

Sisteme incorporate

Limbajul de programare C

Declaratia tipurilor:

- `int i; float f;`
 - `char name[] = "mikro foo";`
 - `int /* type */ i /* identifier */;`
 - `int i; // this is a comment`
 - `int j;`
- `char ch, str[] = "mikro";`
`int mat[3][4]; /* 3 x 4 matrix */`
`ch = str[3]; /* 4th element */`

Operatii conditionate:

```
d = c * (a + b); /* override normal precedence */
```

```
if (d == z) ++x; /* essential with conditional statement */
```

```
func(); /* function call, no args */
```

```
void func2(int n); /* function declaration with parameters */
```

```
if (d == z) { ++x; func(); }
```

```
if (statement) { ... }; /* illegal semicolon! */ else { ...  
};
```

```
start: x = 0;
```

```
...
```

```
goto start;
```

```
int test[5] = { 1, 2, 3, 4, 5 };
```

```
int x = 5;
```

```
int a, b, c;
```

```
a = b + c;
```

Hexadecimal: 0xC367

Binary: 0b11101

Octal: 0777

Value Assigned to Constant

Assumed Type

< -2147483648

Error: Out of range!

-2147483648 – -32769

long

-32768 – -129

int

-128 – 127

short

128 – 255

unsigned short

256 – 32767

int

32768 – 65535

unsigned int

65536 – 2147483647

long

2147483648 – 4294967295

unsigned long

> 4294967295

Error: Out of range!

Float

$\pm 1.17549435082 * 10^{-38} \dots \pm 6.80564774407 * 10^{38}$

`0. // = 0.0`

`-1.23 // = -1.23`

`23.45e6 // = 23.45 * 106`

`2e-5 // = 2.0 * 10-5`

`3E+10 // = 3.0 * 1010`

`.09E34 // = 0.09 * 1034`

Operanzi:

- asm
- auto
- break
- case
- char
- const
- continue
- default
- do
- double
- else
- enum
- extern
- float
- for
- goto
- if
- int
- long
- register
- return
- short
- signed
- sizeof
- static
- struct
- switch
- typedef
- union
- unsigned
- void
- volatile
- while

Opernazi:

| Operator | Operation | Precedence |
|-------------------------|---|------------|
| Binary Operators | | |
| + | addition | 12 |
| - | subtraction | 12 |
| * | multiplication | 13 |
| / | division | 13 |
| % | modulus operator returns the remainder of integer division (cannot be used with floating points) | 13 |
| Unary Operators | | |
| + | unary plus does not affect the operand | 14 |
| - | unary minus changes the sign of operand | 14 |
| ++ | increment adds one to the value of the operand. Postincrement adds one to the value of the operand after it evaluates; while preincrement adds one before it evaluates | 14 |
| -- | decrement subtracts one from the value of the operand. Postdecrement subtracts one from the value of the operand after it evaluates; while predecrement subtracts one before it evaluates | 14 |

Operanzi:

| Operator | Operation | Precedence |
|----------|-----------------------|------------|
| == | equal | 9 |
| != | not equal | 9 |
| > | greater than | 10 |
| < | less than | 10 |
| >= | greater than or equal | 10 |
| <= | less than or equal | 10 |

Operanzi:

| Operator | Operation | Precedence |
|----------|---|------------|
| & | bitwise AND; compares pairs of bits and returns 1 if both bits are 1, otherwise returns 0 | 8 |
| | bitwise (inclusive) OR; compares pairs of bits and returns 1 if either or both bits are 1, otherwise returns 0 | 6 |
| ^ | bitwise exclusive OR (XOR); compares pairs of bits and returns 1 if the bits are complementary, otherwise returns 0 | 7 |
| ~ | bitwise complement (unary); inverts each bit | 14 |
| << | bitwise shift left; moves the bits to the left, discards the far left bit and assigns 0 to the right most bit. | 11 |
| >> | bitwise shift right; moves the bits to the right, discards the far right bit and if unsigned assigns 0 to the left most bit, otherwise sign extends | 11 |

Operanzi logici:

| Operator | Operation | Precedence |
|----------|------------------|------------|
| && | logical AND | 5 |
| | logical OR | 4 |
| ! | logical negation | 14 |

If ???

```
if (expression1) statement1  
else if (expression2)  
if (expression3) statement2  
else statement3 /* this belongs to: if (expression3) */  
else statement4 /* this belongs to: if (expression2) */
```

Switch ???

```
switch (i) {  
  case 1: i++;  
  case 2: i++;  
  case 3: i++;  
}
```

```
switch (phase) {  
  case 0: Lo(); break;  
  case 1: Mid(); break;  
  case 2: Hi(); break;  
  default: Message("Invalid state!");  
}
```

While ??

```
int s = 0, i = 0;  
while (i < n)  
{  
    s += a[i] * b[i];  
    i++;  
}
```

DO ???

```
s = 0;
```

```
i = 0;
```

```
do { s += a[i] * b[i]; i++; }
```

```
while ( i < n );
```

For ???

- `for (s = 0, i = 0; i < n; i++) s += a[i] * b[i];`
- `for (s = 0, i = 0; i < n; s += a[i] * b[i], i++);`
/ valid, but ugly */*

Bibliotece:

- [ADC Library](#)
- [CAN Library](#)
- [CANSPI Library](#)
- [Compact Flash Library](#)
- [EEPROM Library](#)
- [Ethernet Library](#)
- [Flash Memory Library](#)
- [Graphic LCD Library](#)
- [I²C Library](#)
- [Keypad Library](#)
- [LCD Library](#)
- [LCD8 Library](#)
- [Manchester Code Library](#)
- [Multi Media Card Library](#)
- [OneWire Library](#)
- [PS/2 Library](#)
- [PWM Library](#)
- [RS-485 Library](#)
- [Secure Digital Library](#)
- [Software I²C Library](#)
- [Software SPI Library](#)
- [Software UART Library](#)
- [Sound Library](#)
- [SPI Library](#)
- [USART Library](#)
- [USB HID Library](#)
- [Util Library](#)

Conversia tipurilor de date:

- [ByteToStr](#)
- [ShortToStr](#)
- [WordToStr](#)
- [IntToStr](#)
- [LongToStr](#)
- [FloatToStr](#)