

# «ԾՐԱԳՐԱՎՈՐՄԱՆ ՀԻՄՈՒՆՔՆԵՐ» դասընթաց

այլ ոլորտներից դեպի տեխնոլոգիական ոլորտ  
սկսնակների համար



edu2020.am

## ԴԱՍ #2



ՀԱՅ-ՌՈՒՄԱԿԱՆ  
ՀԱՄԱԼՍԱՐԱՆ



ՀԱՅԱՍՏԱՆԻ ՀԱՆՐԱՊԵՏՈՒԹՅԱՆ  
ԲԱՐՁՐ ՏԵԽՆՈԼՈԳԻԱԿԱՆ  
ԱՐԴՅՈՒՆԱԲԵՐՈՒԹՅԱՆ ՆԱԽԱՐԱՐՈՒԹՅՈՒՆ

# Ամբողջ տիպ

C++ում տարբեր մեծության ամբողջ թվեր կարելի է ներկայացնել

1. **short** համեմատաբար փոքր թվերի համար
2. **int** միջին մեծության թվերի համար
3. **long** և **long long** շատ մեծ թվերի համար

Համակարգչից ճարտարապետությունից (Intel, IBM, ARM, ...) կախված ներկայացվող թվերի մեծությունները կարող են տատանվել: Օրինակ՝ *int* տիպը կարող է որոշ դեպքերում պահել թվեր *[-32768, 32767]* միջակայքից, իսկ հաճախ պահում է *[-2147483648, 2147483647]* միջակայքից:

*Ավելի մանրամասն բոլոր դեպքերը կքննարկենք հաջորդ դասին:*



# Ամբողջ տիպեր, օրինակ

```
#include <iostream>
```

```
int main() {  
    short x = 1;  
    int y = 2;  
    long z = 4;  
    long long k = 3;  
    std::cout << "x = " << x << ", y = " << y << ", z = " << z << ", k = " << k;  
}
```



