

# «ԾՐԱԳՐԱՎՈՐՄԱՆ ՀԻՄՈՒՆՔՆԵՐ» դասընթաց

այլ ոլորտներից դեպի տեխնոլոգիական ոլորտ  
սկսնակների համար



edu2020.am

## ԴԱՍ #6



ՀԱՅ-ՌՈՒՄԱԿԱՆ  
ՀԱՄԱԼՍԱՐԱՆ



ՀԱՅԱՍՏԱՆԻ ՀԱՆՐԱՊԵՏՈՒԹՅԱՆ  
ԲԱՐՁՐ ՏԵԽՆՈԼՈԳԻԱԿԱՆ  
ԱՐԴՅՈՒՆԱԲԵՐՈՒԹՅԱՆ ՆԱԽԱՐԱՐՈՒԹՅՈՒՆ

# for ցիկլ

Քանի դեռ **expression**-ի արժեքը *false կամ 0* չէ կատարել **statement1**, **statement2**, ...:

1. **for\_statement1** կատարվում է մեկ անգամ մինչև **ցիկլի մարմնի** կատարվելը
2. **for\_statement2** կատարվում է ամեն անգամ **ցիկլի մարմնի** կատարվելուց հետո

```
for (for_statement1; expression; for_statement2) {  
    statement1;  
    statement2;  
    ....  
}
```



# for ցիկլ

Քանի դեռ **expression**-ի արժեքը *false կամ 0* չէ կատարել **statement1, statement2, ...**:

1. **for\_statement1** կատարվում է մեկ անգամ մինչև **ցիկլի մարմնի** կատարվելը
2. **for\_statement2** կատարվում է ամեն անգամ **ցիկլի մարմնի** կատարվելուց հետո

```
for (for_statement1; expression; for_statement2) {  
    statement1;  
    statement2;  
    ....  
}
```

```
#include <iostream>  
int main() {  
    int i;  
    for (i = 0; i < 10; i += 1) {  
        std::cout << i << std::endl;  
    }  
}
```



0  
1  
2  
3  
4  
5  
6  
7  
8  
9

<https://repl.it/@SevakRAU/ForEx1>





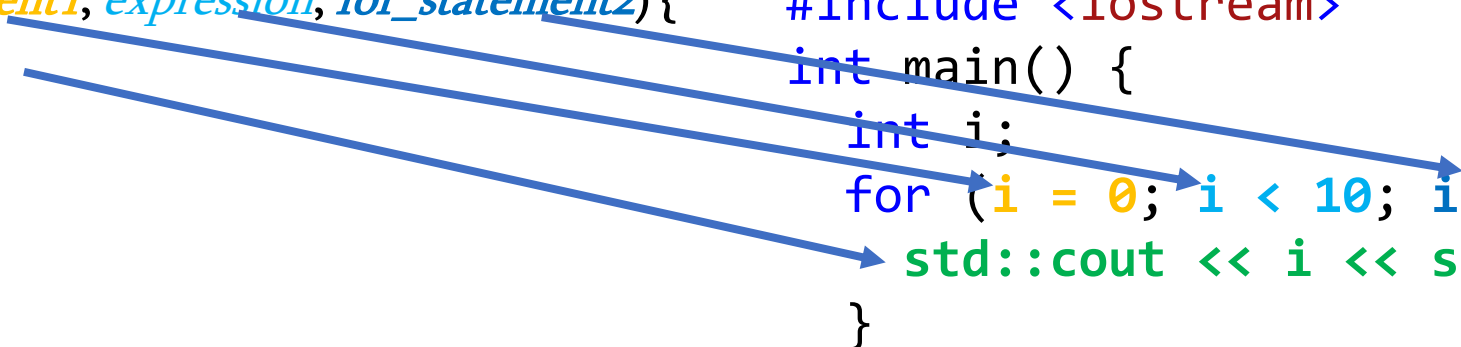
# for ցիկլ

Քանի դեռ **expression**-ի արժեքը *false կամ 0* չէ կատարել **statement1, statement2, ...**:

