

Programozás I.

Projektfeladat

Csóka Nándor

2024

Tartalomjegyzék

Fejlesztői dokumentáció.....	3
Feladat leírása.....	3
Készített program leírása.....	3
Konzolon megjelenő információk	4
Változók és szerepük.....	5
Felhasznált függvények és algoritmusok	6
Felhasználó dokumentáció	9
Hardver és szoftver követelmények	9
Minimális rendszerkövetelmények	9
Ajánlott rendszerkövetelmények.....	9
A program használata	9
Tesztelés	10
Továbbfejlesztési lehetőségek	11
Irodalomjegyzék.....	12

Fejlesztői dokumentáció

Feladat leírása

A készített program egy tesztverseny feldolgozására szolgál, ahol a versenyzők 14 tesztkérdésre adott válaszai és a helyes megoldások alapján pontszámok kerülnek kiszámításra. A program által számított eredményeket egy külön fájlba kiírjuk. A program lehetővé teszi versenyzők kódszámai és konkrét feladatoksorszámok alapján eredmények és statisztikák lekérdezését. Az adatokat a billentyűzetről kéri be a program, majd az eredmények a konzolablakon jelennek meg.

A versenyzőknek minden feladat esetén négy megadott lehetőség (A, B, C, D) közül kell a helyes választ megjelölniük. A versenybizottság garantálja, hogy tesztlapon minden kérdéshez pontosan egy helyes válasz tartozik. A kitöltött tesztlapokat elektronikusan rögzítik, a visszaélések elkerülése végett a versenyzőket betűkből és számokból álló kóddal azonosítják.

A helyes megoldást és a versenyzők válaszait a *valaszok.txt* szöveges állomány tartalmazza. A fájlban legfeljebb 500 versenyző adatai szerepelnek. A fájl első sorában a helyes válaszok szerepelnek. A fájl többi sora a versenyzők kódjával kezdődik, ezt egy szóköz, majd az adott versenyző által adott válaszok sorozata követi. A versenyzők kódja legfeljebb 5 karakterből áll. A válaszok a feladatokkal egyező sorrendben, elválasztójel nélkül, nagybetűvel szerepelnek. Ha a versenyző egy kérdésre nem válaszolt, akkor annak helyén X betű szerepel. Például:

BCCCDBBBBBCDAAA

AB123 BXCDBBACACADBC

AH97 BCACDBDDBCBBCA

Készített program leírása

1. A program beolvassa és tárolja a *valaszok.txt* szöveges állomány adatait.
2. Megjeleníti a képernyőn a fenti mintának megfelelően, hogy hány versenyző vett részt a tesztversenyen.
3. Bekéri egy versenyző azonosítóját, ami megjelenik a mintának megfelelően, a hozzá tartozó válaszokkal együtt. Csak létező azonosító esetén működik.

4. A helyes megoldás kiíródik a képernyőre, illetve, ha az adott versenyző a helyes választ eltalálta, akkor „+” jel kerül a kérdéshez, egyébként pedig egy szóköz.
5. Bekérhető egy feladat sorszáma, amire a program megmutatja hány versenyző adott a feladatra helyes választ és ez a versenyzők hány százaléka (alapértelmezetten 2 tizedesjegy pontossággal).
6. A program meghatározza az egyes versenyzők pontszámát, és a listát kiírja egy külön fájlba *pontok.txt* néven. Az állomány minden sora egy versenyző kódját, majd szóközzel elválasztva az általa elért pontszámot tartalmazza. (A verseny feladatai nem egyenlő nehézségűek: az 1-5. feladat 3 pontot, a 6-10. feladat 4 pontot, a 11-13. feladat 5 pontot, míg a 14. feladat 6 pontot ér).
7. A képernyőre kiírja a versenyen a három legmagasabb pontszámot elérő összes versenyzőt csökkenő sorrendben (díjazottak).

Konzolon megjelenő információk

1. feladat: fájlbeolvasás sikeressége
2. feladat: versenyzők létszámának kiírása
pl. 200 versenyző
3. feladat: egy versenyző kódjának bekérése billentyűzetről, majd az azonosítója és válaszainak megjelenítése
pl. BM577, válaszai: AAAABBBBCCCCDD
4. feladat: helyes megoldások kiírása és összevetése a bekért versenyző válaszaival
pl. AAAABBBBCCCCDD
X X X
5. feladat: feladat sorszámanak bekérése
pl. 106 helyes válasz, ez 34.98%
6. feladat: fájlkiírás sikeressége
7. feladat: díjazottak kódszámának és pontjának kiírása
pl. AB001 30
BA110 25

