Tartalomjegyzék

1. bug0.c

```c
#include <stdio.h>

int main()
{
    printf("hello\n")

    return 0;
}
```

2. bug1.c

```
#include <stdio.h>

int main()
{
    string s = get_string("Mi␣a␣neved?\n");

    printf("Hello␣%s!\n", s);

    return 0;
}
```

3. bug2a.c

```c
#include <stdio.h>

// Irassuk ki a '*' karaktert 5-szor.

int main()
{
    for (int i = 0; i <= 5; ++i)
    {
        printf("*\n");
    }

    return 0;
}
```

4. bug2b.c

```c
#include <stdio.h>

// Irassuk ki a '*' karaktert 5-szor.

int main()
{
    for (int i = 0; i <= 5; ++i)
    {
        printf("# i erteke: %d\n", i);
        printf("*\n");
    }

    return 0;
}
```

5. bug2c.c

```c
#include <stdio.h>

// Irassuk ki a '*' karaktert 5-szor.
// Hasznaljunk debugger-t.

int main()
{
    for (int i = 0; i <= 5; ++i)
    {
        printf("*\n");
    }

    return 0;
}
```

6. const.c

```
#include <stdio.h>

int terulet(const int a, const int b)
{
    return a * b;
}

int main()
{
    const int a = 3;
    const int b = 2;

    const int t = terulet(a, b);
    printf("terulet: %d\n", t);

    return 0;
}
```

7. hello0.c

```c
#include <stdio.h>

int main()
{
    printf("hello␣world\n");

    return 0;
}
```

8. hello1.c

```
#include "prog1.h"
#include <stdio.h>

int main()
{
    string s = get_string("Mi␣a␣neved?\n");

    printf("Hello␣%s!\n", s);

    return 0;
}
```

9. style0.c

```
#include <stdio.h>
int main(){
  printf("hello\n");
return 0;}
```

10. style1.c

```c
#include <stdio.h>

int main()
{
    printf("hello\n");

    return 0;
}
```

11. sum.kt.c

```
fun sum(x: Int, y: Int): Int
{
    x = 5;     // Hiba! x read-only
    return x + y
}

fun main()
{
    println(sum(1, 2))
}
```

12. terulet.rs.c

```
// Rust pelda.
// Rust-ban a read-only mod az alapertelmezes.
// Ha valamit modosithatova akarunk tenni, akkor
// azt kulon jeleznunk kell.

fn terulet(a: i32, b: i32) -> i32
{
    return a * b;
}

fn main()
{
    let mut a: i32 = 3;
    let b: i32 = 2;

    a = 4;
    b = 5;    // Hiba! b immutable

    let t = terulet(a, b);
    println!("terulet: {}", t);
}
```

13. tomb1.c

```c
#include <stdio.h>

void szoroz_10_zel(int n, int tomb[])
{
    for (int i = 0; i < n; ++i)
    {
        tomb[i] *= 10;     // tomb[i] = tomb[i] * 10;
    }
}

int main()
{
    int szamok[4] = { 1, 4, 8, 3 };
    int meret = 4;

    szoroz_10_zel(meret, szamok);
    printf("%d\n", szamok[0]);
    printf("%d\n", szamok[1]);
    printf("%d\n", szamok[2]);
    printf("%d\n", szamok[3]);

    return 0;
}
```

14. tomb2.c

```c
#include <stdio.h>

int sum(int n, const int tomb[])
{
    int osszeg = 0;
    for (int i = 0; i < n; ++i)
    {
        osszeg += tomb[i];
    }

    return osszeg;
}

int main()
{
    int szamok[4] = { 1, 4, 8, 3 };
    int meret = 4;

    int osszeg = sum(meret, szamok);
    printf("osszeg: %d\n", osszeg);
    //
    printf("%d\n", szamok[0]);
    printf("%d\n", szamok[1]);
    printf("%d\n", szamok[2]);
    printf("%d\n", szamok[3]);

    return 0;
}
```

15. tomb3.c

```c
#include <stdio.h>

int sum(const int n, const int tomb[])
{
    int osszeg = 0;
    for (int i = 0; i < n; ++i)
    {
        osszeg += tomb[i];
    }

    return osszeg;
}

int main()
{
    int szamok[4] = { 1, 4, 8, 3 };
    int meret = 4;

    const int osszeg = sum(meret, szamok);

    // osszeg = 20;    // Hiba!

    printf("osszeg: %d\n", osszeg);

    return 0;
}
```