INFORMATIKA PRÓBA ALAPVIZSGA · INFOALAPVIZSGA.HU

INFORMATIKA ALAPVIZSGA

PRÓBA
GYAKORLATI VIZSGA

Programozás Pythonban
Közepes „F” 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 (**szerverek.txt**), ezt a programoddal azonos mappába másold!
- Munkád rendszeres mentésére ügyelj!

1. Pontgyűjtés

Írj programot **pontgyujtes.py** néven! A program kérje be a próbavizsga három részének (Python, HTML, hálózat) pontszámát. Írd ki az összpontszámot, az átlagot (egy tizedesjegyre kerekítve) és a leggyengébb rész pontszámát!

A vizsgázó akkor felelt meg, ha **minden részen legalább 40 pontot** ért el **és** az összpontszáma **legalább 150**. Írd ki az eredményt a minta szerint!

*Tipp: a **min(a, b, c)** függvény három szám közül a legkisebbet adja.*

1. példa futtatás:

```
Python pontok: 60
HTML pontok: 55
Hálózat pontok: 48
Összesen 163 pont, átlag 54.3, a leggyengébb rész 48 pont.
Megfeleltél a próbavizsgán - Tovi gratulál!
```

2. példa futtatás:

```
Python pontok: 70
HTML pontok: 80
Hálózat pontok: 35
Összesen 185 pont, átlag 61.7, a leggyengébb rész 35 pont.
Még nem feleltél meg - Tovi szerint az infoalapvizsga.hu segít bepótolni.
```

2. Tükörszám

Írj programot **forditott_szam.py** néven! Definiálj egy függvényt **forditott** néven, amely a paraméterként kapott pozitív egész szám számjegyeit **fordított sorrendben** tartalmazó számot adja vissza! A megoldáshoz **ciklust és a % és // műveleteket** használd (ne alakítsd szöveggé a számot). Figyelj arra, hogy a végén lévő nullák eltűnnek: 1230 fordítottja 321.

A főprogram kérjen be egy számot, írja ki a fordítottját, a szám és a fordítottjának összegét, végül azt, hogy a szám tükörszám-e (azaz megegyezik-e a fordítottjával)!

1. példa futtatás:

```
Írj be egy pozitív egész számot: 1230
Visszafelé: 321
A szám és a fordítottjának összege: 1551
Ez nem tükörszám.
```

2. példa futtatás:

```
Írj be egy pozitív egész számot: 1221
Visszafelé: 1221
A szám és a fordítottjának összege: 2442
Ez egy tükörszám - Tovi mindkét irányból elolvasta.
```

3. Tovi szerverei

Írj programot **szerverek.py** néven! Hozz létre egy osztályt **Szerver** néven. Az osztályt úgy hozd létre, hogy példányosításkor paraméterekből az alábbi adatokat tárolja el adattagként: nev, cpu, ram.

Hozz létre egy listát **szerverek** néven, majd olvasd be a forrásfájlok között található **szerverek.txt** pontosvesszővel tagolt fájlt (az első sor tartalmazza a mezőneveket)! Minden sor a képzeletbeli Tovi-tanulóközpont egy szerverét tartalmazza: a nevét, a processzormagok számát és a memória méretét GB-ban. Minden sorra példányosítsd az osztályt, majd tárold el a szerverek nevű listába.

Algoritmus segítségével oldd meg az alábbi feladatokat:

1. Írd ki, hány szervert üzemeltet Tovi, összesen hány CPU-maggal és hány GB RAM-mal!
2. Az osztályon belül hozz létre egy tagfüggvényt **kategoria** néven, ami visszatérési értékként a szerver kategóriáját adja vissza: „nagy”, ha legalább 8 magja **és** legalább 32 GB RAM-ja van; különben „közepes”, ha legalább 4 magja van; minden más esetben „kicsi”.
3. Írd ki, hány nagy, közepes és kicsi szerver van!
4. Kérj be a felhasználótól egy szervernevet, majd írd ki a magok számát, a RAM méretét és a kategóriát a minta szerint! Ha nincs ilyen szerver, írd ki a minta szerinti üzenetet.
5. Írd ki a **nagy-szerverek.txt** fájlba a nagy szerverek nevét, soronként egyet!

1. példa futtatás:

```
Tovi 5 szervert üzemeltet, összesen 34 CPU-maggal és 224 GB RAM-mal.
Kategóriák: 2 nagy, 2 közepes, 1 kicsi.
Melyik szervert nézzük meg? tovi-teszt
tovi-teszt: 4 mag, 8 GB RAM - közepes szerver.
```

2. példa futtatás:

```
Tovi 5 szervert üzemeltet, összesen 34 CPU-maggal és 224 GB RAM-mal.
Kategóriák: 2 nagy, 2 közepes, 1 kicsi.
Melyik szervert nézzük meg? tovi-mail
Nincs ilyen nevű szerver.
```

nagy-szerverek.txt fájl tartalma:

```
tovi-db
tovi-ai
```

szerverek.txt fájl első sorai:

```
nev;cpu;ram
tovi-web;4;16
tovi-db;8;64
...
```