**x = 1, y = 2, z = 4, k = 3**

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





# Ամբողջ տիպերի չափի ստուգում

Բերված օրինակը տպում է մեզ հետաքրքիր տիպերի չափերը՝

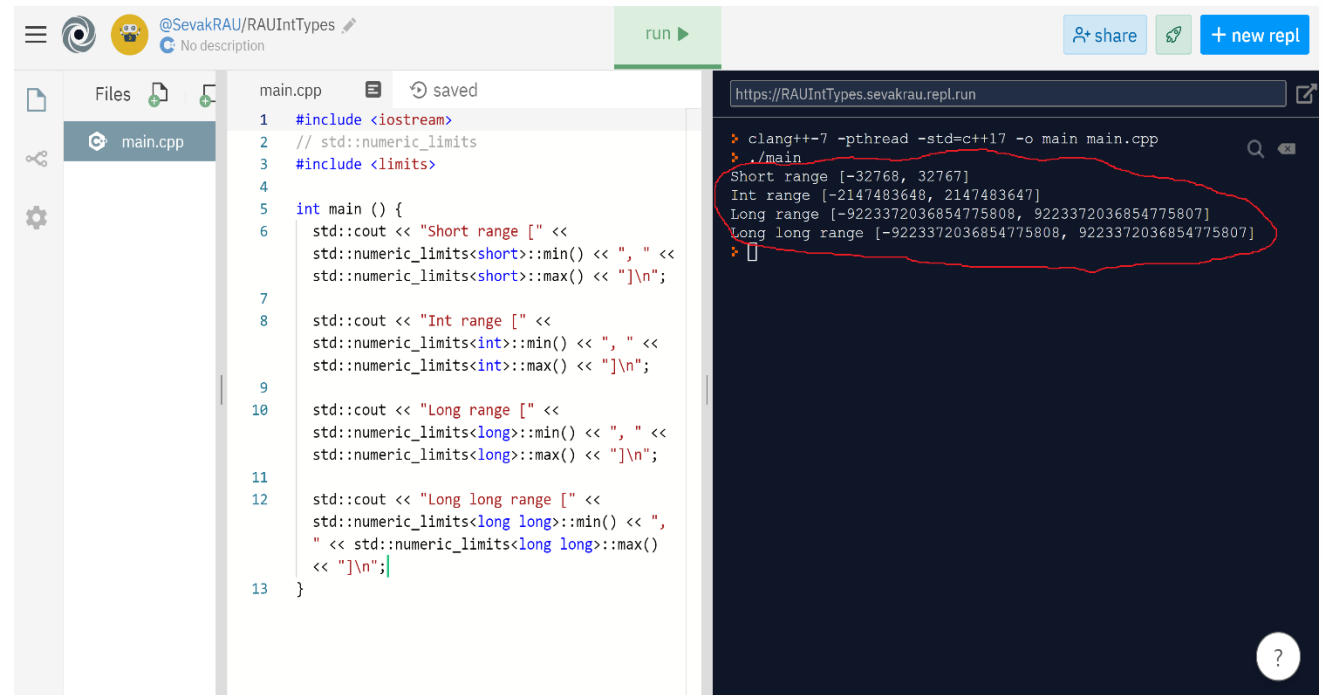
```
#include <iostream>
// std::numeric_limits
#include <limits>

int main () {
    std::cout << "Short range [" <<
        std::numeric_limits<short>::min() << ", " <<
        std::numeric_limits<short>::max() << "]\n";

    std::cout << "Int range [" <<
        std::numeric_limits<int>::min() << ", " <<
        std::numeric_limits<int>::max() << "]\n";

    std::cout << "Long range [" <<
        std::numeric_limits<long>::min() << ", " <<
        std::numeric_limits<long>::max() << "]\n";

    std::cout << "Long long range [" <<
        std::numeric_limits<long long>::min() << ", " <<
        std::numeric_limits<long long>::max() << "]\n";
}
```



```
main.cpp
1 #include <iostream>
2 // std::numeric_limits
3 #include <limits>
4
5 int main () {
6     std::cout << "Short range [" <<
7         std::numeric_limits<short>::min() << ", " <<
8         std::numeric_limits<short>::max() << "]\n";
9
10    std::cout << "Int range [" <<
11        std::numeric_limits<int>::min() << ", " <<
12        std::numeric_limits<int>::max() << "]\n";
13
14    std::cout << "Long range [" <<
15        std::numeric_limits<long>::min() << ", " <<
16        std::numeric_limits<long>::max() << "]\n";
17
18    std::cout << "Long long range [" <<
19        std::numeric_limits<long long>::min() << ", " <<
20        std::numeric_limits<long long>::max() << "]\n";
21
22 }
```

```
https://RAUIntTypes.sevakrau.repl.run
> clang++-7 -pthread -std=c++17 -o main main.cpp
> ./main
Short range [-32768, 32767]
Int range [-2147483648, 2147483647]
Long range [-9223372036854775808, 9223372036854775807]
Long long range [-9223372036854775808, 9223372036854775807]
```

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



# Հայտարարման համարժեքություն

```
#include <iostream>
```

```
int main() {
```

```
    short a = 1;
```

```
    short int b = 1;
```

```
    int c = 2;
```

```
    long d = 3;
```

```
    long int e = 3;
```

```
    long long f = 4;
```

```
    long long int g = 4;
```

```
}
```

Համարժեք գրելաձև

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



# Հայտարարման համարժեքություն

```
#include <iostream>
```

```
int main() {
```

```
    short a = 1;
```

```
    short int b = 1;
```

```
    int c = 2;
```

```
    long d = 3;
```

```
    long int e = 3;
```

```
    long long f = 4;
```

```
    long long int g = 4;
```

```
}
```

Համարժեք գրելաձև

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



# Հայտարարման համարժեքություն

```
#include <iostream>
```

```
int main() {
```

```
    short a = 1;
```

```
    short int b = 1;
```

```
    int c = 2;
```

```
    long d = 3;
```

```
    long int e = 3;
```

```
    long long f = 4;
```

```
    long long int g = 4;
```

