

Programozás II. Gyakorló Feladat

SZTE Szoftverfejlesztés Tanszék

2024. ősz

Ismertető

- A programot C++ nyelven kell megírni.
- A benyújtandó fájl neve kötelezően `feladat.cpp`.
- 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++1y -static -O2 -DTEST_BIRO=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-túlinde克斯, null pointer használat.
- A `riport.txt` és a fordítási log fájlok megtekinthetők az alábbi módon:
 1. Az Eredmények megtekintése felületen a vizsgálandó próba új lapon való megnyitása
 2. A kapott url formátuma:
`https://biro2.inf.u-szeged.hu/Hallg/IBL302g-1/1/hXXXXXX/4/riport.txt`
 3. Az url-ből visszatörölve a 4-esig (`riport.txt` törlése) megkaphatók a 4-es próbálkozás adatai
- 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.
- A megvalósított függvények semmit se írnak ki a standard outputra!

Lokális tesztelés

A minta.zip tartalmaz egy kiindulási feladat.cpp-t, amit a megoldással kiegészítve lokális tesztelésre használhattok. A fordítás az előbbiekben leírt módon történjen. A fájl felépítése a következő.

```
#include <iostream>
#include <string>
#include <cassert>

using namespace std;

////////////////////
// Ide dolgozz!!
////////////////////

//== Teszteles bekapcsolasa kicommentezessel
//define TEST_alma
//== Teszteles bekapcsolas vege

#if !defined TEST_BIRO
/*
Keszits egy fuggvenyt, ami visszaadja az alma sztringet!
*/
void test_alma(){
    #ifdef TEST_alma && !defined TEST_BIRO
        string s = alma();
        assert(s == "alma");
    #endif
}
int main(){
    test_alma();
}
#endif
```

Ha megoldottad az alma feladatot, úgy tudod tesztelni, ha kitörölöd a kommentjelet a `#define TEST_alma` sor elől. Ekkor **újrafordítás** után le fog futni a `test_alma()` függvény tartalma is. Ha a visszaadott sztring nem az elvárt, az `assert()` függvény ezt jelezni fogja. A **define-ok módosítása nem javasolt**, fordítási hibát idézhet elő a `biro-n` való teszteléskor! **A tesztelőkód nem végez teljes körű tesztelést!** Saját felelősségre bővíthető. A sikeres megoldás után a `feladat.cpp` tartalma (mely `biro-ra` is feltölthető):

```
#include <iostream>
#include <string>
#include <cassert>

using namespace std;

string alma(){
    return "alma";
}

//== Teszteles bekapcsolasa kicommentezessel
#define TEST_alma
//== Teszteles bekapcsolas vege

/*
```

```

Készíts egy függvényt, ami visszaadja az alma sztringet!
*/
void test_alma(){
#ifdef TEST_alma && !defined TEST_BIRO
    string s = alma();
    assert(s == "alma");
#endif
}

int main(){
    test_alma();
}

```

Feladat

1. feladat (6 pont)

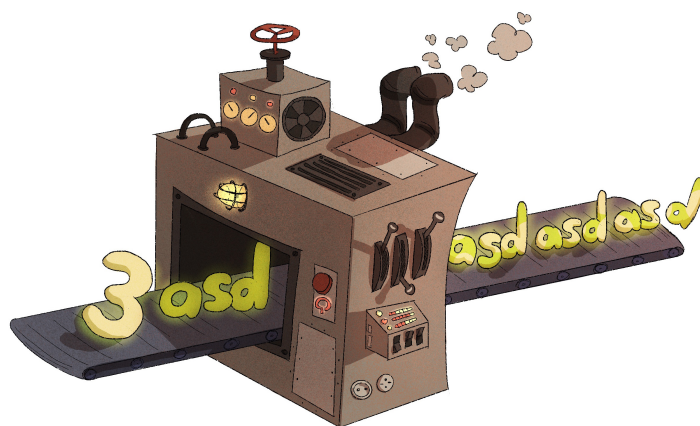
Készíts egy függvényt, amely beolvas egy tetszőleges hosszúságú sztringet a standard inputról, majd megszámolja hány darab „m” és „M” betű található benne. A két betű darabszámának összege legyen a függvény visszatérési értéke. A függvény egész számmal térjen vissza. A függvény semmit se írjon ki a standard outputra!

2. feladat (6 pont)

Készíts egy függvényt, amely beolvas előbb egy egész számot, utána pedig egy tetszőleges hosszúságú sztringet a standard inputról. A függvény visszatérési értéke egy sztring legyen, amelyben annyszor van megismételve (szóköz vagy egyéb elválasztójelek nélkül) a beolvasott sztring, amennyi a beolvasott egész értéke. Ha a beolvasott szám negatív a visszaadott sztring értéke „hiba” legyen. A függvény semmit se írjon ki a standard outputra!

Példa input: 3 asd

Példa output: asdasdasd



3. feladat (4 pont)

Készíts egy függvényt, amelynek paramétere egy előjel nélküli egész szám. A függvény visszatérési értéke egy sztring legyen, amelyben a paraméterül kapott szám 3-szorosa van sztringgé konvertálva. A függvény semmit se írjon ki a standard outputra!

Példa input: 2

Példa output: 6

4. feladat (10 pont)

Készíts egy függvényt, amely két string-et vár paraméterül. A string-ek egy-egy egész számot tartalmaznak. A függvény visszatérési értéke az a szám lesz (sztringről egész számmá konvertálva), amelyiknek az értéke kisebb. Amennyiben az egyik sztring üres vagy a konverzió nem elvégezhető, akkor a másik sztring számértéke legyen visszaadva. Ha mindkét sztring üres vagy nem konvertálható, akkor a visszatérési érték -999 legyen. A függvény semmit se írjon ki a standard outputra!

Példa input: asd 42

Példa output: 42