

# «ԾՐԱԳՐԱՎՈՐՄԱՆ ՀԻՄՈՒՆՔՆԵՐ» դասընթաց

այլ ոլորտներից դեպի տեխնոլոգիական ոլորտ  
սկսնականների համար



edu2020.am

## ԴԱՍ #4



ՀԱՅ-ՌՈՒՄԱԿԱՆ  
ՀԱՄԱԼՍԱՐԱՆ



ՀԱՅԱՍՏԱՆԻ ՀԱՆՐԱՊԵՏՈՒԹՅԱՆ  
ԲԱՐՁՐ ՏԵԽՆՈԼՈԳԻԱԿԱՆ  
ԱՐԴՅՈՒՆԱԲԵՐՈՒԹՅԱՆ ՆԱԽԱՐԱՐՈՒԹՅՈՒՆ

# bool տիպ

*bool* տիպը ընդունում է երկու հնարավոր արժեքներից մեկը՝ *true*, *false* (ճիշտ, սխալ)

```
#include <iostream>
```

```
int main() {  
    bool value1 = true;  
    bool value2 = false;  
}
```

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



# bool տիպ

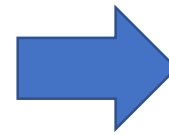
1. *false* – 0

2. *true* – 0-ից տարբեր ցանկացած արժեք

```
#include <iostream>
int main() {
    bool value1 = true;
    bool value2 = false;
    std::cout << "value1 " << value1 << std::endl;
    std::cout << "value2 " << value2 << std::endl;

    bool value3 = 100;
    std::cout << "value3 " << value3 << std::endl;

    bool value4 = 0;
    std::cout << "value4 " << value4 << std::endl;
}
```



value1 1  
value2 0  
value3 1  
value4 0

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





# Համեմատման օպերատորներ

1.  $A > B$ , արժեքը *true* է եթե  $A$ -ն մեծ է  $B$ -ից, հակառակ դեպքում *false*
2.  $A >= B$ , արժեքը *true* է եթե  $A$ -ն մեծ կամ հավասար է  $B$ -ից , հակառակ դեպքում *false*
3.  $A < B$ , արժեքը *true* է եթե  $B$ -ն մեծ է  $A$ -ից, հակառակ դեպքում *false*
4.  $A <= B$ , արժեքը *true* է եթե  $B$ -ն մեծ կամ հավասար է  $A$ -ից, հակառակ դեպքում *false*
5.  $A == B$ , արժեքը *true* է եթե  $A$ -ն հավասար է  $B$ -ին, , հակառակ դեպքում *false*
6.  $A != B$ , արժեքը *true* է եթե  $A$ -ն հավասար չէ  $B$ -ին, , հակառակ դեպքում *false*



# Համեմատման օպերատորներ, օրինակ

```
#include <iostream>
int main() {
    int A = 1, B = 2, C = 1;
    std::cout << "A > B is " << (A > B) << std::endl;
    std::cout << "A >= B is " << (A >= B) << std::endl;
    std::cout << "A > C is " << (A > C) << std::endl;
    std::cout << "A >= C is " << (A >= C) << std::endl;
    std::cout << "A < B is " << (A < B) << std::endl;
    std::cout << "A == B is " << (A == B) << std::endl;
    std::cout << "A == C is " << (A == C) << std::endl;
    std::cout << "A != C is " << (A != C) << std::endl;
    std::cout << "A != B is " << (A != B) << std::endl;
}
```



A > B is 0  
A >= B is 0  
A > C is 0  
A >= C is 1  
A < B is 1  
A == B is 0  
A == C is 1  
A != C is 0  
A != B is 1

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



# Ճյուղավորման օպերատորներ, if

*statement* կկատարվի եթե *expression*-ը *true* է

```
if (expression) {  
    statement;  
}
```

```
#include <iostream>  
  
int main() {  
    int x;  
    std::cin >> x;  
    if (x > 0) {  
        std::cout << "x is positive" << std::endl;  
    }  
}
```

