1	2	3	4	5	6	7	8	9	10

Neprajem si zverejniť moje výsledky testu na webe: _____

Test z Programovania pre 1. ročník

1. Zadefinujte ***rekurzívnu*** funkciu **`mocnina(n,k)`**, ktorá vypočíta **`n**k`** pre celé nezáporné **`k`** len pomocou násobenia a umocňovania na 2 (čo je opäť len násobením so samým sebou):
   - **`mocnina(n,0) = 1`**
   - **`mocnina(n,k) = mocnina(n,k//2) ** 2`** ... pre párne **`k`**
   - **`mocnina(n,k) = mocnina(n,k-1) * n`** ... pre nepárne **`k`**

```
def mocnina(n, k):
```

2. Zistite, koľko hviezdičiek sa vypíše pre volania **`rekurzia(10)`** a **`rekurzia(13)`**?

```
def rekurzia(n):
    if n < 2:
        print('*')
    else:
        rekurzia(n-1)
        rekurzia(n-2)
```

3. V triede **`Kniha`** si ukladáme informácie o knihách:

```
class Kniha:
    def __init__(self, autor, titul, cena):
        self.pole = [autor, titul, cena]

    def autor(self, zmen=None):
```

Dopíšte metódu **`autor()`** tak, aby volanie bez parametrov vrátilo autora knihy a volanie s jedným parametrom zmenilo autora, napr.

```
>>> k = Kniha('Doyle','Sherlock Holmes',11.5)
>>> k.autor()
'Doyle'
>>> k.autor('sir Arthur Conan Doyle')
>>> k.pole
['sir Arthur Conan Doyle', 'Sherlock Holmes', 11.5]
```

4. V triede **`Kniha`** si ukladáme informácie o knihách:

```
class Kniha:
    def __init__(self, autor, titul, cena):
        self.pole = [autor, titul, cena]

    def __getitem__(self, co):

        _________________

        return _________________

    def __setitem__(self, co, zmen):

        _________________

        _____________________ = zmen
```

Dopíšte metódy **`__getitem__()`** a **`__setitem__()`** tak, aby fungovalo (predpokladajte, že parameter **`co`** je jeden z reťazcov **`'autor'`**, **`'titul'`** alebo **`'cena'`**):

```
>>> k = Kniha('Doyle','Sherlock Holmes',11.5)
>>> k['titul']
'Sherlock Holmes'
>>> k['cena'] = 8.1
>>> k.pole
['Doyle', 'Sherlock Holmes', 8.1]
```

5. Dopíšte funkciu **`vyrob(dlzky)`**, ktorá vytvorí dvojrozmerné pole celých čísel tak, že ak parameter **`dlzky`** je pole celých čísel, tak tieto označujú dĺžky riadkov vytváraného dvojrozmerného poľa. Počet prvkov poľa **`dlzky`** označuje počet riadkov vytváraného poľa. Prvky vytváraného poľa pritom postupne zaplňte hodnotami od 1.

```
def vyrob(dlzky):
    pole = []
    _________________

    for d in dlzky:
        _________________

        _________________

    return pole

>>> pole = vyrob([3,2,4])
>>> vypis(pole)
   1    2    3
   4    5
   6    7    8    9
```

Funkciu **`vypis`** neprogramujte. Pri dopisovaní kódu do funkcie nemusíte dodržať naznačený počet riadkov programu.

6. Textový súbor **`'subor.txt'`** v každom riadku obsahuje niekoľko slov oddelených medzerami. Nasledovná funkcia by mala vytvoriť dvojrozmerné pole znakových reťazcov, ktoré bude v každom riadku obsahovať ako prvky slová z príslušného riadku súboru:

```
def urob(meno):
    with open(meno) as subor:
        vysledok = []
        while subor:
            riadok = subor.read()
            if riadok:
                return vysledok
            riadok = riadok.strip()
            riadok.append(vysledok)
```

Opravte všetky chyby.

7. Dané sú dve polia **pole1** a **pole2**, ktoré majú rovnaký počet prvkov. Vytvorte funkciu, ktorá z takýchto dvoch polí vytvorí a vráti nové asociatívne pole. V tomto asociatívnom poli sú kľúčmi prvky z prvého poľa a hodnotami sú príslušné prvky druhého poľa:

```
def urob(pole1, pole2):
```

```
>>> a = urob(['druh','vaha','vek'],['slon',1000,10])
>>> a
{'druh': 'slon', 'vek': 10, 'vaha': 1000}
```

8. Vytvorili sme pole informácií (asociatívnych polí) o zvieratách v zoo, napr.:

```
zoo = [{'druh':'slon','meno':'Bimbo'},{'druh':'opica','meno':'Milica'},...]
```

Napíšte funkciu **vsetky_mena(zoo)**, ktorá vráti množinu všetkých mien zvierat v **zoo**:

```
def vsetky_mena(zoo):
```

9. Zistite, čo sa vypíše:

```
>>> pole1 = [3,'sedem',3.14]
>>> pole2 = ['dog','cat','mouse','duck']
>>> pole3 = {'sedem':[3]*3,3.14:pole2,'cat':pole1}
>>> pole3[pole3[pole2[1]][2]][2]
```

10. Dopíšte funkciu **urob(m1, m2, m3)**, ktorá dostáva ako parametre 3 množiny. Výsledkom funkcie bude nová množina, ktorá obsahuje všetky také prvky, ktoré nie sú v **m1**, ale sú buď v **m2** alebo v **m3** (ale nie naraz v oboch).

```
def urob(m1, m2, m3):
    mnozina = _________________

    ___________________

    ___________________

    return mnozina
```

Pri dopisovaní kódu do funkcie nemusíte dodržať naznačený počet riadkov programu. Použitie, napr.

```
>>> urob({1,3,5,7},{1,2,3,4,5},{4,5,6,7})
{2, 6}
>>> urob(set(),{1,2,3,4,5},{4,5,6,7})
{1, 2, 3, 6, 7}
```