Համարժեք գրելաձև

```
}
```

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



# Գործողություններ ամբողջ տիպերի հետ

Ամբողջ տիպերի համար հասանելի են հետևյալ գործողությունները՝

- + երկու թվերի գումար
- երկու թվերի տարբերություն
- \* երկու թվերի արտադրյալ
- / երկու թվերի բաժանում ամբողջ մասով
- % երկու թվերի բաժանում մնացորդով





# Գործողությունների ամբողջ տիպերի հետ

```
#include <iostream>
```

```
int main() {  
    int a = 10, b = 20, d = 15;  
    int sum = a + b;  
    int diff = a - b;  
    int mul = a * d;  
    int div = b / a;  
    int div_rem = d % a;  
    std::cout << "Sum = " << sum << "\n";  
    std::cout << "Diff = " << diff << "\n";  
    std::cout << "Mul = " << mul << "\n";  
    std::cout << "Div = " << div << "\n";  
    std::cout << "Div. rem. = " << div_rem << "\n";  
}
```



*Sum - 30*  
*Diff - -10*  
*Mul - 150*  
*Div - 2*  
*Div. rem. - 5*

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



# Գործողությունների ամբողջ տիպերի հետ

```
#include <iostream>
```

```
int main() {  
    int a = 10, b = 20, d = 15;  
    int sum = a + b;  
    int diff = a - b;  
    int mul = a * d;  
    int div = b / a;  
    int div_rem = d % a;  
    std::cout << "Sum = " << sum << "\n";  
    std::cout << "Diff = " << diff << "\n";  
    std::cout << "Mul = " << mul << "\n";  
    std::cout << "Div = " << div << "\n";  
    std::cout << "Div. rem. = " << div_rem << "\n";  
}
```

Պետք է հաշվի առնել, որ  
գործողության արդյունքը կարող է  
լինել ավելի մեծ քան օգտագործված  
տիպը կարող է ներկայացնել



*Sum - 30  
Diff - -10  
Mul - 150  
Div - 2  
Div. rem. - 5*

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



# Գործողությունների ամբողջ տիպերի հետ

```
#include <iostream>
```

Օգտագործեք ավելի մեծ  
չափի տիպ

```
int main() {  
    long a = 1000000, b = 2000000, d = 1500000000000;  
    long sum = a + b;  
    long diff = a - b;  
    long mul = a * d;  
    int div = b / a;  
    int div_rem = d % a;  
    std::cout << "Sum - " << sum << std::endl;  
    std::cout << "Diff - " << diff << std::endl;  
    std::cout << "Mul - " << mul << std::endl;  
    std::cout << "Div - " << div << std::endl;  
    std::cout << "Div. rem. - " << div_rem << std::endl;  
}
```

*Sum - 3000000*

*Diff - -1000000*

*Mul - 1500000000000000000*

*Div - 2*

*Div. rem. - 0*

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



# Զրոյի վրա բաժանում ամբողջ թվերի դեպքում

```
#include <iostream>
```

```
int main() {  
    int a = 1, b = 0;  
    a = a / b;  
}
```

Զրոյի վրա բաժանել չի  
թույլատրվում, ծրագրի  
վարքը անորոշ է



# Մահող կետավ թվեր

C++-ում կան մի քանի տիպեր *սահող կետավ թվերը* (ռացիոնալ և իռացիոնալ թվերի մոտավոր ներկայացում) ներկայացնելու համար

1. **float** համեմատաբար փոքր ճշտությամբ ներկայացում
2. **double** և **long double** մեծ ճշտությամբ ներկայացում

```
#include <iostream>
int main() {
    float f = 0.1;
    double d = 0.2;
    long double e = 0.1210223;
    std::cout << "f = " << f << ", d = " << d << ", e = " << e;
}
```

f = 0.1, d = 0.2, e = 0.121022

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





# Մահող կետով թվեր

Տպել 20 նիշի ճշտությամբ

```
int main() {  
    double d = 0.1;  
    std::cout << d << std::endl;  
    std::cout.precision(20);  
    std::cout << d << std::endl;  
}
```

