

# Struktura programov v C-ju

---

- komentarji: `//`, `/* */`

```
/* a */
```

```
/* a /* a */
```

```
/* a /* a */ a */
```

 komentar so prvi trije

- ukazi predprocesorja: `#include` (kar se začne z `#` ni del c-ja) c prevajalnik kliče predprocesor ki naredi drugačno datoteko (doda header file content itd.) in jo da prevajalniku
- deklaracije in definicije: tipov, funkcij, spremenljivk

```
#include <stdio.h>
int collatz(int n){
    if(n % 2 == 0) return n/2;
    else return n*3 + 1;
}

int length(int n){
    int len = 1;

    while(n != 1){
        n = collatz(n);
        len++;
    }

    return len;
}

int main(){
    int n = 6;
    printf("%d\n", length(n));
    return 0;
}
```

funkcije ki jih uporabimo v naši funkciji se morajo nahajati nad našo funkcijo ko prevedemo funkcijo si zapomne samo definicijo v ramu da sprostimo ram

razen če jih prej deklariramo (prototipi):

```
#include <stdio.h>

int length(int);
int collatz(int);
```

```

int main(){
    int n = 6;
    printf("%d\n", length(n));
    return 0;
}

int collatz(int n){
    if(n % 2 == 0) return n/2;
    else return n*3 + 1;
}

int length(int n){
    int len = 1;

    while(n != 1){
        n = collatz(n);
        len++;
    }

    return len;
}

```

pomnilnik med izvajanjem in pomnilnik med izvajanjem

prototipi:

```

int collatz(int n);
int collatz(int);
int collatz(int num); // izogibamo
int collatz(); // izogibamo

```

ni overloadanja istih poimenovanj funkcij

int main(void)

parameter je spremenljivka, argument je izraz

Kontrolni stavki:

- if-then-else
- while-do
- do-while
- for
- switch-case-default
- break-continue
- return
- goto

else vedno spada k najbližjemu ifu:

```

if(n == 2)
    if(m == 3)
        n = n + m;
else n = 0;

// je ekvivalentno:

if(n == 2){
    if(m == 3){
        n = n + m;
    }
    else{
        n = 0;
    }
}

```

ni argumentov pri break, continue; namesto tega uporabljamo goto:

```

while(i > 0){
    ...
    while(j < 0){
        ...
        break;
    }
}

// ekvivalentno:

while(i > 0){
    ...
    while(j < 0){
        ...
        goto outer;
    }
outer:
}

```

```

int collatz(int n){
    if(n % 2 == 0) goto even;
    return n * 3 + 1;
even:
    return n/2;
}

int length(int n){
    int len = 1;
loop:
    if(n == 1) goto result;
}

```

```
    n = collatz(n);  
    len++;  
    goto loop;  
result:  
    return len;  
}
```