

Algoritmus dömping

Feladat 1

1.1. várható bekövetkezés

2 pont

Írj egy függvényt, ami a paraméterben megkapott két érték alapján (R_A , R_B) kiszámolja az „A” bekövetkezésének a valószínűségét a következőképpen:

$$E_A = \frac{1}{1 + 10^{\frac{R_B - R_A}{400}}}$$

E_A : Az A eredmény várható esélye (0 és 1 közötti szám, pl. $0.75 = 75\%$).

R_A , R_B : A két érték

400: Egy matematikai állandó, ami a "szórást" jelzi.

Tesztelés

Be: 10, 20

Ki: 0,485612815834001

1.2. az új értékek

2 pont

Az előző a paraméterben megadott függvény felhasználásával, számoljuk ki a lehetséges új értéket, a következő képlet alapján:

$$R_{uj} = R_{régi} + K (S_{tenyelges} - E_{várható})$$

K: A "K-tényező" (K-Factor). Ez határozza meg, mennyire változhat az érték egy alkalommal. (Legyen most fixen 31)

S: Az eredmény bekövetkezésének igaz értéke. (igaz esetén: 1.0, hamis esetén: 0).

E: előző feladat eredménye.

Az új érték negatívba nem mehet, ebben az esetben 0 értéket vesz fel. Az új érték kéttizedesjegyre kerekítve adja vissza.

Tesztelés

A: 10, B: 20, E_A : 0,485612815834001

A bekövetkezik, új értéke: 25,95

A nem következik be, új értéke: 0

Feladat 2

2.1 az értékek

2 pont

Generálj értékeket egy új **szöveges** tömböt, ami indexet is lementve tartalmazza a pontokat. A véletlenszámok [0,1000) között szerepeljenek kéttizedesjegy pontossággal, majd alul vonás után az indexe 1-től indexelve!

Tesztelés

Random osztály 709 kezdőértékkelű, 10 elemű tömb esetén

125,79_1

851,13_2

350,51_3

889,2_4

309,42_5

152,43_6

578,23_7

136,42_8
692,69_9
718,23_10

2.2 a kiírás

4 pont

A tömböt írasd ki a következő minta szerint:

Spoiler: Érdemes külön függvény(ekbe) szervezni a szétválasztást!

Tesztelés

1: 125,79 pont
2: 851,13 pont
3: 350,51 pont
4: 889,2 pont
5: 309,42 pont
6: 152,43 pont
7: 578,23 pont
8: 136,42 pont
9: 692,69 pont
10: 718,23 pont

Feladat 3 - a rendezés

5 pont

Írj egy eljárást, ami egy javított beillesztéses rendezés segítségével rendezi a tömb elemeit az értékek alapján, majd írasd ki!

Az algoritmus: A javított beillesztéses rendezés lényege, mint a kártyalapokat rakosgatnánk sorba a kezünkbe. Vagyis balról az első elemre úgy tekintünk, hogy jó helyen van, majd elindulunk a rákövetkező elemtől, ami „kiemelünk” és visszafelé haladva addig tologatjuk el az összes elemet, amíg nem igaz az, hogy az elejére értünk és tömb aktuális eleme nagyobb, mint a kiemelt.

Spoiler: megadom az algoritmust -3 pontért!

Tesztelés

1: 125,79 pont
8: 136,42 pont
6: 152,43 pont
5: 309,42 pont
3: 350,51 pont
7: 578,23 pont
9: 692,69 pont
10: 718,23 pont
2: 851,13 pont
4: 889,2 pont

Feladat 4 – a keresés

3 pont

Írj egy függvényt, ami a feltöltött és rendezett tömbben megkeresi a paraméterben kapott elem helyét!

Tesztelés (Az előző rendezett tömböt felhasználva)

ID: 6
Ki: 2

Feladat 5 – A szimuláció – ELO-Rating

7 pont



Az élő-pontrendszer (Elo Rating System) a modern online játékok (League of Legends, CS:GO, sakk, CodingGame) szíve-lelke. A Gyilkos számok 3. évadának 16. részében (A nagy meccs) pontosan ezt használják annak elemzésére, hogy egy esélytelenebbnek tűnő harcos hogyan győzhetett le egy profit. **Nem az számít, hogy nyersz-e, hanem hogy KI ellen nyersz.**

- Ha egy **Profi** megver egy **Kezdőt**: alig kap pontot a profi és a kezdő alig kap pontlevonást.
- Ha egy **Kezdő** megver egy **Profit**: Rengeteg pontot kap a kezdő (óriási teljesítmény), és sok pontlevonást kap a profi.

A feladat szimuláld a ranglistát!

1. A felhasználó megadja, hogy meccsek számát!
2. Rendezd, majd írd ki a kezdőállapotát!
3. Majd minden meccs esetén módosítsa a ranglistát:
 - a. Generáljon két különböző indexet [1-N] között!
 - b. A meccs 90%-ban a várható valószínűség alapján dől el!
 - c. Módosítsa a pontokat
 - d. Ellenőrzéshez: rendezd, majd írja ki a ranglistát! (Szükség esetén a változásokat is)

Példa:

Be: 100

---Kezdetek---

1: 125,79 pont

8: 136,42 pont

6: 152,43 pont

5: 309,42 pont

3: 350,51 pont

7: 578,23 pont

9: 692,69 pont

10: 718,23 pont
2: 851,13 pont
4: 889,2 pont

---Vége---

1: 184 pont
9: 238,82 pont
2: 278,88 pont
6: 307,54 pont
10: 438,34 pont
7: 526,72 pont
8: 662,2 pont
5: 712,23 pont
3: 853,54 pont
4: 909,34 pont