Változók és szerepük

név	típus	funkció
db	integer	A versenyzők számát tárolja.
tmb	dinamikus tömb	A versenyzők kódját és válaszait tartalmazza.
helyesValaszok	string	A helyes válaszokat tartalmazza.
kod	string	Egy adott versenyző azonosítója.
valaszok	string	A versenyzők által adott válaszok sorozata.
tv	osztály objektuma	Hivatkozás a függvényekre.
feladatIndex	integer	Egy adott feladat sorszáma, amelynek statisztikáját kéri.
maxVersenyzokSzama	integer	A programban maximálisan kezelhető versenyzők száma (500).
i, j	integer	ciklusváltozók
helyesDb	integer	Egy adott feladatra helyes választ adó versenyzők száma.
szazalek	double	Az előbbi helyes válaszok aránya a teljes versenyzői létszámhoz képest.
pontok	dinamikus tömb	A versenyzők kódját és pontszámait tartalmazza.
pontErtekek	integer	A 14 feladatra adott pontértékek tömbje.
pontszam	integer	Egy adott versenyző pontszámát tárolja a feldolgozás során.
maxPontok	integer	Az első három legnagyobb pontszámot tárolja.

Felhasznált függvények és algoritmusok

int main()

A program fő függvénye, mely meghívja az egyes részfeladatokat megoldó függvényeket. Itt van megadva a beolvasott *valaszok.txt* és a kiírandó fájl *pontok.txt* neve is.

void versenyzokSzamaKiir()

Funkció: Kiírja a versenyzők számát.

Algoritmus:

Be: A versenyzők számát tartalmazza.

Ki: Egy szöveg, amely tartalmazza a versenyzők számát.

Az algoritmus kiírja a "2. feladat: A versenyen [db] versenyző vett részt." üzenetet.

Algoritmus vége.

void valaszokMegjelenitese()

Funkció: Egy adott versenyző válaszainak kiírása.

Algoritmus:

Be: Egy versenyző kódja.

Ki: A versenyző válaszai megjelennek.

Ciklus $i := 1$ -től db-ig 1-esével

Ha $tmb[i].kod = kod$

akkor kiírja a versenyző válaszait.

Ciklus vége.

Algoritmus vége.

void helyesMegoldasok()

Funkció: Helyes megoldások és egy adott versenyző helyes találatainak megjelenítése.

Algoritmus:

Be: Egy versenyző kódja.

Ki: A helyes válaszok és a helyes találatok megjelenítése.

Ciklus $i := 1$ -től db -ig 1-esével

Ha egy válasz helyes, kiír egy "+" jelet, különben egy " " karaktert.

Ciklus vége.

Algoritmus vége.

void feladatStatisztika()

Funkció: Egy kérdés helyes megoldásainak számát és arányát jeleníti meg.

Algoritmus:

Be: Egy feladat sorszáma.

Ki: A helyes válaszok száma és aránya.

Ciklus $i := 1$ -től db -ig 1-esével

Ha a versenyző adott feladatra adott válasza helyes:

$helyesDb := helyesDb + 1$

Ciklus vége.

$szazalek := helyesDb / db * 100$

Algoritmus vége.

void pontszamitas()

Funkció: Pontszámok kiszámítása és mentése fájlba, valamint a díjazottak kiírása.

Algoritmus:

Be: A versenyzők válaszai és a helyes válaszok.

Ki: Pontszámok számítása, legjobb versenyzők meghatározása és kiírása fájlba.

Inicializál egy dinamikus tömböt (pontok) a versenyzők pontszámainak tárolására.

Ciklus $i := 1$ -től db -ig 1-esével

Ciklus $j := 1$ -től $helyesValaszok$ -ig 1-esével

Ha $tmb[i] = helyesValaszok[j]$

akkor $pontszam := pontszam + pontErtek$

Ciklus vége.

Ciklus vége.

A pontok tömb rendezése csökkenő sorrendbe sort algoritmus segítségével.

Az első három legnagyobb pontszám meghatározása és kiírása.

Ciklus $i := 1$ -től db -ig 1-esével

Ciklus $j := 1$ -től $count$ -ig 1-esével

Ha $pontok[i].pont = \max Pontok[j]$

akkor kiírja a versenyző kódszámát és a pontszámát.

Ciklus vége.

Ciklus vége.

A pontszámok kiírása egy fájlba.

Algoritmus vége.

Teszt (const `string&` fnev)

Funkció: Konstruktor, speciális függvény. Beolvassa az adatokat a megadott fájlból, és feltölti a tömböt. Neve megegyezik az osztály nevével.

~Teszt()

Funkció: Destruktor, speciális függvény. Dinamikusan lefoglalt memória felszabadítása a feladata. Neve megegyezik az osztály nevével.

