

Programozás II. ZH

SZTE Szoftverfejlesztés Tanszék

2023. ősz

Technikai ismertető

- A programot C++ nyelven kell megírni.
- A megoldást a *Bíró* fogja kiértékelni.
 - A Feladat beadása felületen a Feltöltés gomb megnyomása után ki kell várni, amíg lefut a kiértékelés. **Kiértékelés közben nem szabad az oldalt frissíteni vagy a Feltöltés gombot újból megnyomni** különben feltöltési lehetőség veszik el!
- Feltöltés után a *Bíró* a programot g++ fordítóval és a
-std=c++17 -static -O2 -DTEST_BIR0=1
paraméterezéssel fordítja és különböző tesztesetekre futtatja.
- A program működése akkor helyes, ha a tesztesetek futása nem tart tovább 5 másodpercnél és hiba nélkül (0 hibakóddal) fejeződik be, valamint a program működése a feladatkiírásnak megfelelő.
- A *Bíró* által a `riport.txt`-ben visszaadott lehetséges hibakódok:
 - Futási hiba 6: Memória- vagy időkorlát túllépés.
 - Futási hiba 8: Lebegőpontos hiba, például nullával való osztás.
 - Futási hiba 11: Memória-hozzáférési probléma, pl. tömb-túindexelés, null pointer használat.
- A `riport.txt` és a fordítási log fájlok megtekinthetők az alábbi módon:
- A programot 20 alkalommal lehet benyújtani, a megadott határidőig.
- A programban szerepelhet `main` függvény, amely a pontszámításkor nem lesz figyelembe véve. Azonban ha fordítási hibát okozó kód van benne az egész feladatsor 0 pontos lesz.

Általános követelmények, tudnivalók

- Csak a leírásban szereplő osztályokat, metódusokat és adattagokat kell megvalósítani, egyéb dolgokért nem jár plusz pont.
- Minden metódus, amelyik nem változtatja meg az objektumot, legyen konstans! Ha a paramétert nem változtatja a metódus, akkor a paraméter legyen konstans!
- string összehasonlításoknál az egyezés a pontos egyezést jelenti, azaz ha kis-nagy betűben térnek el, akkor már nem tekinthetők egyenlőnek (pl. a "piros" != "Piros")
- A leírásokban bemutat példákban a string-ek köré rakott idézőjelek nem részei az elvárt kimenetnek, azok csak a string határait jelölik. Például ha az szerepel, hogy a példa bemenetre az elvárt kimenet az, hogy "3 alma", akkor az elvárt kimenet idézőjelek nélkül az 3 alma, de a szóköz szükséges!
 - A tesztesetekben nem lesz ékezetes szöveg kiírása.
- Az 1. és 2. ZH során STL tárolók használata nem megengedett! Tárolók használata 0 pontot eredményez.
- Az elvárt kimeneteknek karakterről karakterre olyan formátumúnak kell lennie, ami a feladatban le van írva (szóközöket és sortöréseket is beleértve).
- Ha az objektum másolása nem triviális (azaz a fordító által generált másolás nem elegendő), akkor a megfelelő másolást is meg kell valósítani.

Kiindulási projekt, megoldás feltöltése

- **A megoldáshoz az előre kiadott osztályok módosítása szükséges lehet.**
 - Nem minden ZH esetében van kiindulási projekt.
- Feltöltéskor ezeket az osztályokat is fel kell tölteni és a módosításokat is pontozhatja a bíró!
- Egyes tesztesetekben a bíró módosított osztályt is használhat ezen kiinduló osztályok helyett, ezzel tesztelve a valóban helyes működést!

Zh alatt használható segédanyag

- **A ZH során használható segédanyag elérhető bíróban.**
 - <https://biro.inf.u-szeged.hu/kozos/prog2/>