<https://repl.it/@SevakRAU/RAUIfEx1>



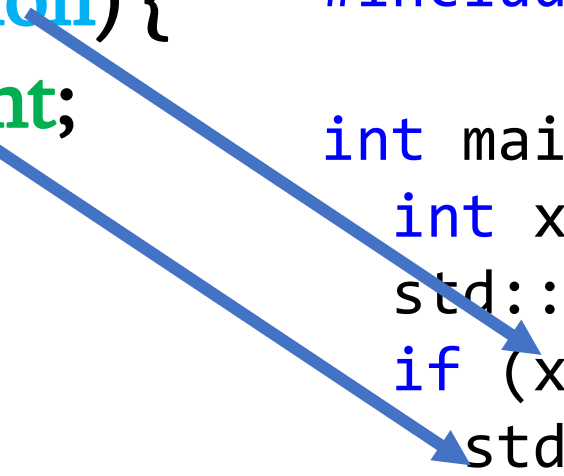
# Ճյուղավորման օպերատորներ, if

*statement* կկատարվի եթե *expression*-ը *true* է

```
if (expression) {  
    statement;  
}
```

```
#include <iostream>  
  
int main() {  
    int x;  
    std::cin >> x;  
    if (x > 0) {  
        std::cout << "x is positive" << std::endl;  
    }  
}
```

<https://repl.it/@SevakRAU/RAUIfEx1>



# Ճյուղավորման օպերատորներ, if

*statement1... statementN* կկատարվեն եթե *expression*-ը *true* է

```
if (expression) {  
    statement1;  
    .....  
    statementN;  
}  
  
#include <iostream>  
  
int main() {  
    int x;  
    std::cin >> x;  
    if (x == 0) {  
        std::cout << "x is zero, lets add 10 to it" << std::endl;  
        x += 10;  
    }  
    std::cout << "x is " << x << std::endl;  
}
```

<https://repl.it/@SevakRAU/RAUEx2>

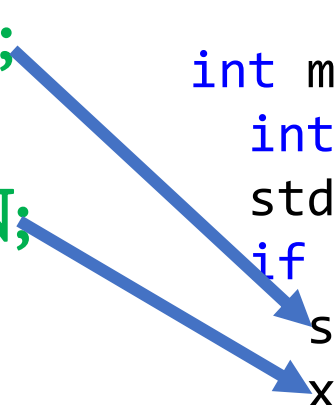




# Ճյուղավորման օպերատորներ, if

*statement1... statementN* կկատարվեն եթե *expression*-ը *true* է

```
if (expression) {  
    statement1;  
    .....  
    statementN;  
}  
  
#include <iostream>  
  
int main() {  
    int x;  
    std::cin >> x;  
    if (x == 0) {  
        std::cout << "x is zero, lets add 10 to it" << std::endl;  
        x += 10;  
    }  
    std::cout << "x is " << x << std::endl;  
}
```

A diagram with two blue arrows. The first arrow starts from the 'statement1;' line in the C++ code block and points to the 'std::cout << "x is zero, lets add 10 to it" << std::endl;' line in the C++ code block. The second arrow starts from the 'statementN;' line in the C++ code block and points to the 'x += 10;' line in the C++ code block.

<https://repl.it/@SevakRAU/RAUEx2>



# Ճյուղավորման օպերատորներ, else

*statement1... statementN* կկատարվեն եթե *expression*-ը *true* է, հակառակ դեպքում կկատարվեն *e-statement1... e-statementM*

```
if (expression) {  
    statement1;  
    .....  
    statementN;  
} else {  
    e-statement1;  
    .....  
    e-statementM;  
}
```

```
#include <iostream>  
  
int main() {  
    int x;  
    std::cin >> x;  
    if (x > 0) {  
        std::cout << "x is positive" << std::endl;  
    } else {  
        std::cout << "x is not positive" << std::endl;  
    }  
}
```

<https://repl.it/@SevakRAU/RAUIfElseEx1>



# Ճյուղավորման օպերատորներ, else if

```
#include <iostream>
```

```
int main() {  
    int x;  
    std::cin >> x;  
    if (x > 0) {  
        std::cout << "x is positive" << std::endl;  
    } else if (x == 0) {  
        std::cout << "x is zero" << std::endl;  
    } else {  
        std::cout << "x is negative" << std::endl;  
    }  
}
```