Felhasználó dokumentáció

Hardver és szoftver követelmények

Minimális rendszerkövetelmények

Windows 10/11 alapú személyi számítógép (PC)

1 GHz, legalább 2 magos processzor vagy erősebb

4 GB RAM

500 MB lemezterület

Integrált videokártya

Windows 10 (32/64 bites) 1809-es vagy újabb verzió, vagy Windows 11

Ajánlott rendszerkövetelmények

Windows 10/11 alapú személyi számítógép (PC)

2 GHz-es, legalább 4 magos processzor vagy erősebb

8 GB RAM

1 GB szabad lemezterület

Dedikált videokártya 1 GB VRAM-mal

Windows 10 (64 bites) 21H1-es vagy újabb verzió, vagy Windows 11 (64 bites)

A program használata

- 1) A program telepítést nem igényel. Másolja a Tesztverseny mappát a C:/ meghajtóra.
- 2) Futtassa a programot a Beadandó.exe fájlra történő dupla kattintással.
- 3) A program beolvassa a bemásolt mappában található *valaszok.txt* szöveges fájlt, majd a konzolablak megjelenik a képernyőn és bekér a felhasználótól egy versenyző kódját.
- 4) Adja meg egy versenyző kódját (legfeljebb 5 karakterhosszúságú betű-és szám kombináció). pl. AB001
- 5) Adjon meg egy kérdéssorszámot (1-14-ig terjedő egész szám).
- 6) Ellenőrizze az eredményeket a konzolon és a program által létrehozott *pontok.txt* fájlban.

Képernyőkép egy helyes lefutás esetén:

```
Microsoft Visual Studio Debug Console
1. feladat: Fajl beolvasas es adatok tarolasa sikeres.
2. feladat: A versenyen 303 versenyző vett részt.
3. feladat: Kerem egy versenyző kódját (legfeljebb 5 karakteres szám-betű azonosító) : AL580
3. feladat: A AL580 kódú versenyző válaszai: AACCCBBBCDAACA
4. feladat: Helyes válaszok: BCCCDBBBBCDAAA
AL580 helyes válaszai: +++++ + + +
5. feladat: Kerem egy feladat sorszámat (1-14.-ig terjedő szám): 7
5. feladat: 102 versenyző adott helyes választ, ez a versenyzők 33.66%-a.
6. feladat: A pontszámok kiírása megtörtént a pontok.txt fájlba.
7. feladat: A díjazottak:
J0001 56
DG490 52
UA889 52
FX387 49

C:\Users\Nandi\Desktop\C++\Prog I\Beadandó feladatok\17\Beadandó\Debug\Beadandó.exe (process 10648) exited with code 0 (0x0).
Press any key to close this window . . .|
```

Tesztelés

A tesztelést 5db véletlenszerűen a *valaszok.txt* fájlból kiválasztott kódszámmal, 2db fiktív egyéb formátumú kódszámmal és 3db 1-14-en kívül eső feladatsorszámmal teszteltük Windows 11 számítógépen.

- ✓ Az 5db létező kódszám esetén a program az elvártaknak megfelelően működött hiba nélkül.

Mivel a feladatleírás megengedi, hogy feltételezzük, hogy csak helyes kódok lekérdezése történik, így fiktív kódszám megadása esetén a program tovább fut és a 3. feladat esetén üres válaszokat ad vissza.

- 5 jegyű létező kódszám helyett 34-gyel és 111111-gyel teszteltünk, a program tovább futott hibaüzenet nélkül.

Amennyiben az 5. feladatban 1-14-nél nagyobb vagy kisebb számot adtunk meg a következőket tapasztaltuk.

- 15 esetén 100%-ot ír ki a konzolra és tovább fut a program.
- 0 esetén hibaüzenetet ír ki és leáll a program.
- 1000 esetén szintén leáll a program.
- *Negatív szám* esetén is hibaüzenettel leáll a program.

Megszorításokat az adatbevitelre nem implementáltunk, mert a feladat megengedi, hogy a felhasználó által megadott adatok helyességét feltételezhetjük, hogy az elvárt formátumnak megfelelnek.

Továbbfejlesztési lehetőségek

- Grafikus felhasználói felület kialakítása, vagy a konzolablak megjelenítésének formázása.
- További statisztikai elemzések bevezetése pl. hányan buktak meg, átlagszámítás, visszatérő versenyzők korábbi évekből stb.
- Adatbázis-kapcsolat létrehozása szöveges fájlok helyett.

Irodalomjegyzék

- [1] w3schools.com: <https://www.w3schools.com/cpp/default.asp>
- [2] Stack Overflow: <https://stackoverflow.com>
- [3] A C++ programozási nyelv alapjai – Dr. Pásztor Attila
- [4] A C++ programozási nyelv középiskolásoknak – Pánczél István
- [5] NJE Moodle elektronikus tananyag
- [6] Egy beadandó feladat dokumentációs minta – PDF
- [7] YouTube: „C++ Haladóknak” tutorial videósorozat:
https://www.youtube.com/watch?v=wP-1lDgIbHI&list=PLuYHIg_0xQJtYJaPPIPgLNT2d11xZNNUS&index=12
- [8] OpenAI: ChatGPT ingyenes verzió – kódolás során hibaüzenetek elemzése, debugging, minimális és ajánlott szoftverkövetelmény esetén Windows 10/11 verzió meghatározása