Ciklusok C-ben

A ciklusok segítségével utasításokat ismétlünk meg. Három fő típust ismerünk:

1. for: ismétlés meghatározott számszor
2. while: ismétlés, amíg a feltétel igaz
3. do-while: ismétlés, amíg a feltétel igaz, de mindig lefut legalább egyszer (akkor is, ha a feltétel hamis)

Ezen kívül van két vezérlő utasítás:

1. break → kilép a ciklusból
2. continue → átugorja az adott iteráció további részét, és a következőre ugrik

---

for ciklus

Akkor használjuk, ha tudjuk előre, hányszor akarjuk ismételni az utasítást.

Szintaxis:

```
for (kezdőérték; feltétel; léptetés) {
    // ciklusmag
}
```

Példa:

```
#include <stdio.h>

int main(void) {
    for (int i = 1; i <= 5; i++) {
        printf("%d ", i);
    }
    return 0;
}
```

---

while ciklus

Akkor használjuk, ha nem tudjuk előre, hányszor kell ismételni, de van egy feltétel.

Szintaxis:

```
while (feltétel) {
    // ciklusmag
}
```

Példa:

```
#include <stdio.h>

int main(void) {
    int i = 1;
    while (i <= 5) {
        printf("%d ", i);
        i++;
    }
    return 0;
}
```

---

do-while ciklus

Ez mindig lefut legalább egyszer, mert a feltételt a ciklus végén ellenőrzi.

Szintaxis:

```
do {
    // ciklusmag
} while (feltétel);
```

Példa:

```
#include <stdio.h>

int main(void) {
    int i = 1;
    do {
        printf("%d ", i);
        i++;
    } while (i <= 5);
    return 0;
}
```

---

break utasítás

A break azonnal kilép a legbelső ciklusból.

Példa: első 1-től induló szám, ami osztható 7-tel

```
#include <stdio.h>

int main(void) {
    for (int i = 1; i <= 100; i++) {
        if (i % 7 == 0) {
            printf("Találtam: %d\n", i);
            break;  // kilép a ciklusból
        }
    }
    return 0;
}
```

---

continue utasítás

A continue kihagyja az aktuális iteráció hátralévő részét, és a következőre ugrik.

Példa: 1-től 10-ig, de a páros számokat kihagyjuk

```
#include <stdio.h>

int main(void) {
    for (int i = 1; i <= 10; i++) {
        if (i % 2 == 0) {
            continue;  // párosakat átugorjuk
        }
        printf("%d ", i);
    }
    return 0;
}
```

Gyakorlás

1. Írj programot, ami beolvas egy N számot, és kiszámolja az 1+2+…+N összeget for ciklussal.

```
#include <stdio.h>

int main(void) {
    int N, sum = 0;
    scanf("%d", &N);

    for (int i = 1; i <= N; i++) {
        sum += i;
    }
```

```
    printf("Osszeg: %d\n", sum);
    return 0;
}
```

2. Kérj be egy pozitív számot, és számold ki a számjegyeinek összegét (pl. 123 → 1+2+3=6).

```
#include <stdio.h>

int main(void) {
    int x, sum = 0;
    scanf("%d", &x);

    while (x > 0) {
        sum += x % 10;  // utolsó számjegy
        x /= 10;        // levágjuk a számjegyet
    }

    printf("Szamjegyek osszege: %d\n", sum);
    return 0;
}
```

3. A program kérjen be számokat addig, amíg a felhasználó nem találja el a titkos számot (pl. 7). Minden tipp után írd ki: „Túl kicsi” vagy „Túl nagy”.

```
#include <stdio.h>

int main(void) {
    int tipp;
    int titkos = 7;

    do {
        printf("Adj meg egy szamot: ");
        scanf("%d", &tipp);

        if (tipp < titkos) {
            printf("Tul kicsi!\n");
        } else if (tipp > titkos) {
            printf("Tul nagy!\n");
        }
    } while (tipp != titkos);

    printf("Eltalaltad!\n");
    return 0;
}
```

4. Olvass be egy számot N, majd írd ki az összes páros számot 1 és N között. Használj continue-t a páratlanok átugrására.

```
#include <stdio.h>

int main(void) {
    int N;
    scanf("%d", &N);

    for (int i = 1; i <= N; i++) {
        if (i % 2 != 0) continue;  // ha páratlan, átugorjuk
        printf("%d ", i);
    }
    printf("\n");
    return 0;
}
```

5. Olvass be egy N számot, és számold ki az N! értékét (pl. 5! = 120) for ciklussal. (folyamatábra)

```
#include <stdio.h>

int main(void) {
    int N;
    long long fakt = 1;  // nagyobb számokra is jó legyen

    scanf("%d", &N);

    for (int i = 1; i <= N; i++) {
        fakt *= i;
    }

    printf("%d! = %lld\n", N, fakt);
    return 0;
}
```

6. Olvass be eg N számot, és írd ki a Fibonacci sorozat N elemét for ciklussal. (folyamatábra)

```
#include <stdio.h>

int main(void) {
    int N;
    scanf("%d", &N);

    int a = 0, b = 1;
```

```
    printf("Fibonacci sorozat: ");
    for (int i = 0; i < N; i++) {
        printf("%d ", a);
        int kov = a + b;
        a = b;
        b = kov;
    }
    printf("\n");

    return 0;
}
```

From:
https://edu.iit.uni-miskolc.hu/ - **Institute of Information Science - University of Miskolc**

Permanent link:
https://edu.iit.uni-miskolc.hu/tanszek:oktatas:szamitastechnika:ciklusok

Last update: **2025/10/01 12:42**