1. **for\_statement1** կատարվում է մեկ անգամ մինչև **ցիկլի մարմնի** կատարվելը
2. **for\_statement2** կատարվում է ամեն անգամ **ցիկլի մարմնի** կատարվելուց հետո

```
for (for_statement1; expression; for_statement2) {  
    statement1;  
    statement2;  
    ....  
}
```

```
#include <iostream>  
int main() {  
    int i;  
    for (i = 0; i < 10; i += 1) {  
        std::cout << i << std::endl;  
    }  
}
```



0  
1  
2  
3  
4  
5  
6  
7  
8  
9

<https://repl.it/@SevakRAU/ForEx1>



# for ցիկլի առանձնահատկություններ

```
#include <iostream>
int main() {
    for (;;) {
        std::cout << "infinite ..." << std::endl;
    }
}
```

Կարելի է դատարկ թողնել

Եթե դատարի պայման չի նշվում  
կատանանք անվերջ ցիկլ

<https://repl.it/@SevakRAU/ForInf1>



# for ցիկլի առանձնահատկություններ

```
#include <iostream>
int main() {
    int i;
    for (i = 0; i < 10;) {
        std::cout << "infinite ..." << std::endl;
    }
}
```

Անվերջ ցիկլ, քանի-որ i-ի  
արժեքը չենք մեծացնում

<https://repl.it/@SevakRAU/ForInf2>



# for ցիկլի առանձնահատկություններ

Փոփոխական կարելի է հայտարարել ցիկլի ներսում

```
#include <iostream>
int main() {
    for (int i = 0; i < 10;) {
        std::cout << i << std::endl;
        i += 1;
    }
}
```



0  
1  
2  
3  
4  
5  
6  
7  
8  
9

Անվերջ ցիկլ չէ

<https://repl.it/@SevakRAU/ForEx3>



# for ցիկլի պայմանների միավորում

Եթե մուտքագրված ամբողջ  $i$  թիվը  $[10, 20]$  միջակայքից է ապա տպել  $[i, 20]$  միջակայի ամբողջ թվերը:

```
#include <iostream>
int main() {
    int i;
    std::cin >> i;
    for (; i >= 10 && i <= 20; i += 1) {
        std::cout << i << std::endl;
    }
}
```

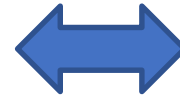
<https://repl.it/@SevakRAU/ForEx4>





# for և while ցիկլերի համարժեքություն

```
for (for_statement1; expression; for_statement2) {  
    statement1;  
    statement2;  
    ....  
}
```

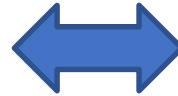


```
for_statement1;  
while(expression) {  
    statement1;  
    statement2;  
    ....  
    for_statement2;  
}
```



# for և while ցիկլերի համարժեքություն

```
#include <iostream>
int main() {
    int i;
    for (i = 0; i < 10; i += 1) {
        std::cout << i << std::endl;
    }
}
```



```
#include <iostream>
int main() {
    int i;
    i = 0;
    while (i < 10) {
        std::cout << i << std::endl;
        i += 1;
    }
}
```

<https://repl.it/@SevakRAU/ForEx1>

<https://repl.it/@SevakRAU/ForWhileEq>



# Խնդիրներ

1. Մուտքագրված է  $X$  ամբողջ թիվ  $[0, 100]$  միջակայքից: Տպել  $[0, X]$  միջակայքի բոլոր ամբողջ թվերը աճման/նվազման կարգով: *(Միջակայքին պատկանելիության ստուգումը կատարել for ցիկլում)*
2. Եթե մուտքագրված  $X$  ամբողջ թիվ  $[0, 20]$  միջակայքից: Տպել  $[1, X]$  միջակայքի բոլոր ամբողջ թվերի գումարը/արտադրյալը: *(Միջակայքին պատկանելիության ստուգումը կատարել for ցիկլում)*



