

Programozási nyelvek I.

- zárthelyi dolgozat -

2021. 05. 13.

Feladatok

1. Feladat

Írj egy programot, amely egész számokat vár parancssori argumentumként, és ezek közül a középsőt a két szomszédja egész osztású hányadosának hatványára emeli. Az osztásnál a nagyobb szomszédot oszd el a kisebbik szomszéddal! Ha a felhasználó nem ad meg értékeket, vagy páros az argumentumszám, akkor a program írjon ki egy hibaüzenetet a standard hibakimenetre és lépjen ki 1-es hibakóddal! Feltételezheted, hogy a felhasználó egész számokat ad meg bemenetként.

Példa: 7 5 2 esetén, a középső elem 5, a két szomszéd egész osztású hányadosa $7 / 2 = 3$, az eredmény $5^3 = 125$.

1 2 3 4 5 6 7 esetén a középső elem 4, és a hatvány amire emelni kell $5 / 3 = 1$, tehát az eredmény 4.

Megjegyzés: A math.h library használatához a fordításnál szükséges a „-lm” kapcsoló

```
veress@veress:~/judit/prog/zh2$ gcc feladat1.c -lm
veress@veress:~/judit/prog/zh2$ ./a.out
Nem megfelelő argumentumszám!
veress@veress:~/judit/prog/zh2$ ./a.out 1
Nem megfelelő argumentumszám!
veress@veress:~/judit/prog/zh2$ ./a.out 1 2
Nem megfelelő argumentumszám!
veress@veress:~/judit/prog/zh2$ ./a.out 1 2 3
Eredmény: 8
veress@veress:~/judit/prog/zh2$ ./a.out 1 2 3 4
Nem megfelelő argumentumszám!
veress@veress:~/judit/prog/zh2$ ./a.out 7 5 2
Eredmény: 125
veress@veress:~/judit/prog/zh2$ ./a.out 1 2 3 4 5 6 7
Eredmény: 4
```

2. Feladat

Adott a *szavak.txt* szöveges állomány, melynek 500 sora van és soronként egy szót tartalmaz. Írj egy programot, amelyben feldolgozod a fájl tartalmát (ami nem módosítható!), a program írja ki a standard kimenetre, hogy hány olyan szó található a fájlban, amely több, mint négy szótagból áll. A fájl nem tartalmaz ékezetes karaktereket.

Minden szó annyi szótagú, ahány magánhangzó van benne, az egy karakterből álló szavak egy szótagúak.

Példa 10 szó esetén:

```
1  ABLAKFIOK
2  ABLAKFULKE
3  ABLAKKERET
4  ABLAKKOSAR
5  ABLAKMELYEDES
6  ABLAKNYILAS
7  ABLAKOS
8  ABLAKOZ
9  ABLAKPARKANY
10 ABLAKPARNA
11

veress@veress:~/judit/prog/zh2$ gcc feladat2.c
veress@veress:~/judit/prog/zh2$ ./a.out
A több, mint négy szótagból álló szavak száma: 1
A legnagyobb szótagszám: 5
veress@veress:~/judit/prog/zh2$
```

3. Feladat

Írj egy programot, amelyben véletlenszerűen feltöltesz egy 6x6-os mátrixot az [55,155] zárt intervallumból. A randomszámgenerátort a 33-as értékkel inicializáld! A program írja ki a képernyőre a mátrixot, valamint a széleken elhelyezkedő elemek átlagát.

Példa 3x3-es mátrix esetén:

```
veress@veress:~/judit/prog/zh2$ gcc feladat3.c
veress@veress:~/judit/prog/zh2$ ./a.out
82 112 138
80 58 130
144 57 135
A szélső elemek átlaga: 109.75
```

4. Feladat

Egy digitális kép tárolásánál minden egyes képpont színét tároljuk. A képpontok színét az RGB kód adja. Az RGB kód a vörös (R), zöld (G) és a kék (B) színösszetevő értékét határozza meg. Ezen színösszetevők értéke 0 és 255 közötti egész szám lehet.

A *kep.txt* fájlban egy 50×50 képpontos kép képpontjainak RGB kódjai vannak tárolva. A fájlt nem módosíthatod! Az állomány a képet sorfolytonosan, a képpontok RGB kódját pontosvesszővel elválasztva tartalmazza, minden képpontot egy újabb sorban.

Írj egy programot, amelyben módosítsd a képet úgy, hogy ha a képpontok KÉK színösszetevőinek értéke kisebb, mint 100, akkor hússzal növeld ezt az értéket! A módosított kép képpontjainak színét írd ki a "*kekitett.txt*" nevű szövegfájlba a bemeneti fájl formátumával egyezően! A képet sorfolytonosan tárold, minden képpontot új sorba, a képpontok RGB kódját pontosvesszővel elválasztva írd ki!

Ha a program sikeresen lefutott írd ki „#Kész!” üzenetet a standard kimenetre!

Példa a egy 10 soros bemeneti és kimeneti fájl esetén:



```
veress@veress:~/judit/prog/zh2$ gcc feladat4.c
veress@veress:~/judit/prog/zh2$ ./a.out
#Kész!
veress@veress:~/judit/prog/zh2$
```

File	Line	RGB Code
kep.txt	1	178;117;150
	2	50;100;200
	3	255;255;255
	4	255;0;0
	5	255;0;0
	6	200;96;64
	7	255;255;255
	8	50;100;200
	9	20;0;255
	10	0;0;0
kekitett.txt	1	178;117;150
	2	50;100;200
	3	255;255;255
	4	255;0;20
	5	255;0;20
	6	200;96;84
	7	255;255;255
	8	50;100;200
	9	20;0;255
	10	0;0;20