

# Tartalomjegyzék

- 1. 1. gyakorlat
  - ◆ 1.1 C fordítás konzolban
    - ◇ 1.1.1 Els? fordítás gcc-vel
    - ◇ 1.1.2 Második fordítás gcc-vel
  - ◆ 1.2 CodeBlocks használata
    - ◇ 1.2.1 Használat
  - ◆ 1.3 CodeLite használata
    - ◇ 1.3.1 Beállítások
    - ◇ 1.3.2 Használat
  - ◆ 1.4 Feladatok
    - ◇ 1.4.1 1. Feltételes print
    - ◇ 1.4.2 Segítség

## 1. gyakorlat

### C fordítás konzolban

Els? fordítás gcc-vel

Nyiss egy szövegszerkeszt?t, és másold be egy új fájlba a "http://wiki.math.bme.hu/hello.c" http://wiki.math.bme.hu kódját:

```
#include <stdio.h>
int main() {
    int x = 2;
    printf("hello world! %d\n", x);
    return 0;
}
```

Nyiss egy konzolt és menj abba a könyvtárba (cd paranccsal) ahová elmentetted a "http://wiki.math.bme.hu/hello.c" http://wiki.math.bme.hu-t.

Fordítsd le, készíts "http://wiki.math.bme.hu/hello" http://wiki.math.bme.hu nev? futtatható fájlt:

```
gcc -o hello hello.c
```

Futtasd le a kapott programot!

### Második fordítás gcc-vel

Mostantól használjuk a gcc warning-kapcsolóit, hogy lássuk a kisebb hibákat, figyelmeztetéseket is!

```
gcc -W -Wall -o hello hello.c
```

Rontsd el a "http://wiki.math.bme.hu/hello.c" http://wiki.math.bme.hu kódot a következő módokon, majd mindig mentsd el és fordítsd le újra (és a következő elrontás el?tt mindig állítsd vissza a helyes verziót):

- törölsz egy pontosvesszőt a sor végén?
- a printf-es sorban x helyett valami más betűt írsz
- létrehozol egy y nevű egész típusú változót az x után, de nem használod a kódban
- kitörölsz az utolsó } -t
- törölsz a "http://wiki.math.bme.hu/return 0;" http://wiki.math.bme.hu sort
- törölsz a legelső #include sort

## CodeBlocks használata

### Használat

- A feladatokhoz hozzatok létre mindig egy új üres file-t (File / New / Empty file), ezt mindenképpen .c kiterjesztéssel mentsetek el pl: *feladat1.c*.
- Fordítani a Build / Build-el tudtok, vagy a megfelelő ikonnal, a lefordított programot a Build / Run -al indíthatjátok, vagy egyszerűbben F9-el fordíthattok majd utána azonnal indíthatjátok is a programot.

## CodeLite használata

### Beállítások

- Indítsd el a CodeLite fejlesztői környezetet!
- Állítsd át az indentálást (tabulálást/beljebbezést) a *Settings -> Global Editor Preferences* menüpont alatt a *General -> Indentation* résznél hogy ne használjon tab karaktereket
- Hozz létre egy új Workspace-et és benne egy új projektet (mindenképp külön könyvtárban: *Create the project under a separate directory* legyen bepipálva), segítség itt az oldal vége felé

### Használat

- Minden feladat megoldását új projektbe kell tenni, hogy a *main()* függvények ne ütközzenek (egy projekten belül nem lehet több bel?le).
- A narancssárga az aktív projekt a workspace-en belül. Ha menüből kiválasztod a fordítást (*Build -> Build project* vagy F7 billentyű?) akkor ez mindig az aktív projektet fogja lefordítani, ha épp másik file van megnyitva a szerkesztőben, akkor is! Ugyanígy a futtatásnál is (menüből *Build -> Run* vagy Ctrl-F7) az aktív projekt *main()* függvénye fog lefutni!

## Feladatok

Minden feladatot új file-ba mentsd a CodeBlocks-ban (esetleg Codelite-ban új project-be)!

### 1. Feltételes print

#### Segítség

A következő egy egyszerű feltétel, leteszteli hogy 'x' nulla vagy nem, majd kiírja a számot:

```
#include<stdio.h>

int main(void) {
    int x = -5;

    if(x == 0){
        printf("Nulla");
    } else{
        printf("Nem nulla");
    }
    printf("A szam: %d \n\n", x);

    return 0;
}
```

Egészítsd ki a következ? kódot a megjegyzések helyén!

```
#include<stdio.h>

int main(void) {
    /* x és y változók deklarálása, értékadás */

    printf("X erteke: %d\n", x);
    printf("Y erteke: %d\n", y);

    printf("\n");

    /* egy feltételes utasítással (if, else if, else) érd el hogy mindig igaz állítás kerüljön a k
    printf("%d nagyobb mint %d!\n", x, y);
    printf("%d nagyobb mint %d!\n", y, x);
    printf("%d ugyanannyi mint %d!\n", x, x);

    return 0;
}
```