→ 0.1

→ 0.100000000000000000555

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



# Մահող կետով թվեր

Մոտարկումով ներկայացնելու  
պատճառով վերագրված  
արժեքը տարբերվում է  
իրականից

```
int main() {  
    double d = 0.1;  
    std::cout << d << std::endl;  
    std::cout.precision(20);  
    std::cout << d << std::endl;  
}
```



0.1



0.100000000000000000555

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



# Գործողություններ սահող կետով թվերի հետ

Սահող կետով տիպերի համար հասանելի են հետևյալ գործողությունները՝

- + երկու թվերի գումար
- երկու թվերի տարբերություն
- \* երկու թվերի արտադրյալ
- / երկու թվերի բաժանում

```
#include <iostream>
int main() {
    float a = 1.2, b = 0.3;
    float sum = a + b;
    float diff = a - b;
    float mul = a * b;
    float div = a / b;
    std::cout << "sum = " << sum << ", diff = " << diff << ", mul = " << mul << ", div = " << div;
}
```



sum = 1.5, diff = 0.9, mul = 0.36, div = 4

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



# Ճշտության կորուստ

```
#include <iostream>
int main() {
    float f = 3.999999;
    f = f * f;
    double d = 3.999999;
    d = d * d;
    long double ld = 3.999999;
    ld = ld * ld;
    std::cout.precision(20);
    std::cout << "f =" << f << ", d = " << d << ", ld = " << ld;
}
```

Որքան մեծ տիպ ենք  
օգտագործում, այնքան  
փոքր է կորուստի չափը

f=15.99999237060546875,  
d = 15.999992000000998971,  
ld = 15.999992000000998882

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



# Զրոյի վրա բաժանում սահող կետով թվերի դեպքում

```
#include <iostream>
```

```
int main() {  
    float a = -1, b = 1, c = 0;  
    a = a / c;  
    b = b / c;  
    std::cout << a << " " << b;  
}
```



$-\infty$   $\infty$   
***-inf inf***

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





# Տվյալների մուտքագրում

```
#include <iostream>
```

```
int main() {  
    int x;  
    std::cout << "Insert square edge ";  
    std::cin >> x;  
    std::cout << "\nSquare area is " << x * x << std::endl;  
}
```

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



# Տվյալների մուտքագրում

```
#include <iostream>
```

```
int main() {  
    int x;  
    std::cout << "Insert square edge ";  
    std::cin >> x;  
    std::cout << "\nSquare area is " << x * x << std::endl;  
}
```