Titokzatos világ

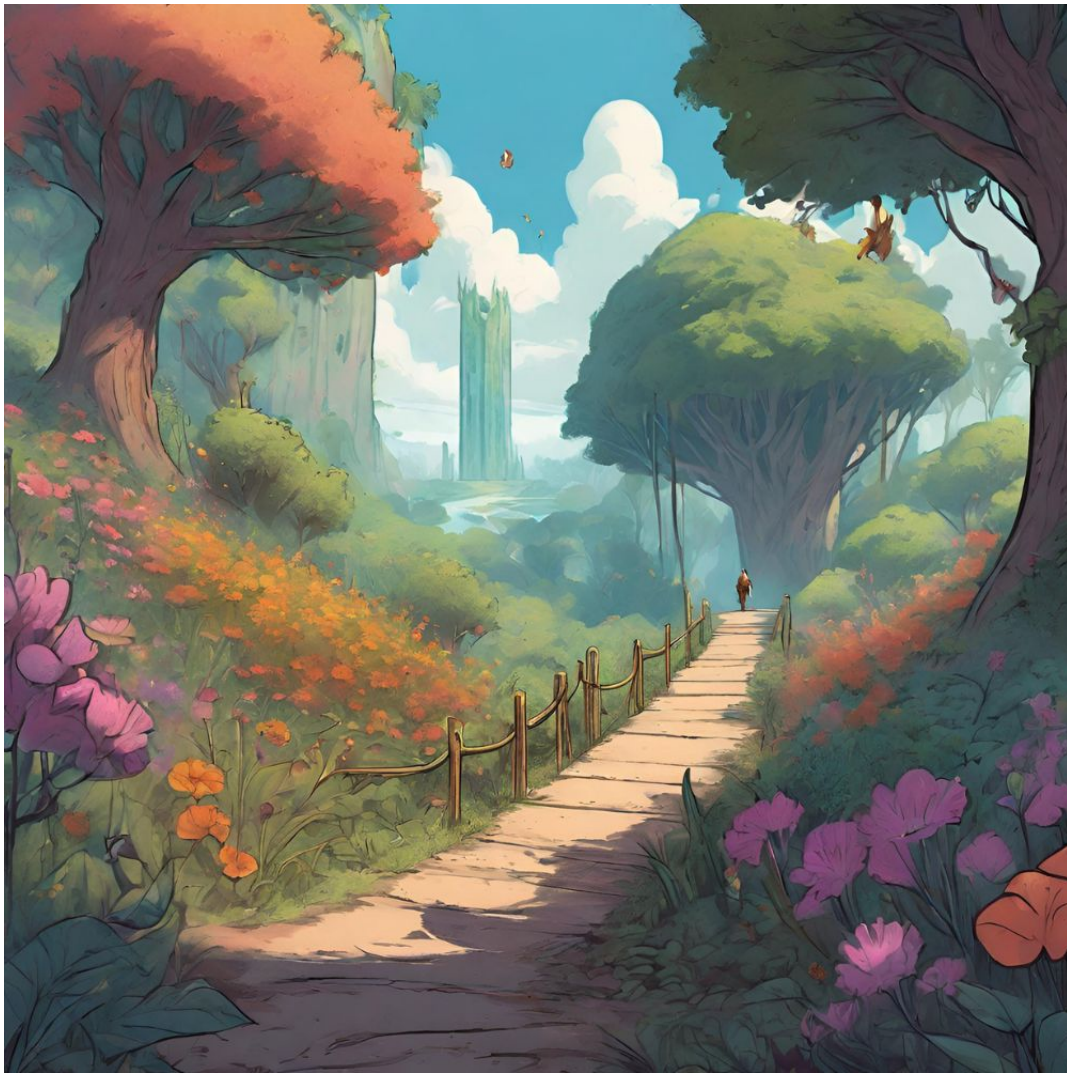
A feladat elkezdéséhez a minta.zip-ben található forráskódot kell megnyitni, és azt kell módosítani, kiegészíteni.



Az egyik reggeli futásunk során egy titokzatos helyre bukkantunk, ami egy átjáró volt a Föld egy titkos pontjára, ahol eddig még soha senki sem járt. Itt minden gyönyörű színes, csodálatos virágok, különleges állatok pompáznak mindenhol. Egy szóval nagyon uuu.

1. feladat: felfedezés (10 pont)

Az átjáró végén egy térképet találtunk, amin rajta vannak a legfőbb látványosságok, amelyeket meg tudunk nézni. Na de vajon mit lenne érdemes legelőször megnézni?



Az *Uticel* struktúra tartalmazza az úti cél nevét és a távolságát méterben.

Egészítsd ki az *Uticel* struktúrát a `<` operátor megvalósításával, hogy az úti célok egy halmazba tehetőek legyenek. A rendezés a név alapján történjen (az az úti cél a "kisebb", amelyiknek a neve előrébb van az ABC-ben). Figyelj oda a metódus megfelelő fejlécére!

Valósítsd meg a mintában található *eselyes_celpontok* függvényt!

A függvény paraméterben egy *Uticel* objektumokból álló halmazt kap, ami tartalmazza a potencionális úti célokat.

A függvény a **count_if** függvény segítségével határozza meg, hogy hány hozzánk közel álló úti cél van a térképen. Azok számítanak közeli úti célnak, amelyek 1000 méternél közelebb vannak. A függvény visszatérési értéke ez az érték legyen.

Megjegyzés: A megoldásra csak akkor kapsz pontot, ha a count_if függvényt pontosan egyszer használod és helyesen hívod meg.

