

```
#define FALSE 0
#define TRUE 1

typedef char Name[ 50 ];
typedef unsigned long ID;
typedef float TotalPayments;
typedef char BOOL;

typedef struct
{
    Name name;
    ID id;
    TotalPayments totalpayments;
    BOOL win;
} Person;

// Ellenőrizni, hogy az aktuális személy befizetése magasabb, mint a megadott érték.
BOOL check(Person const person, TotalPayments const currentmax)
{
    return person.totalpayments > currentmax;
}

// Kiválasztani, hogy melyik személy fizette be a legtöbbet, a legelső találat nyer.
ID selectwinner(unsigned const size, Person database[ size ])
{
    [ Kiválaszt ] current = 0;
    for(unsigned i = 0; i < size;)
    {
        if( [ Kiválaszt ] )
        {
            [ Kiválaszt ] = FALSE;
            [ Kiválaszt ] = TRUE;
            [ Kiválaszt ] ;
        } else {
            ++i;
        }
    }
    return [ Kiválaszt ] ;
}
```

Helyes!

1. válasz:

person.totalpayments

Helyes!

2. válasz:

>

Megadott válasz

3. válasz:

float

Helyes válasz

4. válasz:

unsigned

Megadott válasz

5. válasz:

check(database[i]) > 0

Helyes válasz

6. válasz:

check(database[i], database[current].totalpayments)

Megadott válasz

7. válasz:

database[i]

Helyes válasz	database[current].win
	6. válasz:
Megadott válasz	database[current].win
Helyes válasz	database[i].win
	7. válasz:
Helyes válasz	current = i++
Megadott válasz	current = ++i
	8. válasz:
Megadott válasz	database[i].win
Helyes válasz	database[current].id

2. kérdés1 / 1 pont

Az előző feladatban látható kódot milyen módon tudjuk hatékonyabbá, olvashatóbbá, jobbra tenni?

Figyelem, több válasz is lehetséges (rossz válasz pontlevonást jelent)!

Helyes!

☒ A for ciklusba helyezzük a ciklusváltozó inkrementálását. Ezzel egyértelműsítjük is a ciklusváltozó növelésének helyét (a ciklustörzs lefutása végén), valamint a postfix és prefix inkrementálás közötti különbségek eltűnnek.

☐ A BOOL typedef-ben int-re módosítjuk a char típust, ezzel csökkentve a Person struct-ban a padding-t.

Helyes!

☒ A check függvény törzsét a felhasználási helyre másoljuk vagy inline kulcsszót (C99 verziótól) teszünk elé, hogy a compiler bemásolja a függvény törzsét a hívás helyére. Ezzel a stack-n foglalt memóriát csökkentjük és a programunknak sem kell különböző memóriacímek között ugrálnia.

☐ A TRUE és FALSE makrókat töröljük és a használatuk helyén 0 és 1 értékeket alkalmazunk értelemszerűen. Ezzel a preprocesszor feladatát csökkentjük és gyorsítjuk a fordítás folyamatát.