# Ավելացման և նվազեցման օպերատորներ

$x = x + 1$ ; համարժեք է  $x++$ ; (postfix) կամ  $++x$ ; (prefix) գրելաձևին  
 $x = x - 1$ ; համարժեք է  $x--$ ; (postfix), կամ  $--x$ ; (prefix) գրելաձևին

```
#include <iostream>
int main() {
    int x = 1, y = 2, z = 1, k = 2;
    x++;
    y--;
    ++z;
    --k;
    std::cout << "x++ = " << x << std::endl;
    std::cout << "y-- = " << y << std::endl;
    std::cout << "++z = " << z << std::endl;
    std::cout << "--k = " << k << std::endl;
}
```



$x++ = 2$   
 $y-- = 1$   
 $++z = 2$   
 $--k = 1$

<https://repl.it/@SevakRAU/IncDecEx1>



# prefix և postfix գրելաձևի տարբերությունը

```
#include <iostream>
int main() {
    int x, y, a = 1, b = 1, c = 1;
    x = ++a;
    y = b++;
    std::cout << "x = " << x << std::endl;
    std::cout << "y = " << y << std::endl;
    std::cout << "c++ = " << c++ << std::endl;
    std::cout << "c = " << c << std::endl;
}
```

→ x = 2  
→ y = 1  
c++ = 1  
c = 2

<https://repl.it/@SevakRAU/PrePostEx2>





# prefix և postfix գրելաձևի տարբերությունը

```
#include <iostream>
int main() {
    int x, y, a = 1, b = 1, c = 1;
    x = ++a;
    y = b++;
    std::cout << "x = " << x << std::endl;
    std::cout << "y = " << y << std::endl;
    std::cout << "c++ = " << c++ << std::endl;
    std::cout << "c = " << c << std::endl;
}
```

a-ի արժեքը փոխվում է ընթացիկ  
հրամանի կատարման ժամանակ

b-ի արժեքը փոխվում է ընթացիկ  
հրամանի ավարտից հետո

x = 2

y = 1

c++ = 1

c = 2

<https://repl.it/@SevakRAU/PrePostEx2>



# Օրինակներ

```
#include <iostream>
int main() {
    // print 0-9
    for (int i = 0; i < 10;) {
        std::cout << i << std::endl;
        i++;
    }
    // print 0-9
    int i = 0;
    while (i < 10) {
        std::cout << i << std::endl;
        i++;
    }
    // print 1-10
    for (int i = 0; i++ < 10;) {
        std::cout << i << std::endl;
    }
    // print 1-10
    i = 0;
    while (i++ < 10) {
        std::cout << i << std::endl;
    }
}
```

i-ի արժեքը մեծացնելուց  
հետո է տպում

i-ի արժեքը մեծացնելուց  
հետո է տպում

<https://repl.it/@SevakRAU/LoopIncDecEx1>



# Խնդիրներ

1. Մուտքագրել 10 թիվ, դուրս բերել ամենամեծ և ամենափոքր թվերի տարբերությունը:
2. Մուտքագրել 10 թիվ, դուրս բերել առաջին 2 ամենամեծ թվերը:



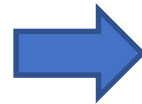
# break հրաման

- **break** հրամանը թույլ է տալիս ընդհատել ցիկլի աշխատանքը
- **break** հրամանից հետո ղեկավարումը փոխանցվում է ցիկլի անմիջապես հաջորդ հրամանին



# Օրինակ

```
#include <iostream>
int main() {
    for (int i = 0; i < 10; i++) {
        std::cout << i << std::endl;
        if (i > 5) {
            break;
        }
    }
    std::cout << "Loop end";
}
```



0  
1  
2  
3  
4  
5  
6  
Loop end

Եթե **i** փոփոխականի  
արժեքը մեծ է **5-ից**  
**դադարեցնել ցիկլի**  
**կատարումը**

<https://repl.it/@SevakRAU/ForBrEx1>

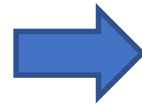




# Օրինակ

```
#include <iostream>
int main() {
    for (int i = 0; i < 10; i++) {
        std::cout << i << " ";
        if (i > 5) {
            break;
        }
    }
    std::cout << "Loop end";
}
```

