

Tartalomjegyzék

| | |
|------------------------------|----|
| 1. array_to_array.c | 2 |
| 2. kiir.c | 3 |
| 3. mem_szemet1.c | 4 |
| 4. mem_szemet2.c | 5 |
| 5. osszead0.c | 6 |
| 6. osszead1.c | 7 |
| 7. osszead2.c | 8 |
| 8. osszead3.c | 9 |
| 9. osszead4.c | 10 |
| 10.out_of_bounds.c | 11 |
| 11.parameteratadas.c | 12 |
| 12.prog1.c | 13 |
| 13.prog1.h | 14 |
| 14.szerencsetlen_osszeadas.c | 15 |

1. array_to_array.c

```
#include <stdio.h>

int main()
{
    int t1[3] = { 1, 2, 3 };
    int t2[3];

    // t2 = t1;

    for (int i = 0; i < 3; ++i)
    {
        t2[i] = t1[i];
    }

    return 0;
}
```

2. kiir.c

```
#include <stdio.h>

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

    puts("world");

    putchar('A');
    putchar('B');
    putchar('B');
    putchar('A');
    putchar('\n');

    puts("");

    printf("__END__");

    return 0;
}
```

3. mem_szemet1.c

```
#include <stdio.h>

int main()
{
    int x;

    // Hiba! Nem adtunk
    // neki kezdőértéket!
    printf("%d\n", x);

    return 0;
}
```

4. mem_szemet2.c

```
#include <stdio.h>

int main()
{
    int counter;

    for (int i = 0; i < 10; ++i)
    {
        // Hiba! Ugy noveljuk az erteket, hogy nem adtunk
        // neki kezdoerteket!
        counter += 1;
    }

    printf("%d\n", counter);

    return 0;
}
```

5. osszead0.c

```
#include <stdio.h>

int add_two_ints(int a, int b)
{
    return a + b;
}

int main()
{
    int n1 = 2;
    int n2 = 8;

    // TODO: add_two_ints() meghivása

    return 0;
}
```

6. osszead1.c

```
#include <stdio.h>

int add_two_ints(int a, int b)
{
    return a + b;
}

int main()
{
    int n1 = 2;
    int n2 = 8;

    int result = add_two_ints(n1, n2);
    printf("Eredmeny: □%d\n", result);

    return 0;
}
```

7. osszead2.c

```
#include <stdio.h>

int add_two_ints(int a, int b)
{
    return a + b;
}

int main()
{
    int n1 = 2;
    int n2 = 8;

    printf("Eredmeny: □%d\n", add_two_ints(n1, n2));

    return 0;
}
```


8. osszead3.c

```
#include <stdio.h>

// Probaljuk lefordítani. Mi a hiba?

int main()
{
    int n1 = 2;
    int n2 = 8;

    printf("Eredmeny: □%d\n", add_two_ints(n1, n2));

    return 0;
}

int add_two_ints(int a, int b)
{
    return a + b;
}
```

9. osszead4.c

```
#include <stdio.h>

int add_two_ints(int a, int b);    // elodeklaracio

int main()
{
    int n1 = 2;
    int n2 = 8;

    printf("Eredmeny: □%d\n", add_two_ints(n1, n2));

    return 0;
}

int add_two_ints(int a, int b)
{
    return a + b;
}
```

10. out_of_bounds.c

```
#include <stdio.h>

int main()
{
    int tomb[10];

    printf("%d\n", tomb[14]);

    return 0;
}
```

11. parameteratadas.c

```
#include <stdio.h>

void set_int(int x)
{
    x = 42;
}

void set_array(int array[])
{
    array[0] = 22;
}

int main()
{
    int a = 10;
    int b[4] = { 0, 1, 2, 3 };
    set_int(a);
    set_array(b);
    printf("%d, %d\n", a, b[0]);
    return 0;
}
```

12. prog1.c

```
#include "prog1.h"

//
// //////////////////////////////////////
//
//      Implementation
//

#include <stdio.h>
#include <string.h>

const int BUFSIZE = 1024;

/**
 * Ez egy dinamikusan lefoglalt sztringet ad vissza,
 * amit a hívó oldalon majd valamikor fel kell szabadítani.
 */
string get_string(const string prompt)
{
    char buf[BUFSIZE];

    printf("%s", prompt);
    fgets(buf, sizeof(buf), stdin);
    buf[strlen(buf)-1] = '\0';

    return strdup(buf);
}
```

13. prog1.h

```
#ifndef PROG1_H
#define PROG1_H

/**
 * Our own type for (pointers to) strings.
 */
typedef char * string;

//
// //////////////////////////////////////
//
// Public Interface
//

string get_string(const string prompt);

#endif    // PROG1_H
```

14. szerencsetlen_osszeadas.c

```
#include <stdio.h>

// Ez csak egy pelda arra, hogy egy fuggvenyt tobbfelekeppen is
// implementalhatunk. Az itt bemutatott modszert nem szep, a
// gyakorlatban ket szamot NE igy adjunk össze!

int add_two_ints(int a, int b)
{
    int sum = 0;
    while (a > 0) {
        ++sum;
        --a;
    }
    while (a < 0) {
        --sum;
        ++a;
    }
    while (b > 0) {
        ++sum;
        --b;
    }
    while (b < 0) {
        --sum;
        ++b;
    }
    return sum;
}

int main()
{
    int n1 = 16;
    int n2 = -4;

    int result = add_two_ints(n1, n2);
    printf("Eredmeny: %d\n", result);

    return 0;
}
```