

Przykład 1:

przykładowy program w języku pokazane

jest działanie operatorów "+", "!", "==" oraz funkcji print i assert.

```
fn main() -> int {
    int x = 5 + 3;
    bool t = true;
    bool f = !t;
    string s = "string!";
    print(x);
    // 8
    print(t);
    // true
    print(f);
    // false
    print(s);
    // string!
    assert (x == 8);
    return 0;
}
```

Przykład 2:

Przekazywanie przez wartość i referencję:

```
fn funkcja1(int &arg) -> void {
    arg++;
}
fn funkcja2(int arg) -> void {
    arg++;
}
fn main() -> int {
    int x = 5;
    funkcja1(&x);
    // x = 6
    funkcja2(x);
    // x = 6
    funkcja2(&x);
    // x = 7
}
```

Przykład 3:

Statyczne wiązanie:

```
int x = 2;
fn funkcja1(int x) -> int {
    return x + 3;
}
fn funkcja2(int z) -> int {
```

```

    return x + z;
}

fn main() -> int {
    int y = 7;
    int res = funkcja(y);
    // res = 10
    res = funkcja2(y);
    // res = 9
}

```

Przykład 4:

Funkcje zagnieżdżone wiązane statycznie:

```

fn funkcja1(int x) -> int {
    x = x + 10;
    fn funkcja2(int x) -> int {
        return x + 5;
    }
    int res = funkcja2(x);
    // res = initial value of x + 15
    return res + 1;
    // returned value is initial value of x + 16
}

int res = funkcja1(1);
// res = 17

```

Przykład 5:

Tablice wielowymiarowe:

```

fn main() -> int {
    int x = 10;
    int y = 5;
    int i = 0;
    int [][]arr1;
    while (i < x) {
        int j = 0;
        while (j < y) {
            arr1[i][j] = 5;
            j++;
        }
        i++;
    }
    // creates 10 x 5 matrix filled with fives.
    int len1 = arr1.len(1);
    // len1 = 10
    int len2 = arr1.len(2);
    // len2 = 5
}

```

```
    int dim = arr1.dim;
    // dim = 2
}
```

Przykład 6:

Operacje na stringach:

```
fn main() -> int {
    string str = "abcde";
    string b = str.charAt(1);
    // b = "b"
    string rev = str.reverse;
    // rev = "edcba"
    str.append("abcde");
    // str = "abcdeabcde"
    string cut = str.cut(0,5);
    // cut = "abcdea"
}
```

Przykład 7:

Operacje na krotkach.

```
fn main() -> int {
    tuple<int,bool,string> tup1 = <<1,true,"abc">>;
    tuple<int, tuple<int,bool>> tup2 = <<1,<<3,false>>>>;
    <<i,b,s>> = tup1;
    // i = 1 ; b = true ; s = "abc"
}
```