

Programozás 1

– 1. zárthelyi dolgozat –

2021. márc. 31., 10.00 órás csoport

Feladatok

1. Írjon egy programot, ami parancssori argumentumként vár két sztringet. A program fésülje össze a két sztringet s írja ki a képernyőre az eredményt.

Összefésülni csak két azonos hosszúságú sztringet lehet. Az összefésülés alatt azt értjük, hogy vesszük az első sztring első karakterét, ill. a második sztring első karakterét. Majd vesszük az első sztring második karakterét, ill. a második sztring második karakterét, stb.

Például a „Bea” és az „abc” összefésülése a „Baebac” eredményt adja.

Az összefésülést a következő eljárással valósítsa meg:

```
void merge(string s1, string s2);
```

Mint látható, lehet használni a `prog1` könyvtárat. Az eljárás kap két sztringet, s kiírja az eredményt (az összefésült sztringet) a képernyőre.

A programnak több dolgot is le kell ellenőriznie, s hiba esetén informatív módon tájékoztatni kell a felhasználót a hiba jellegéről. (Ha az ékezetek gondot okoznak, akkor az ékezetek használatától el lehet tekinteni).

Futási példák:

```
$ ./a.out
Hiba! Adj meg pontosan két sztringet!
```

```
$ ./a.out na
Hiba! Adj meg pontosan két sztringet!
```

```
$ ./a.out aa bb cc
Hiba! Adj meg pontosan két sztringet!
```

```
$ ./a.out abc 1234
Hiba! A két sztring legyen azonos hosszúságú!
```

```
$ ./a.out Bea abc
Baebac
```

```
$ ./a.out hello Beata
hBeelaltoa
```

2. Írjon egy programot, ami véletlenszámokkal feltölt egy 1000 elemű tömböt. A véletlenszámok a [125, 850) (balról zárt, jobbról nyílt) intervallumból kerüljenek ki.

Az 1000-es értéket tárolja el egy nevesített konstansban, s a tömb létrehozásakor már ezt a nevesített konstanszt használja!

A könnyebb tesztelés miatt a véletlenszám-generátort a 2021 konstans értékkel inicializálja (ezzel azt érzük el, hogy a program többszöri futtatás esetén is ugyanazokat a véletlenszámokat fogja előállítani ugyanabban a sorrendben).

A tömb véletlenszámokkal való feltöltését egy eljárással valósítsa meg! Az eljárás deklarációs része így nézzen ki:

```
void feltolt(const int n, int tomb[]);
```

Vagyis az eljárás megkapja a tömböt, s a fent említett módon feltölti a tömböt véletlenszámokkal.

A program kimenete egyetlen egész szám legyen, ami azt mutatja meg, hogy a tömbben hány darab olyan szám van, aminek az értéke nagyobb mint a tömbben lévő számok átlaga.

Futási példa:

```
$ ./a.out
245
```

A „245” helyett természetesen a helyes értéket kell kiírni.

3. A Fibonacci-sorozat következő elemét úgy kapjuk meg, hogy vesszük a sorozat előző két elemének az összegét. Ha a sorozat az 1 és 2 értékekkel indul, akkor a sorozat első 10 eleme a következő lesz:

1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

Mennyi lesz a sorozat 4 milliónál kisebb *páratlan* elemeinek az összege?

A program kimenete egyetlen egész szám (a fenti kérdésre adott válasz) legyen.

4. Írjon programot, ami interaktív módon sztringeket olvas be ‘*’ végjelig. A program a sztringek bevitele után írjon ki egy kis statisztikát. Példa:

```
$ ./a.out
Szöveg (vége: *): hello
Szöveg (vége: *): hello world
Szöveg (vége: *): ***
Szöveg (vége: *): END
Szöveg (vége: *): 2021
Szöveg (vége: *): *
```

```
Kisbetűk száma: 15
Nagybetűk száma: 3
Szóközők száma: 1
Számjegyek száma: 4
```

A statisztika kiíratásakor a számok egymás alatt, balra igazítva jelenjenek meg.