INFORMATIKA PRÓBA ALAPVIZSGA · INFOALAPVIZSGA.HU

INFORMATIKA ALAPVIZSGA

PRÓBA
GYAKORLATI VIZSGA

Programozás Pythonban
Nehéz „A” feladatsor

Ajánlott időtartam: 60 perc

Beadott dokumentumok	
Piszkozati pótlapok száma	

Értékelésre az alábbi állományokat adom be:	
Programozás Pythonban	

INFOALAPVIZSGA.HU
Gyakorolj a tanuljtovival.hu oldalon!

- A vizsgán három programot kell Python programozási nyelven elkészítened!
- A három feladat elkészítésére összesen 60 perc áll rendelkezésedre.
- A három részfeladat egymástól függetlenül is megoldható, és egyre nehezedik: oszd be az idődet, a harmadikra hagyj többet!
- A program üzeneteinek megfogalmazásában kövesd a példákat! Az üzenetek szövegét, írásjeleit és a számok helyét pontosan tartsd meg.
- Azokat a részeket, amiket a felhasználó gépel be, a mintában *dőlt betűvel és aláhúzással* találod!
- A felhasználó által megadott adatokat – *hacsak a feladat külön nem kéri* – nem kell ellenőrizni, feltételezheted, hogy ha pl. a felhasználónak egy számot kell beírnia, akkor számot is fog beírni.
- A harmadik feladathoz forrásfájl tartozik (**feladatok.txt**), ezt a programoddal azonos mappába másold!
- Munkád rendszeres mentésére ügyelj!

1. Gyakorlási sorozat

Írj programot **gyakorlasi_sorozat.py** néven! A program kérje be, hogy hány napról ad meg adatot a felhasználó, majd kérje be egyenként, hogy az egyes napokon hány percet gyakorolt (a minta szerint, a nap sorszámával).

A bekérés végén írd ki:

- hány nap alatt összesen hány percet gyakorolt a felhasználó;
- naponta átlagosan hány percet gyakorolt, egy tizedesjegyre kerekítve;
- hány **Tovi-nap** volt. Tovi-nap az a nap, amelyen a felhasználó **legalább 30 és legfeljebb 90** percet gyakorolt – a két feltételnek **egyszerre** kell teljesülnie.

Végül a program írja ki Tovi véleményét: ha az összes perc legalább 100 **és** a Tovi-napok száma legalább 2, akkor Tovi szerint remek hét volt; minden más esetben Tovi szerint lehet még fokozni.

1. példa futtatás:

```
Hány napról adsz meg adatot? 3
1. nap percei: 45
2. nap percei: 20
3. nap percei: 100
3 nap alatt összesen 165 percet gyakoroltál.
Naponta átlagosan 55.0 percet töltöttél az infoalapvizsga.hu feladataival.
Ebből 1 nap volt Tovi-nap.
Tovi szerint lehet még fokozni – holnap találkozunk az infoalapvizsga.hu-n!
```

2. példa futtatás:

```
Hány napról adsz meg adatot? 4
1. nap percei: 30
2. nap percei: 60
3. nap percei: 90
4. nap percei: 15
4 nap alatt összesen 195 percet gyakoroltál.
Naponta átlagosan 48.8 percet töltöttél az infoalapvizsga.hu feladataival.
Ebből 3 nap volt Tovi-nap.
Tovi szerint remek heted volt, így kell ezt csinálni!
```

2. Vizsgakód-generátor

Írj programot **vizsgakod.py** néven! Tovi minden vizsgázónak egyedi vizsgakódot készít. Definiálj egy függvényt **vizsgakod** néven, amelynek két paramétere van: a vizsgázó neve és születési éve. A függvény visszatérési értékként adja vissza a vizsgakódot, amely három, kötőjellel elválasztott részből áll:

- a név első három betűje **nagybetűvel** (feltételezheted, hogy a név legalább három betűs);
- a születési év **számjegyeinek összege**;
- a születési év **97-tel való osztási maradéka**.

Például Anna, 2010 esetén a kód: **ANN-3-70**.

A főprogram kérje be, hogy hány vizsgázó kódját kell elkészíteni, majd kérje be egyenként a vizsgázók nevét és születési évét! Minden vizsgázó után azonnal írd ki a kódját a függvény segítségével!

Végül írd ki, hogy a kódok közül hány **Tovi-kód**, azaz hány kódban legalább 10 a számjegyösszeg! A számoláshoz a függvény által visszaadott kódot használd, ne számold ki újra az összeget.

