

## A csoport

### 1. feladat

- **globális változók**
  - globális változó, fordítási időben értékelődik ki egyszer, ha nem inicializáljuk akkor 0-ra fog inicializálódni, elérésük a scope operátor segítségével lehetséges (::), használatuk kerülendő
- **fordítás 3 lépése (nem kell részletezni)**
  - preprocessálás (előfeldolgozás)
  - fordítás (a tárgykód létrehozása)
  - linkelés (szerkesztés)

### 2. feladat - helyes-e az alábbi program? Ha igen miért, ha nem miért?

```
int** const ppl;  
int* const * const * const pppl = &ppl;
```

- Helytelen.

Az első sorban egy int-re mutató mutatóra mutató konstans mutató van. Konstans mutatónak deklaráláskor kell kezdőértéket adni. Fordítási idejű hiba.

### 3. feladat - írd meg a swap függvényt referencia szerinti érték átadással

```
void swap(int& i, int& j)  
{  
    int tmp = i;  
    i = j;  
    j = tmp;  
}
```

## B csoport

### 1. feladat

- **pointerek**
  - olyan nyelvi elemek, melyek egy adott típusú memóriaterületre mutatnak. A mutatóknak egy másik változó memóriacímét kell értékül adni. A mutatott értéket a dereferáló operátor (\*) segítségével tudjuk elérni. Megkülönböztetünk konstansra mutató mutatót és azt amikor maga a mutató konstans. Az elsőnél a mutatón keresztül nem tudjuk módosítani a mutatott értéket viszont át tudjuk állítani a mutatót, hogy máshova mutasson, a másodikonál pedig a mutatott értéket tudjuk a mutatón keresztül módosítani viszont nem tudjuk átállítani a mutatót. A másodikonál deklaráláskor értéket kell adni a mutatónak.
- **ODR (One Definition Rule)**
  - Egy adott függvényt (objektumot, osztály) akárhányszor deklarálhatunk, azonban ha a deklarációk ellentmondanak egymásnak, akkor fordítási hibát kapunk. Definálni viszont a legtöbb esetben pontosan egyszer szabad. Több definíció vagy a definíció hiánya problémát okozhat.

## 2. feladat - helyes-e az alábbi program? Ha igen miért, ha nem miért?

```
const int** ppl;  
int const* * const* pppl = &ppl;
```

- Helyes.

Az első sorban egy const int-re mutató mutatóra mutató mutató van.

A második sorban szintén egy const int-re mutató mutatóra mutató konstans mutatóra mutató mutató van. Mind a kettő const int-re mutat, rendben van.

## 3. feladat - írd meg a swap függvényt pointer szerinti érték átadással

```
void swap(int* i, int* j)  
{  
    int tmp = *i;  
    *i = *j;  
    *j = tmp;  
}
```

## C csoport

### 1. feladat

- **referenciák**
  - egy létező objektum alternatív neve. Definiálásánál meg kell adni azt az objektumot is, amelyre referál. Nem egy változó, mint a pointer, hanem csak azonosító, ezért nincs címe és nem is változtatható meg. Amíg létezik mindig ugyan oda mutat.
- **konstans korrektség**
  - konstansok értéke nem változtatható meg. Deklarációkor definiálnunk kell az értékét, onnantól kezdve a program futása során nem változtatható meg. Fordítási időben értékelődik ki.

## 2. feladat - helyes-e az alábbi program? Ha igen miért, ha nem miért?

```
int* const* ppl;  
int* const** const pppl = &ppl;
```

- Helyes.

Az első sorban egy int-re mutató konstans mutatóra mutató mutató van.

A második sornál egy int-re mutató konstans mutató mutató mutatóra mutató konstans mutató van. Az értékadás rendben van.

## 3. feladat - írd meg a swap függvényt referencia szerinti érték átadással

```
void swap(int& i, int& j)  
{  
    int tmp = i;  
    i = j;  
    j = tmp;  
}
```