

Bevezetés a programozásba és algoritmikába

Vizsga

2019. június 08.

programozási feladatok

E-mailben megkaptatok mindent ami a vizsga megoldásához szükséges. A feladatokhoz tartozó projektek main.java fájljait a canvas rendszerbe kell feltölteni, a megfelelő feladathoz. A fájlt tetszőleges számúszor újra lehet tölteni a vizsga végéig, az utolsó feltöltött verziót értékeljük.

Kód formázás: amelyik feladat megoldásának formázása nem követi a standard formázási konvenciókat az a feladat nem kerül kijavításra.

A megoldásokat nem csak funkció, de általános kódminőség szerint is értékeljük, az is beleszámít a pontszámba.

1. (18 pont) Írjon egy programot ami az első parancssori argumentumaként kapott helyen lévő szöveges fájlból beolvasson egy duplapontos számokat tartalmazó kétdimenziós tömböt. Majd ezt a tömböt oszloponként kiírja egy azonos elérési úton, de .bin-re cserélt kiterjesztésű fájlba. A szöveges, olvasandó fájl formátuma: az első sorban szóközzel elválasztva szerepel a tömb mérete. Utána soronként szóközzel elválasztva a tömb sorai. A bináris, írandó fájl formátuma: az első bájtokon a tömb sorainak és oszlopainak száma szerepel egész számként, utána a tömb elemei oszloponként.

Pl.:

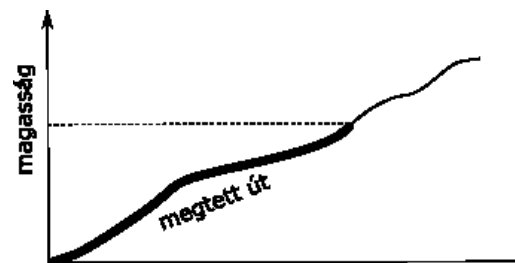
Szöveges fájl: 2 3 1 2.5 3 4 5 6	Bináris fájl (csak binárisan): 23142.5536
---	--

2. (15 pont) Van egy hegy, amire oly módon vezet fel egy ösvény hogy az folyamatosan felfelé emelkedik. Tudjuk, hogy 890 méteres magasságban van egy szép kilátó rét, de nem tudjuk, hogy mennyit kell az eléréséig gyalognunk.

A feladathoz tartozó mountain.java fájl tartalmát másolja egy új projekt main.java fájljába, majd írja meg úgy a megkezdett calculateDistanceToHeight függvényt, hogy annak segítségével 1 méteres pontossággal tudjuk meghatározni bármely magasság eléréséhez szükséges utat.

Ennek segítségével számolja ki a kilátóigpontig szükséges túra hosszát, és a kapott értéket kommentként írja is bele fájlba.

Törekedjen minél hatékonyabb algoritmust adni.



3. (12 pont) Írjon egy programot, ami egy **rekurzív algoritmussal** kiszámolja egy tömb elemeinek a háromszög összegét.

Egy tömb elemeinek a háromszögösszegét úgy kapjuk meg, hogy a tömb egymás mellett lévő elemeit addig adogatjuk össze, amíg már csak egy elem marad.

Pl.: A {1, 2, 3, 4, 5} tömb háromszögösszege 48, mert:

[1, 2, 3, 4, 5] -> [3, 5, 7, 9] -> [8, 12, 16] -> [20, 28] -> 48

4. (10 pont) Írjon egy programot, ami kiírja a képernyőre az összes olyan 9 (vagy valamilyen más) hosszú karakterláncot, ami csak 0 és 1 karaktereket tartalmaz, de nincs benne sehol egymást követő 1-es. Számolja is meg a megoldások számát és a megoldások után azt is írja ki, hogy hány megoldás volt.