2. feladat: célpont kiválasztása (7 pont)

Nem könnyű a választás, mert rengeteg gyönyörű szép hely található a térképen, amit szeretnénk megnézni. Ezért úgy döntöttünk, hogy szavazni fogunk, és oda megyünk, amelyik hely a legtöbb szavazatot kapja.



Valósítsd meg a mintában található *celpont_valasztas* függvényt!

A függvény paraméterben egy stringekből álló vektort kap, ami 1-1 szavazatot tartalmaz (hogyan melyik helyre szavaztunk). A függvény feladata, hogy visszaadja, hogy melyik hely kapta közülük a legtöbb szavazatot. Feltehetjük, hogy a legtöbb szavazatot kapó hely egyértelmű lesz.

PÉLDA

Input: ["Ezust pagoda", "Orokos tavak", "Orokos tavak", "Idokapu", "Tundervolgy", "Ezust pagoda", "Ezust pagoda", "Kristalybarlangok"]

Output: "Ezust pagoda"

3. feladat: Túra az új világban (7 pont)

Elindultunk felfedezni a gyönyörű szép új világot: az ezüst pagodát, a krisálybarlangokat... De sajnos nem mindenki maradt velünk végig az úton, mert páraoknak vissza kellett menni dolgozni, mert a mérsékelten okos (tehát nem az) főnökük értelmetlen feladatokkal traktálta őket folyamatosan.



A Turazo struktúra tartalmazza az ember nevét, illetve a főnökének az intelligenciáját. **A struktúrán ne változtass!**

Valósítsd meg a mintában található `visszamennek_dolgozni` függvényt. A függvény első paraméterben egy vektort kap referenciaként, amely `Turazo` objektumokat tárol. Második paramétere egy egész szám (átlagos intelligencia).

A függvény távolítsa el a vektorból azokat az embereket az `erase` és `remove_if` függvények segítségével, amelyek főnökének intelligenciája kisebb a paraméterben kapott átlagos intelligenciától.

Megjegyzés: Az eredményre csak akkor kapsz pontot, ha a `remove_if` függvényt pontosan egyszer használod és helyesen hívod meg.

4. feladat: Csend és nyugalom (7 pont)

Akiknek a főnöke intelligens, azok megtehetik, hogy egy napot nyugodtan eltölthetnek ebben az eddig nem látott világban. Olyan helyet próbálunk meg keresni, ahol egy állat sem él, hogy teljes csendben gyönyörködhessünk a tájban.



Valósítsd meg a mintában található *elhagyatott_helyek* függvényt. A függvény egy map-et vár paraméterben, ami megmondja, hogy az egyes helyeken (kulcsok) hány állat él (érték).

A függvény határozza meg a **find_if** függvény segítségével, hogy van-e olyan hely, ahol nem él egy állat se. Egy ilyen helyen szeretnénk nyugodtan eltölteni egy napot. A függvény adja vissza, hogy van-e ilyen hely.

Megjegyzés: A feladatra csak akkor kapsz pontot, ha a find_if függvényt pontosan egyszer használod és helyesen hívod meg.

5. feladat: az utazás véget ért (7 pont)

Rengeteg érdekes helyen voltunk a nap folyamán, és egy listát készítettünk az általunk legcsodálatosabbnak tartott helyekről. Mindenki ráírta a listára a kedvenc helyeit, ami természetesen azt is eredményezte, hogy egy hely többször is felkerült a listára.



Valósítsd meg a mintában található *egyedi_helyek* függvényt, amely paraméterben a feljegyzett helyeket kapja meg egy vektorban.

A függvény egy másik vektort adjon vissza, amelyben minden hely csak egyszer van benne (és nyilván minden helynek benne kell lennie, ami az eredeti vektorban is benne volt). Egyéb követelmény a visszaadott vektorra nincs, azonban az eredeti vektor maradjon változatlan!

6. feladat: fotózás (1 pont)

A nap során mindannyian készítettünk fotókat erről az eddig nem látott világról. Persze akkor még nem tudtuk azt, hogy amint elhagyjuk a területet, akkor az összes fotó megsemmisül, mintha csak álmodtuk volna az egészet.



A **Foto** struktúra tartalmazza a fotót készítő ember nevét, illetve az általa lefotózott helyet.

Valósítsd meg a mintában található *fotozas* függvényt!

A függvény egy **Foto** objektumokból álló tömböt vár paraméterben, illetve a tömb hosszát.

A függvény feladata, hogy visszaadjon egy olyan vektort, ami tartalmazza a fotózó emberek neveit az általa lefotózott helyek darabszáma szerinti csökkenő sorrendben. Tehát a vektor elején legyen az az ember, aki a legtöbb helyet fotózta le, míg a vektor végén legyen az az ember, aki a legkevesebb helyet fotózta le.

A feladat megoldása során feltehetjük, hogy mindenki más mennyiségű fotót készített.