

**Feladat 1:** Próbáljuk ki a következő kódot és magyarázzuk meg a hiba okát.

```
#include<stdio.h>
int main()
{
    {
        int a = 10, b = 20;
    }
    {
        printf("a = %d, b = %d\n", a, b);
    }
}
```

Megoldás: az *a* és *b* változók nem léteznek a második blokkban.

---

**Feladat 2:** Figyeljük meg, hogyan kezeli a **C** nyelv a blokkokat. Két kapcsos zárójel közötti utasításokat 1 blokknak kell tekinteni. Mi lesz a kimenete a következő kódnak és miért?

```
#include <stdio.h>
int main()
{
    int n = 1;
    {
        int n = 2;
        printf("%d\n", n);
    }
    printf("%d\n", n);
}
```

Megoldás: hiába azonosak a változónevek *n*, a C nyelv a blokkon belül egyedileg kezeli, így a külső *n* más értéket fog tartalmazni.

---

**Feladat 3:** Miért nem növekszik az *x* minden ciklusban?

```
#include <stdio.h>
int main() {
    for (int i = 0; i < 3; i++) {
        int x = 0;
        x++;
        printf("%d ", x);
    }
}
```

```
    return 0;  
}
```

Megoldás: mert a ciklusban `int x = 0;` mindig újra inicializálja.

---

**Feladat 4:** Milyen sorrendben írja ki a számokat?

```
#include <stdio.h>  
int main() {  
    int x = 1;  
    {  
        x = 2;  
        {  
            int x = 3;  
            printf("%d ", x);  
        }  
        printf("%d ", x);  
    }  
    printf("%d\n", x);  
}
```

Megoldás: 3, 2, 2

---

**Feladat 5:** Miért kapunk hibás eredményt?

```
#include <stdio.h>  
int main() {  
    int n = 5;  
    {  
        int n = n + 1;  
        printf("%d\n", n);  
    }  
}
```

Megoldás: undefined behavior (nem definiált viselkedés). A C szabályai szerint az új változó inicializálása előtt még nem létezik, tehát a `n` a jobb oldalon még nem a külső `n`-t, hanem az éppen most definiálás alatt álló (még nem inicializált) `n`-t jelenti

---

**Feladat 6:** Mi lesz a két kiírt szám és miért?

```
#include <stdio.h>
int main() {
    int a = 5;
    int b = 10;
    {
        int b = a + 1;
        printf("%d ", b);
    }
    printf("%d\n", b);
}
```

Megoldás: 6, 10

---

**Feladat 7:** Miért változik meg az a értéke, pedig a módosítás egy belső blokkban történt?

```
#include <stdio.h>
int main() {
    int a = 5;
    {
        int *p = &a;
        *p = 10;
    }
    printf("%d\n", a);
}
```

Megoldás: a pointer *p* közvetlenül a memóriába ír.

---

**Feladat 8:** Melyik sor melyik *x*-re hivatkozik?

```
#include <stdio.h>
int x = 100;

int main() {
    int x = 10;
    {
        extern int x;
        printf("%d ", x);
    }
    printf("%d\n", x);
}
```

}

Megoldás: extern int x; a globális változóra vonatkozik

From:

<https://edu.iit.uni-miskolc.hu/> - **Institute of Information Science - University of Miskolc**

Permanent link:

[https://edu.iit.uni-miskolc.hu/tanszek:oktatas:szamitastechnika:blokkok\\_-\\_ciklusok](https://edu.iit.uni-miskolc.hu/tanszek:oktatas:szamitastechnika:blokkok_-_ciklusok)

Last update: **2025/10/29 07:45**