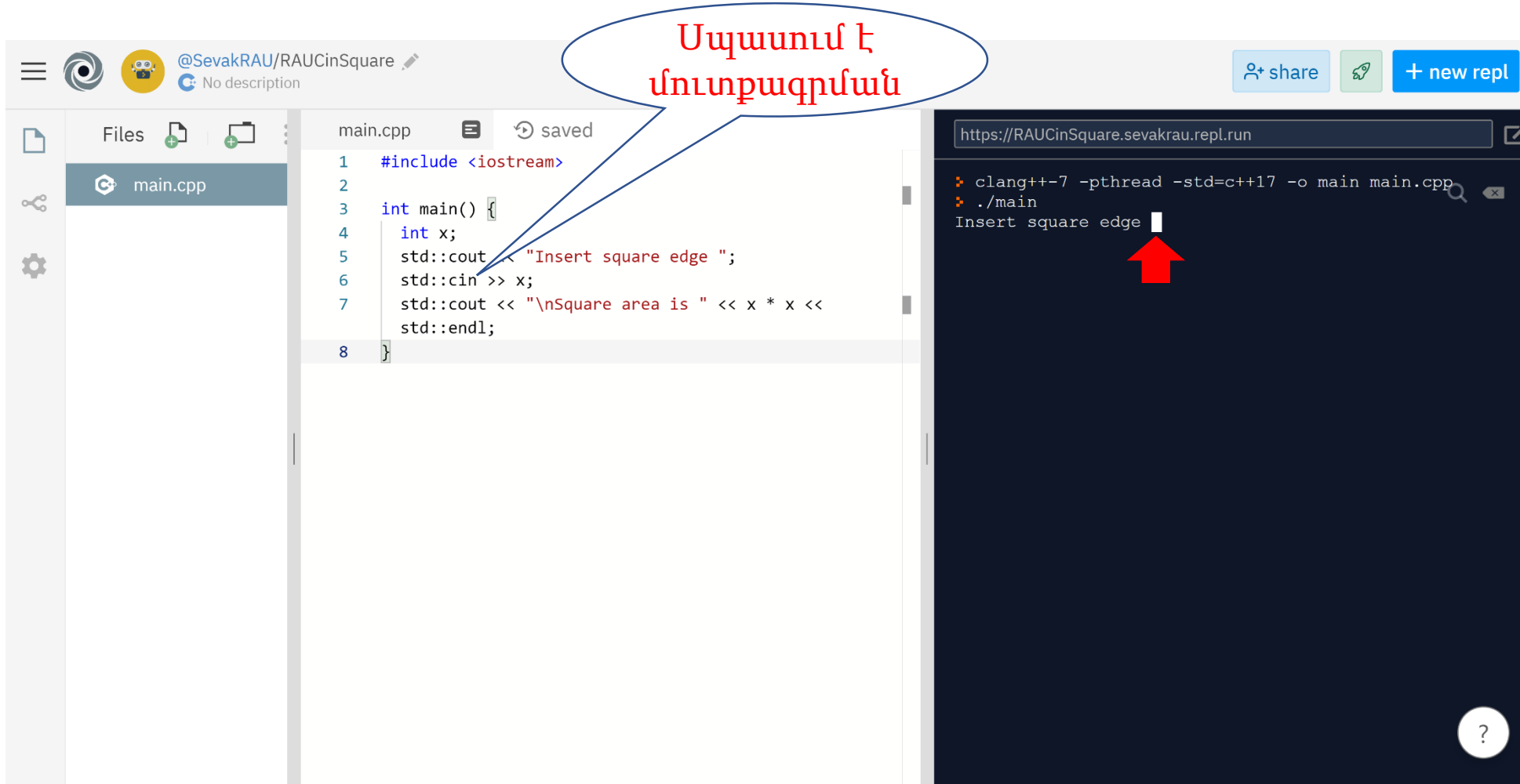
Մուտքագրման  
հրաման

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



# Տվյալների մուտքագրում

Մպատում է  
մուտքագրման



The screenshot displays a web-based C++ compiler interface. On the left, a file explorer shows 'main.cpp'. The code editor contains the following C++ code:

```
1 #include <iostream>
2
3 int main() {
4     int x;
5     std::cout << "Insert square edge ";
6     std::cin >> x;
7     std::cout << "\nSquare area is " << x * x <<
8     std::endl;
9 }
```