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



# Ճյուղավորման օպերատորներ, else if

```
#include <iostream>
```

```
int main() {  
    int x;  
    std::cin >> x;  
    if (x > 0) {  
        std::cout << "x is positive" << std::endl;  
    } else if (x == 0) {  
        std::cout << "x is zero" << std::endl;  
    } else {  
        std::cout << "x is negative" << std::endl;  
    }  
}
```

else վերաբերվում է  
ամենամոտ if-ին

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



# Ճյուղավորման օպերատորներ, else if

```
#include <iostream>
```

```
int main() {  
    int x;  
    std::cin >> x;  
    if (x > 0) {  
        std::cout << "x is positive" << std::endl;  
    } else if (x == 0) {  
        std::cout << "x is zero" << std::endl;  
    } else {  
        std::cout << "x is negative" << std::endl;  
    }  
}
```

*else* վերաբերվում է  
ամենամոտ *if*-ին

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





# Ներդրված ճյուղավորման օպերատորներ

```
#include <iostream>
```

```
int main() {  
    int x;  
    std::cin >> x;  
    if (x > 0) {  
        if (x % 2 == 0) {  
            std::cout << "x is positive even number" << std::endl;  
        } else {  
            std::cout << "x is positive odd number" << std::endl;  
        }  
    } else {  
        std::cout << "x is not positive" << std::endl;  
    }  
}
```

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



# Ներդրված ճյուղավորման օպերատորներ

```
#include <iostream>
```

```
int main() {  
    int x;  
    std::cin >> x;  
    if (x > 0) {  
        if (x % 2 == 0) {  
            std::cout << "x is positive even number" << std::endl;  
        } else {  
            std::cout << "x is positive odd number" << std::endl;  
        }  
    } else {  
        std::cout << "x is not positive" << std::endl;  
    }  
}
```

*x մեծ է 0 դեպքն է*

*x փոքր հավասար  
է 0 դեպքն է*

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



# Խնդիրներ

1. Մուտքագրել 2 թիվ և տպել դրանցից մեծագույնը/փոքրագույնը
2. Մուտքագրել թիվ, եթե այն զույգ է տպել այդ թիվը բաժանած 2-ի, հակառակ դեպքում բազմապատկած 3-ով
3. Մուտքագրել թիվ, եթե այն զույգ է մուտքագրել ևս մեկ թիվ, եթե նոր մուտքագրած թիվն էլ է զույգ տպել այդ երկու թվեր արտադրյալը: Մնացած բոլոր դեպքերում ոչինչ չանել:



# Տրամաբանական օպերատորներ (և, and)

&& OPERATOR (and)		
a	b	a && b
true	true	true
true	false	false
false	true	false
false	false	false

```
#include <iostream>
```

```
int main() {  
    int x;  
    std::cin >> x;  
    if (x >= 10 && x <= 20) {  
        std::cout << "x is in interval [10, 20]" << std::endl;  
    }  
}
```

<https://repl.it/@SevakRAU/ANDEx1>



# Տրամաբանական օպերատորներ (և, and)

&& OPERATOR (and)		
a	b	a && b
true	true	true
true	false	false
false	true	false
false	false	false

```
#include <iostream>
```

```
int main() {  
    int x;  
    std::cin >> x;  
    if (x >= 10 && x <= 20) {  
        std::cout << "x is in interval [10, 20]" << std::endl;  
    }  
}
```

<https://repl.it/@RAU/ANDEx1>

Եթե **false** է հաջորդ պայմանը չի  
ստուգվում (*lazy evaluation*)





# Տրամաբանական օպերատորներ (կամ, or)

OPERATOR (or)		
a	b	a    b
true	true	true
true	false	true
false	true	true
false	false	false

```
#include <iostream>
```

```
int main() {  
    int x;  
    std::cin >> x;  
    if (x < 5 || x > 10) {  
        std::cout << "x is not in interval [5, 10]" << std::endl;  
    }  
}
```

<https://repl.it/@SevakRAU/OREx1>



# Տրամաբանական օպերատորներ (կամ, or)

OPERATOR (or)		
a	b	a    b
true	true	true
true	false	true
false	true	true
false	false	false

```
#include <iostream>
```

```
int main() {  
    int x;  
    std::cin >> x;  
    if (x < 5 || x > 10) {  
        std::cout << "x is not in interval [5, 10]" << std::endl;  
    }  
}
```

<https://repl.it/@akRAU/OREx1>

Եթե **true** է հաջորդ պայմանը չի  
ստուգվում (**lazy evaluation**)



# Տրամաբանական օպերատորներ (Ժխտում)

Ժխտում - !

