, 13, 11, 12, 11, 11, 12]
```