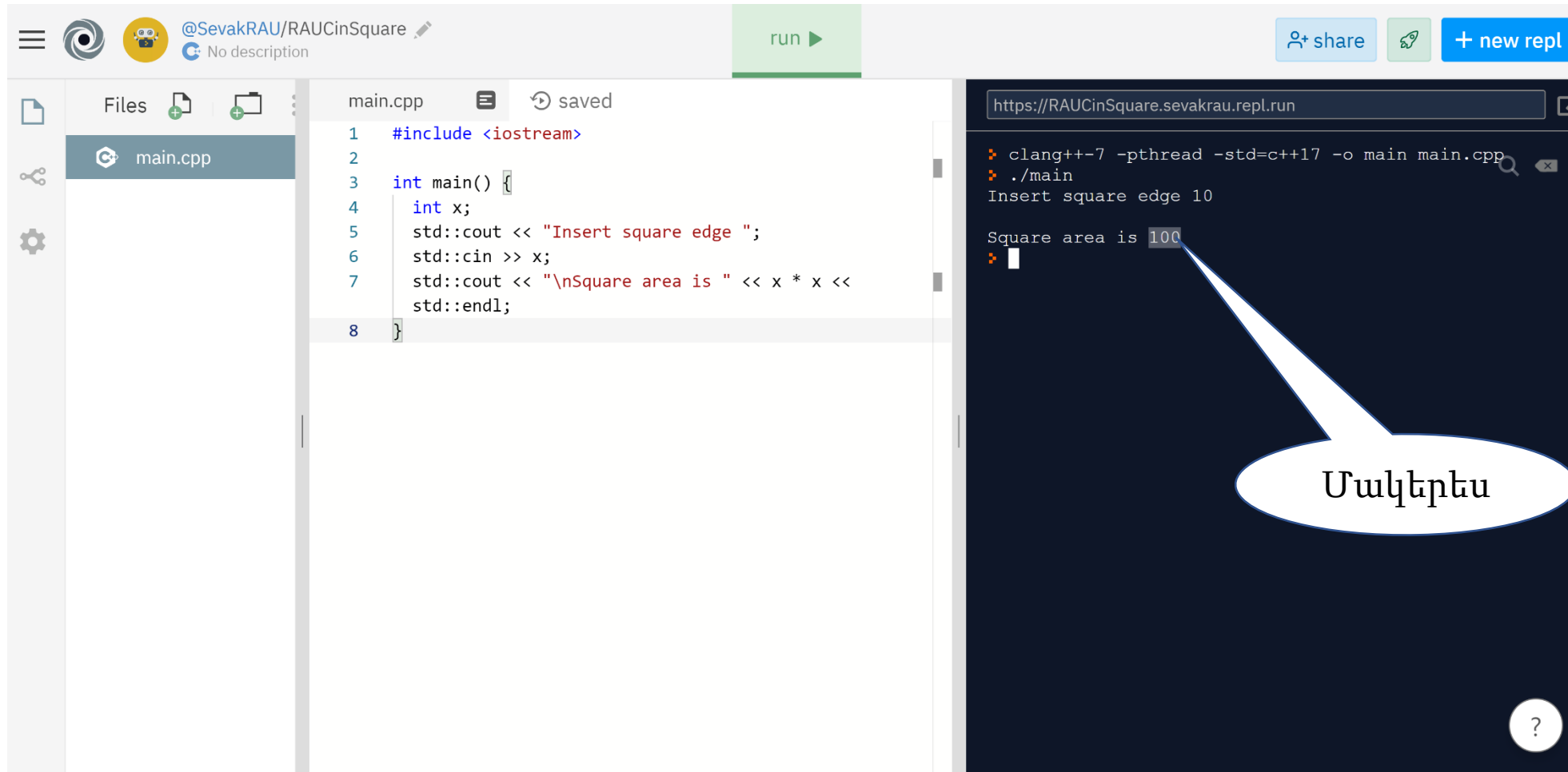
On the right, the terminal window shows the command prompt with the following commands and output:

```
clang++-7 -pthread -std=c++17 -o main main.cpp
./main
Insert square edge
```

A red arrow points to the input prompt 'Insert square edge' in the terminal. The URL bar shows 'https://RAUCinSquare.sevakrau.repl.run'.



# Տվյալների մուտքագրում



The screenshot displays a web-based C++ REPL interface. The top bar includes the user profile '@SevakRAU/RAUCinSquare', a 'run' button, and options to 'share' or 'new repl'. The left sidebar shows a file explorer with 'main.cpp' selected. The main editor area contains the following C++ code:

```
1 #include <iostream>
2
3 int main() {
4     int x;
5     std::cout << "Insert square edge ";
6     std::cin >> x;
7     std::cout << "\nSquare area is " << x * x <<
8     std::endl;
9 }
```

The right panel shows the terminal output for the command `clang++-7 -pthread -std=c++17 -o main main.cpp`. The program prompts 'Insert square edge 10' and outputs 'Square area is 100'. A white callout bubble with the text 'Մակերես' (Area) points to the value '100' in the output.