! true -> false

! false -> true

```
#include <iostream>
```

```
int main() {  
    int x;  
    std::cin >> x;  
    if (!(x >= 0)) {  
        std::cout << "x is negative!" << std::endl;  
    }  
}
```

<https://repl.it/@SevakRAU/NOTEx1>



# Տրամաբանական օպերատորներ (Ժխտում)

Ժխտում - !

! true -> false

! false -> true

```
#include <iostream>
```

```
int main() {  
    int x;  
    std::cin >> x;  
    if (!(x >= 0)) {  
        std::cout << "x is negative!" << std::endl;  
    }  
}
```

<https://repl.it/@vakRAU/NOTEx1>

Համարժեք է  $x < 0$



# Տրամաբանական օպերատորներ (Ժխտում)

Ժխտում - !

int x;

! x -> 0, եթե x != 0

! x -> 1, եթե x == 0

```
#include <iostream>
```

```
int main() {
```

```
    int x = 0, y = 10;
```

```
    std::cout << "!x is " << !x << std::endl;
```

```
    std::cout << "!y is " << !y << std::endl;
```

```
}
```



!x is 1

!y is 0

<https://repl.it/@SevakRAU/NOTEx2>





# Խնդիրներ

1. Մուտքագրել թիվ և ստուգել, որ այն  $[1, 10]$  ամբողջ թվերի միջակայքից է և զույգ/կենս:
2. Մուտքագրել թիվ և ստուգել, որ այն **չ** պատկանում  $[1, 10]$  ամբողջ թվերի միջակայքին և զույգ/կենս է:



# switch օպերատոր

Կատարվում են *statement1,....,statementN*, *statement* եթե *expression*-ը համընկնում է *const-expressionI*-ի հետ

```
switch (expression) {  
    case const-expression1: statement1;  
    .....  
    case const-expressionI: statementI;  
    .....  
    case const-expressionN: statementN;  
    default: statement;  
}
```

```
#include <iostream>  
int main() {  
    int x;  
    std::cin >> x;  
    switch (x) {  
        case 1:  
            std::cout << "Case 1" << std::endl;  
        case 2:  
            std::cout << "Case 2" << std::endl;  
        case 3:  
            std::cout << "Case 3" << std::endl;  
        default:  
            std::cout << "Default" << std::endl;  
    }  
}
```

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



# switch օպերատոր

Կատարվում են *statement1, ..., statementN*, *statement* եթե *expression*-ը համընկնում է *const-expressionI*-ի հետ

```
switch (expression) {  
    case const-expression1: statement1;  
    .....  
    case const-expressionI: statementI;  
    .....  
    case const-expressionN: statementN;  
    default: statement;  
}
```

```
#include <iostream>  
int main() {  
    int x;  
    std::cin >> x;  
    switch (x) {  
        case 1:  
            std::cout << "Case 1" << std::endl;  
        case 2:  
            std::cout << "Case 2" << std::endl;  
        case 3:  
            std::cout << "Case 3" << std::endl;  
        default:  
            std::cout << "Default" << std::endl;  
    }  
}
```

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



# switch օպերատոր

Եթե *expression*-ը չի համընկնում ոչ մի *statmentI*-ի հետ ապա կատարվում է միայն *statment*-ը

```
switch (expression) {  
    case const-expression1: statmentI;  
    .....  
    case const-expressionI: statmentI;  
    .....  
    case const-expressionN: statmentN;  
    default: statment;  
}
```

Կատարվում է  
միայն սա

```
#include <iostream>  
int main() {  
    int x;  
    std::cin >> x;  
    switch (x) {  
        case 1:  
            std::cout << "Case 1" << std::endl;  
        case 2:  
            std::cout << "Case 2" << std::endl;  
        case 3:  
            std::cout << "Case 3" << std::endl;  
        default:  
            std::cout << "Default" << std::endl;  
    }  
}
```

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



# switch օպերատոր

```
switch (expression) {  
    case const-expression1: statement1;  
    .....  
    case const-expressionI: statementI;  
    .....  
    case const-expressionN: statementN;  
    default: statement;  
}
```

