

Tartalomjegyzék

- 1 Feladatok
 - ◆ 1.1
Távolság
 - ◆ 1.2
Min-max
 - ◆ 1.3
Rendezés
 - ◆ 1.4 C
string
 - ◆ 1.5
Hisztogram
 - ◆ 1.6
Csillagképek
 - ◆ 1.7 Zárt
terület
mérete

Feladatok

Minden feladathoz nyiss új projektet IDE-ben vagy írd új file-ba ha parancssorból dolgozol!

Távolság

- Írjunk függvényt ami kap két 2 dimenziós koordinátát, kiszámolja a távolságukat és visszaadja azt.
- Definiáljuk hozzá a megfelelő struct típust.
- Írjunk **main** függvényt amiben leteszteljük a működését!
- A gyökvonáshoz használjuk a következő könyvtárat (hasonló mint az stdio.h):

```
#include<math.h>
```

Ez után tudjuk használni az **sqrt** függvényt:

```
sqrt(4)
```

Min-max

Írjunk függvényt mely paraméterként kap 2 int pointert min és max. A függvény ellenőrizze, hogy a min-ben tárolt érték kisebb (vagy egyenlő) mint a max-ban tárolt és ha nem akkor cserélje meg az értékeket.

Rendezés

Írjunk függvényt mely nagyság szerint rendezi egy (int) tömb elemeit.

C string

Elemezzük ez a kis kódot:

```
#include <stdio.h>

int main() {
    char str[] = "puppy";
    printf("The word is %s.\n", str);
}
```

```
    return 0;
}
```

Próbáljuk meg kiírni az str tömb elemeit egyesével mintha egész számok lennének (%d printf-ben)! Az str-nek 6 eleme van (annak ellenére, hogy csak 5 karakterből áll).

Beolvasni is tudunk string-et, de ehhez egy előre létrehozott kellően nagy char tömb kell:

```
char str[100];
scanf("%s", str);
```

Tehát a string-ek valójában karakter tömbök, melyek végén egy úgynevezett lezáró nulla karakter áll. Ennek a karakter kódja '\0'. Egy szokásos string bejáró **for** ciklus:

```
for(i = 0; str[i] != '\0'; i++) {
    printf("%c", str[i]);
}
```

Hisztogram

- Írj függvényt, mely megszámolja egy egész számokból álló tömb elemeinek a gyakoriságát.
- A függvény paraméterként kapja meg a vizsgált tömböt és annak méretét. Feltételezzük, hogy a tömbben 0 és 10 közötti értékek szerepelnek (miért van erre szükség?).
- Az eredmény tömböt adja vissza paraméteren keresztül.
- Írjunk hozzá **main** függvényt ami beolvas 10 számot egy tömbbe lefuttatja a tömbre a megírt függvényt, majd kiírja az eredményt a kimenetre.

Csillagképek

Adottak csillagok x és y koordinátái (2 dimenziós térben dolgozunk) határozzuk meg azokat, melyek egy csillagképbe tartoznak! A csillagkép alatt csillagok olyan halmazát értjük ahol minden csillagból minden csillagba el lehet jutni legfeljebb 10 egységnyi távolság lépésekkel csillagok között. (Nem a teljes úthossz számít, csak az hogy van egy akárhány csillagból álló út amiben minden csillag-csillag lépés maximum 10 egység.)

A csillagok koordinátái:

```
float x[] = {28.1,13.9,29.1,25,6.5,16.4,5.9,24.6,4.4,6.3,24,2.4,18.8,12.9,2.5,26.8,22.4,13,21.3,4.
float y[] = {29.2,12.4,10.8,5,25.6,4,17.2,22.4,29.3,28.7,16.3,23,3.3,3.6,4.9,23.9,27.4,1.7,23.8,19
```

Kimenetre írjuk ki soronként vesszővel elválasztva az egy csillagképbe tartozó csillagok indexeit!

Ha megvagyunk a feladat megoldásával, akkor módosítsuk úgy a programot, hogy 1 struct tömb tárolja a koordinátákat!

Zárt terület mérete

Adott a következő (vagy hasonló) ábra, melyen '.' jelzi az üres teret és '#' jelzi a falakat. Írjunk függvényt mely kiszámolja a legnagyobb területű "http://wiki.math.bme.huszoba"http://wiki.math.bme.hu méretét (hány '.' karakter van benne). Ez egy 37x19-es tábla.

```
.....
...#####.....
...#.....#.....
```

```

...#.....#.....
...#.....#.....
...#.....#.....
...#.....#.....
...#.....#####
...###.....#.....
...#..##.....##.....#.....
...#...##.....##.....#.....
...#.....##.....##.....#.....
...#.....#####.....#.....
...#.....#.....#.....#.....
...#.....##.....#.....#.....
...#.....##.....#.....#.....
...#.....##.....#.....#.....
...#...##.....#.....#.....
...#####
.....

```