

## Tartalomjegyzék

- 1 El?adás
- 2 C fordítás konzolban
  - ◆ 2.1 Els? fordítás gcc-vel
  - ◆ 2.2 Második fordítás gcc-vel
- 3 CodeBlocks használata
  - ◆ 3.1 Használat
- 4 CodeLite használata
  - ◆ 4.1 Beállítások
  - ◆ 4.2 Használat
- 5 Ciklusok
  - ◆ 5.1 For ciklus
  - ◆ 5.2 While ciklusok
- 6 Feladatok
  - ◆ 6.1 Üzemanyag
  - ◆ 6.2 Legnagyobb közös osztó
  - ◆ 6.3 Fizika

### El?adás

[https://docs.google.com/presentation/d/1TiC4mp\\_oNR2o3q5L0lV9nO-Q1YizBtmsYDRB2HByd3w/edit#slide=id.g27017](https://docs.google.com/presentation/d/1TiC4mp_oNR2o3q5L0lV9nO-Q1YizBtmsYDRB2HByd3w/edit#slide=id.g27017)

### C fordítás konzolban

#### Els? fordítás gcc-vel

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

Fordítsd le, készíts "http://wiki.math.bme.hu/hello" nevű 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
```

## 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 fordíthatok 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 *main()*).
- 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!

## Ciklusok

## For ciklus

- A for ciklus nem a sage-ben megszokottak szerint működik, hisz itt nincsenek listáink amiket bejárhatnánk.
- Ehelyett a for a while ciklushoz nagyon hasonlóan működik, a szintaxis:

```
for(inicializálás; feltétel; inkrementálás){
    utasítások
}
```

- Az inicializálás részben adhatjuk meg azokat az utasításokat amiket csak egyszer a ciklus kezdetekor szeretnénk végrehajtani, ha úgy gondolunk rá mint egy szummára, akkor ez lehet az  $i = 0$  például.
- A ciklus akkor áll le, amikor a feltétele hamis lesz, tehát amíg igaz, addig fut.
- Az inkrementálás lépés a ciklus belsejében levő utasítások (a ciklus magja) után hajtódik végre. Itt tipikusan növelünk egy ciklusváltozót, de akármi mást is lehetne csinálni.

- Példa egy for ciklusra, ami kiírja a számokat 0-tól 9-ig:

```
int i;
for (i=0; i<10; i++) {
    printf("A ciklusváltozo erteke: %d\n", i);
}
```

- Az `++` egyenértékű az `i = i + 1` vagy az `i += 1` utasítással.

- Példa egy sokkal kevésbé hagyományos for ciklusra:

```
int i = 1;
int j = 1;
for (; i + j != j * 2; i = j + 1) {
    j = i * 2;
}
```

- Ez a példa azt is mutatja, hogy a ciklus fejének különböző részei egymástól függetlenül elhagyhatók, valamint hogy létezik végtelen ciklus.

## While ciklusok

- A while ciklus a már sage-ben megszokottak szerint működik, addig fut, amíg igaz a fejében található feltétel.
- Talán annyi plusz van, hogy bevezetjük a hátul tesztelős ciklust, a do while ciklust. Ez a nevéből is kikövetkeztethetően utólag tesztel, tehát a magja legalább egyszer lefut.
- A szintaxis:

```
while(feltétel){
    utasítások
}

do{
    utasítások
} while(feltétel);
```

- Példa egy bonyolultabb while ciklusra, a feladatokban lesz egy hozzá nagyon hasonló:

- A felhasználótól egy ciklusban egész számokat kér be addig, amíg 0 értéket nem kap. Ekkor pedig kiírja a képernyőre a kapott nemnulla számok átlagát.

```
#include<stdio.h>

int main(void) {
    int i = 0;
    float sum = 0;
    int szam = 0;

    do{
        scanf("%d", &szam);
        sum += szam;
        i++;
    } while(szam != 0);

    i--;
    printf("%f", sum / i);

    return 0;
}
```

- Bejött egy új típus, a float, ő a lebegőpontos szám, mondhatjuk, hogy a tizedestört. A printf és scanf-ben használandó mintája a "<http://wiki.math.bme.hu/%f>"<http://wiki.math.bme.hu>.

## Feladatok

- Írd ki az els? 100 számot és azok négyzetét
- Írd ki az els? k Fibonacci számot (most C-ben:))
- Írd ki az els? 10 számot és paritásukat.
- Írd ki az els? 12 szám faktoriálisát.

## Üzemanyag

- A program bemenetként kapja meg, hogy a felhasználó hány gallon üzemanyagot használna fel.
- A program írja ki az alábbi információkat:
  - ◆ A felhasznált üzemanyag mennyisége literben
    - ◇ 1 gallon = 3.7854 l
  - ◆ Az üzemanyag gyártásához szükséges olaj mennyisége hordóban mérve.
    - ◇ Mindenképpen egész hordók vásárlására van szükség.
    - ◇ 1 hordó olajból 19.5 gallon üzemanyag készíthet?.
  - ◆ A megfelel? CO2 terhelés kg-ban
    - ◇ 1 gallon üzemanyag esetén ez ~ 9 kg CO2
  - ◆ A hordókban vásárolt alapanyag olaj ára az aktuális árfolyamon 3 különböz? pénznemben: EUR, USD, HUF.

## Legnagyobb közös osztó

- Határozd meg két szám legnagyobb közös osztóját

## Fizika

- A program bemenetként kapja meg, hogy az ?rutazáshoz használt ?rhajó sebessége hány %-a a fény sebességének (300 000 000 m/s).
  - ◆ Például 10% esetén a bemenet 0.1 és ekkor az ?rhajó sebessége 30 000 000 m/s
  - ◆ Az ?rhajó tömegét, feltéve, hogy nyugalmi tömege 70,000 kg.
  - ◆ Azon eltelt id?ket, melyeket az ?rhajósok érzékelnének, ha a következ? objektumokra utaznának a Földr?l:
    - ◇ Alpha Centauri: 4.3 fényév

## Informatika2-2014/Gyakorlat09

- ◇ Androméda Galaxis 2 000 000 fényév
- ◆ Segítség:
  - ◇ <http://hu.wikipedia.org/wiki/Id%C5%91dilat%C3%A1ci%C3%B3>
  - ◇ [http://hu.wikipedia.org/wiki/Relativisztikus\\_t%C3%B6meg](http://hu.wikipedia.org/wiki/Relativisztikus_t%C3%B6meg)