Ընդհատել ցիկլի  
կատարումը



0  
1  
2  
3  
4  
5  
Loop end

<https://repl.it/@SevakRAU/ForBrEx1>



# Օրինակ

```
#include <iostream>
int main() {
    for (int i = 0; i < 10; i++) {
        std::cout << i << std::endl;
        if (i > 5) {
            break;
        }
    }
    std::cout << "Loop end";
}
```

break հրամանից  
հետո կատարվում է  
այս հրամանը

0  
1  
2  
3  
4  
5  
Loop end

<https://repl.it/@SevakRAU/ForBrEx1>



# continue հրաման

- **continue** հրամանը թույլ է տալիս բաց թողել ցիկլի ընթացիկ իտերացիան

```
#include <iostream>
int main() {
    for (int i = 0; i < 10; i++) {
        if (i == 5) {
            continue;
        }
        std::cout << i << std::endl;
    }
}
```



0  
1  
2  
3  
4  
6  
7  
8  
9

5 չի տպվել

<https://repl.it/@SevakRAU/ForContEx1>



# Ներդրված ցիկլեր

Ցիկլի մեջ կարելի է այլ ցիկլեր գրել (ներդնել):

```
#include <iostream>
int main() {
    for (int i = 0; i < 5; i++) {
        int j = 0;
        while (j < 3) {
            std::cout << "(i, j) - " << i << " " << j << std::endl;
            j++;
        }
    }
}
```



```
(i, j) - 0 0
(i, j) - 0 1
(i, j) - 0 2
(i, j) - 1 0
(i, j) - 1 1
(i, j) - 1 2
(i, j) - 2 0
(i, j) - 2 1
(i, j) - 2 2
(i, j) - 3 0
(i, j) - 3 1
(i, j) - 3 2
(i, j) - 4 0
(i, j) - 4 1
(i, j) - 4 2
```

<https://repl.it/@SevakRAU/NestedForEx1>



# Ներդրված ցիկլեր, օրինակ

0-4 թվերից ամեն մեկը տպել 4 անգամ

```
#include <iostream>
int main() {
    for (int i = 0; i < 5; i++) {
        int j = 0;
        while (j < 4) {
            std::cout << i << " ";
            j++;
        }
        std::cout << "\n";
    }
}
```



```
0 0 0 0
1 1 1 1
2 2 2 2
3 3 3 3
4 4 4 4
```

<https://repl.it/@SevakRAU/NestedLoopEx2>





# Խնդիրներ

1. Մուտքագրել 10 թիվ, ամեն մուտքագրած թվի համար տպել այդ թվի թվանշանների սիմետրիկ տեղափոխությամբ ստացված թիվը
2. Տպել 10x10 վրա շախմատի տախտակ (սև գույն 0, սպիտակ գույն 1)
3. Մուտքագրել 10 թիվ, դուրս բերել ամենամեծ/ամենափոքր թվանշանների գումար/արտադրյալ ունեցող թիվը:
4. Մուտքագրված դրական տասական թվի համար  $[1, 100]$  միջակայքից տպել նրա երկուական կոդը (ուղիղ կոդ)
5. Ուղիղ կոդով մուտքագրված երկուական կոդի համար տպել համապատասխան տասական թիվը



# Տնային աշխատանք

← → ↻ earth.ispras.ru/rau\_gov/ ☆

## Վարժություններ

- 00 Նախավարժանք
- 01 Թվաբանություն և ճյուղավորում
- 02 Ցիկլեր և ստատիկ զանգվածներ
- 03 Դինամիկ զանգվածներ և ֆունկցիաներ
- 04 Երկչափ զանգվածներ և դասեր

[https://earth.ispras.ru/rau\\_gov/](https://earth.ispras.ru/rau_gov/)



# Տնային աշխատանք

4,7,8,9 խնդիրները տնային աշխատանք



# Շնորհակալություն. Հարցե՞ր