1. példa futtatás:

```
Hány vizsgázó kódját készítsük el? 2
Név: Anna
Születési év: 2010
Anna vizsgakódja: ANN-3-70
Név: Bence
Születési év: 2009
Bence vizsgakódja: BEN-11-69
A(z) 2 kódból 1 db Tovi-kód (a számjegyösszeg legalább 10). Sok sikert az alapvizsgán az infoalapvizsga.hu-tól!
```

2. példa futtatás:

```
Hány vizsgázó kódját készítsük el? 1
Név: Tovi
Születési év: 2024
Tovi vizsgakódja: TOV-8-84
A(z) 1 kódból 0 db Tovi-kód (a számjegyösszeg legalább 10). Sok sikert az alapvizsgán az infoalapvizsga.hu-tól!
```

3. Tovi feladatbankja

Írj programot **feladatbank.py** néven! Hozz létre egy osztályt **Feladat** néven. Az osztályt úgy hozd létre, hogy példányosításkor paraméterekből az alábbi adatokat tárolja el adattagként: cim, temakor, nehezseg, perc, megoldok.

Hozz létre egy listát **feladatok** néven, majd olvasd be a forrásfájlok között található **feladatok.txt** pontosvesszővel tagolt fájlt (az első sor tartalmazza a mezőneveket)! Minden sor az infoalapvizsga.hu feladatbankjának egy feladatát tartalmazza: a címét, a témakörét, a nehézségét (1 = könnyű, 2 = közepes, 3 = nehéz), az ajánlott időt percben, valamint hogy hány tanuló oldotta meg. Minden feladatra példányosítsd az osztályt, majd tárold el a feladatok nevű listába.

Algoritmus segítségével oldd meg az alábbi feladatokat:

1. Írd ki egy sorban, hogy hány feladat van a feladatbankban, ebből hány **Python** témakörű, és összesen hány percnyi gyakorlást tartalmaz!
2. A **statistics** modul **mean** függvényével számold ki, hogy egy feladatot átlagosan hány tanuló oldott meg, és írd ki egy tizedesjegyre kerekítve! Ezután írd ki egy sorban, vesszővel és szóközzel elválasztva azoknak a feladatoknak a címét, amelyeket az átlagnál **többen** oldottak meg (a fájlban lévő sorrendben)!
3. Az osztályon belül hozz létre egy tagfüggvényt **szint** néven, ami visszatérési értékként a nehézség szövegét adja vissza: 1 esetén „könnyű”, 2 esetén „közepes”, 3 esetén „nehéz”!
4. Kérj be a felhasználótól egy témakör nevét, majd írd ki, hogy az adott témakörben hány könnyű, hány közepes és hány nehéz feladat van! (Az előző feladatban lévő tagfüggvényt használd fel hozzá.) Ha a témakörből egyetlen feladat sincs, írd ki, hogy Tovi még nem készített ilyen feladatot!
5. Írd ki a **tovi-ajanlo.txt** fájlba a **nehéz** feladatokat a megoldók száma szerint **csökkenő** sorrendben! Minden feladat külön sorba kerüljön a minta szerinti formában (cím, kötőjel, megoldók száma és a „fő” szó).

Segítség a statistics modulhoz: a modul **mean** függvénye egy számokat tartalmazó lista átlagát adja vissza.

```
import statistics
pontok = [10, 20, 60]
atlag = statistics.mean(pontok) # 30
```

A megoldásban tehát előbb egy listába kell gyűjtened a feladatok megoldóinak számát, majd erre a listára hívod meg a függvényt.

1. példa futtatás:

```
Az infoalapvizsga.hu feladatbankjában 11 feladat van, ebből 6 Python témakörű, összesen 280 percnyi gyakorlás.
Egy feladatot átlagosan 26.8 tanuló oldott meg.
Átlag feletti megoldószámú feladatok: Változók és kiírás, Feltételek, For ciklus, Listák, HTML alapok
Írd be egy témakör nevét: HTML
HTML témakör: 1 könnyű, 1 közepes, 0 nehéz feladat.
```

2. példa futtatás:

```
Az infoalapvizsga.hu feladatbankjában 11 feladat van, ebből 6 Python témakörű, összesen 280 percnyi gyakorlás.
Egy feladatot átlagosan 26.8 tanuló oldott meg.
Átlag feletti megoldószámú feladatok: Változók és kiírás, Feltételek, For ciklus, Listák, HTML alapok
Írd be egy témakör nevét: SQL
Ebből a témakörből Tovi még nem készített feladatot!
```

tovi-ajanlo.txt fájl tartalma:

```
Fájlkezelés - 17 fő
Osztályok - 12 fő
Alhálózatok - 9 fő
```

A feladatok.txt fájl első sorai:

```
cim;temakor;nehezseg;perc;megoldok
Változók és kiírás;Python;1;10;48
Feltételek;Python;1;15;41
For ciklus;Python;2;20;35
...
```