# Շրջանագծի մակերես

```
#include <iostream>

int main() {
    float Pi = 3.14;
    float radius;
    std::cout << "Insert cycle radius ";
    std::cin >> radius;
    float area = Pi * radius *radius;
    std::cout << "\nCycle are is " << area << std::endl;
}
```

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



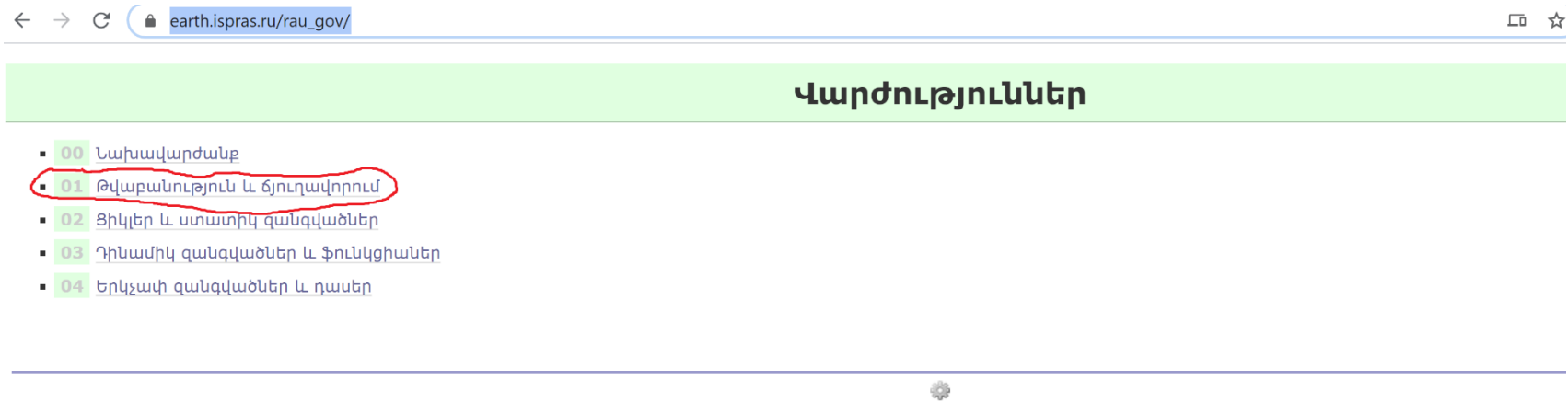


# Խնդիրներ

1. Հաշվել ուղղանկյան մակերեսը
2. Հաշվել քառակուսու պարագիծը
3. Հաշվել ուղղանկյան պարագիծը
4. Հաշվել եռանկյան մակերեսը, եթե տրված են հիմքը և բարձրությունը



# Տնային աշխատանք



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



# Տնային աշխատանք



Вход в систему [Մաս 1: Թվաբանություն և ճյուղավորում]

login:  пароль:  язык: Russian


[ejudge 3.7.9 #1 \(2020-01-24 17:49:41\)](#).

Copyright © 2000-2019 Alexander Chernov.



# Տնային աշխատանք

## 1-6 Խնդիրները տնային աշխատանք

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

НастройкиИнфоИтогПосылкиПоложение участниковОтправить вопросСообщенияВыйти из системы [sevaksargsyan]

10:42:42 / RUNNING / приём решений

1234567891011121314151617181920

**Сдать решение задачи 6-Սահող կետով թվերի բաժանում**

Ограничение времени: 1 с  
Ограничение памяти: 64М  
Оставшиеся послылки: 16

**Задача 6: Սահող կետով թվերի բաժանում**

Գրել ծրագիր, որը մուտքում կստանա 2 սահող կետով թիվ և կարտածի առաջինի և երկրորդի բաժանման արդյունքում ստացված քանորդը: (դաս 2)

**Примеры**

Входные данные

1 2

Результат работы

0.5

Входные данные

1 3



# Շնորհակալություն. Հարցեր?