Կարող է  
բացակայել

```
#include <iostream>  
  
int main() {  
    int x;  
    std::cin >> x;  
    switch (x) {  
        case 1:  
            std::cout << "Case 1" << std::endl;  
        case 2:  
            std::cout << "Case 2" << std::endl;  
        case 3:  
            std::cout << "Case 3" << std::endl;  
    }  
}
```

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





# switch օպերատոր, break

```
#include <iostream>
int main() {
    int x;
    std::cin >> x;
    switch (x) {
        case 1:
            std::cout << "Case 1" << std::endl;
            break;
        case 2:
            std::cout << "Case 2" << std::endl;
            break;
        case 3:
            std::cout << "Case 3" << std::endl;
            break;
    }
}
```

Կլատարի միայն այն *case*  
որի հետ համընկել է և չի  
անցնի հաջորդներին

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

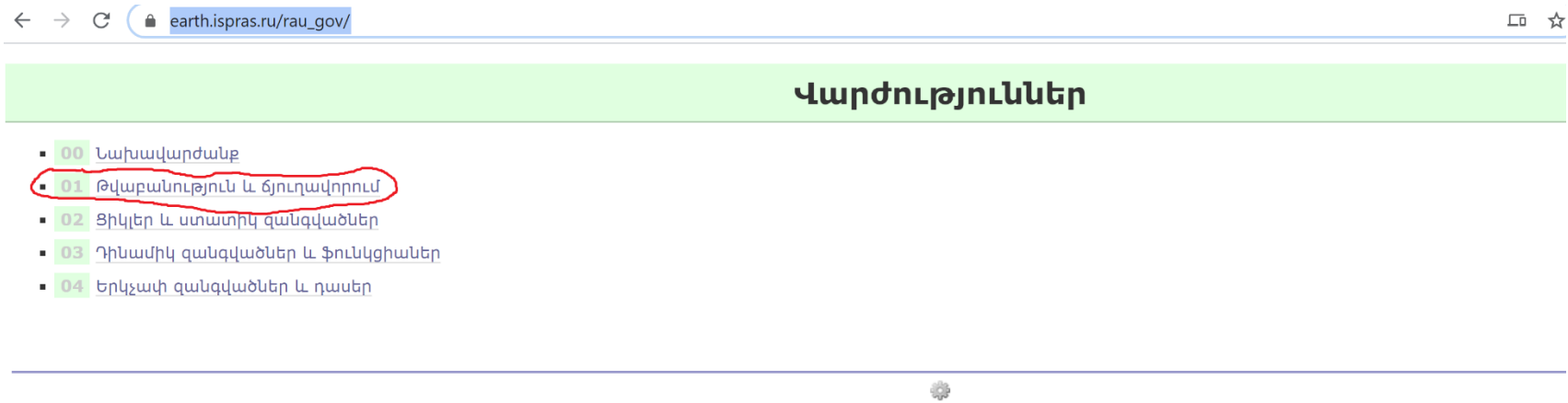


# Խնդիրներ

1. Մուտքագրել թիվ  $A = [1, 10]$  ամբողջ թվերի միջակայքից և դուրս բերել բոլոր թվերը որոնք մեծ հավասար են մուտքագրված թվից և պատկանում են  $A$  միջակայքին:
2. Մուտքագրել ամիսը թվով  $(1, \dots, 12)$  և դուրս բերել համապատասխան ամսվա անվանումը (հունվար, ....., դեկտեմբեր):



# Տնային աշխատանք




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



# Տնային աշխատանք

## 11-20 Խնդիրները տնային աշխատանք



Севак Саргсян [Մաս 1: Թվաբանություն և ճյուղավորում]: Сдать решение

НастройкиИнфоИтогПосылкиПоложение участниковОтправить вопросСообщенияВыйти из системы [sevak]

10:37:41 / RUNNING

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----

### Сдать решение задачи 11-Հավասարություն

Ограничение времени: 1 с  
Ограничение памяти: 64M  
Оставшиеся послылки: 32

### Задача 11: Հավասարություն

Գրել ծրագիր, որը մուտքում կստանա 2 դրական ամբողջ թիվ և կարտածի YES, եթե այդ թվերը հավասար են և NO հակառակ դեպքում: (դաս 4)

### Примеры

Входные данные



# Շնորհակալություն. Հարցե՞ր

