Pszeudokód	**C++ nyelv**
`Beolvas n`	`cin>>n;`
`Kiír n`	`cout<<n;`
Értékadás `a ← 1`	`a=1;`

Feltételes utasítás

`Ha feltétel akkor`
`│ utasítás`
`■`
	`if (feltétel)`
	`  utasítás;`

`Ha feltétel akkor`
`│ akkor utasítás1`
`│ különben utasítás2`
`■`
	`if (feltétel)`
	`  utasítás1;`
	`else`
	`  utasítás2;`

Ismétlő utasítások - ciklusok

1. Elöltesztelő utasítás:

`Amíg feltétel végezd el`
`│ utasítások`
`■`
	`while (feltétel)`
	`{`
	`  utasítások`
	`}`
		`← ciklusmag`

Működési elve:

1. Ellenőrzi a feltételt.
2. Ha a feltétel IGAZ (C/C++ 1 vagy bármely nullától különböző érték), akkor belép a ciklusba és elvégzi az itt található műveleteket (ciklusmagot), majd visszalép az 1. pontra (újra ellenőrzi a feltételt).
3. Ha a feltétel HAMIS (C/C++ 0), akkor kilép a ciklusból (a következő utasítással folytatja a programot).

Mivel a feltételek vizsgálata az utasítások elvégzése előtt van, sajátos esetben lehet hogy a ciklusmag egyszer sem fut le (0, 1 vagy több alkalommal lesz végrehajtva a ciklusmag).

2. Hátultesztelő utasítás:

`Ismételd`
`│ utasítások`
`Ameddig feltétel`
	`do`
	`{`
	`  utasítások`
	`} while !(feltétel);`
	FIGYELEM! Átírásnál ide a pszeudokódos feltétel tagadását kell írni.
		`← ciklusmag`

Működési elve PSZEUDOKÓDBAN:

1. Elvégzi a ciklusmagban található utasításokat.
2. Ellenőrzi a feltételt.
3. Ha a feltétel HAMIS, akkor visszalép az 1. pontra (újra elvégzi az utasításokat).
4. Ha a feltétel IGAZ, akkor kilép a ciklusból (a következő utasítással folytatja a programot).

Mivel a feltételek vizsgálata az utasítások elvégzése után van, a ciklusmag legalább egyszer mindenképp lefut. (legalább egyszer mindenképp végre lesz hajtva a ciklusmag)

Működési elve C/C++-ban:

1. Elvégzi a ciklusmagban található utasításokat.
2. Ellenőrzi a feltételt.
3. Ha a feltétel 1 vagy bármely nullától különböző érték (igaz), akkor visszalép az 1. pontra (újra elvégzi az utasításokat).
4. Ha a feltétel 0 (hamis), akkor kilép a ciklusból (a következő utasítással folytatja a programot).

3. Ismert lépésszámú - számlálós – ciklus

a. növekvő

Minden i ← ké, vé -re végezd el
| utasítások
■
	for(i=ké; i<=vé; i++)
	{
	utasítások
	}

Működési elve:

1. Az *i* változóba beírja a kezdőértéket (*ké*).
2. Ellenőrzi, hogy az *i* kisebb vagy egyenlő-e mint a végső érték (*vé*), ha teljesül a feltétel akkor a 3. ponthoz lép különben a 6. ponthoz.
3. Végrehajtja a ciklusmagban szereplő utasításokat.
4. Az *i* értékét megnöveli 1-el.
5. A 2-es ponthoz lép (újra ellenőrzi a feltételt).
6. Ha az *i* meghaladta a végső értéket, akkor kilép a ciklusból (a következő utasítással folytatja a programot).

Sajátos esetben, ha a kezdőérték>végsőérték, akkor a ciklusmagban szereplő utasításokat egyszer sem hajtja végre (kezdőérték=végsőérték egy végrehajtást jelent).

b. csökkenő

Minden i ← ké, vé, -1 -re végezd el
| utasítások
■
	for(i=ké; i>=vé; i--)
	{
	utasítások
	}

Működési elve:

1. Az *i* változóba beírja a kezdőértéket (*ké*).
2. Ellenőrzi, hogy az *i* nagyobb vagy egyenlő-e mint a végső érték (*vé*), ha teljesül a feltétel akkor a 3. ponthoz lép különben a 6. ponthoz.
3. Végrehajtja a ciklusmagban szereplő utasításokat.
4. Az *i* értékét csökkenti 1-el.
5. A 2-es ponthoz lép (újra ellenőrzi a feltételt).
6. Ha az *i* kisebb lett, mint a végső érték (*vé*), akkor kilép a ciklusból (a következő utasítással folytatja a programot).

Sajátos esetben, ha a kezdőérték<végsőérték, akkor a ciklusmagban szereplő utasításokat egyszer sem hajtja végre (kezdőérték=végsőérték egy végrehajtást jelent).