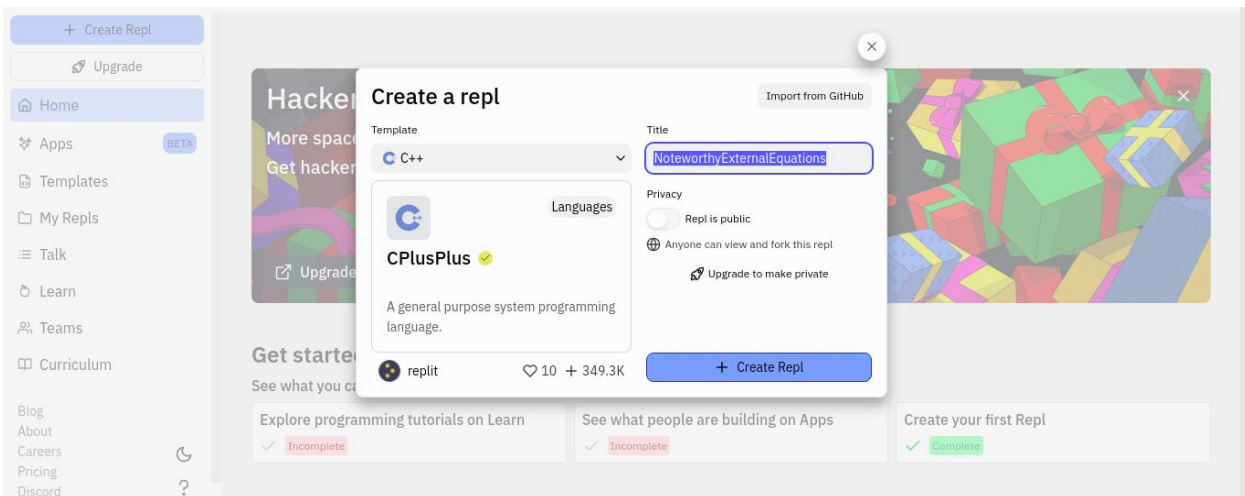
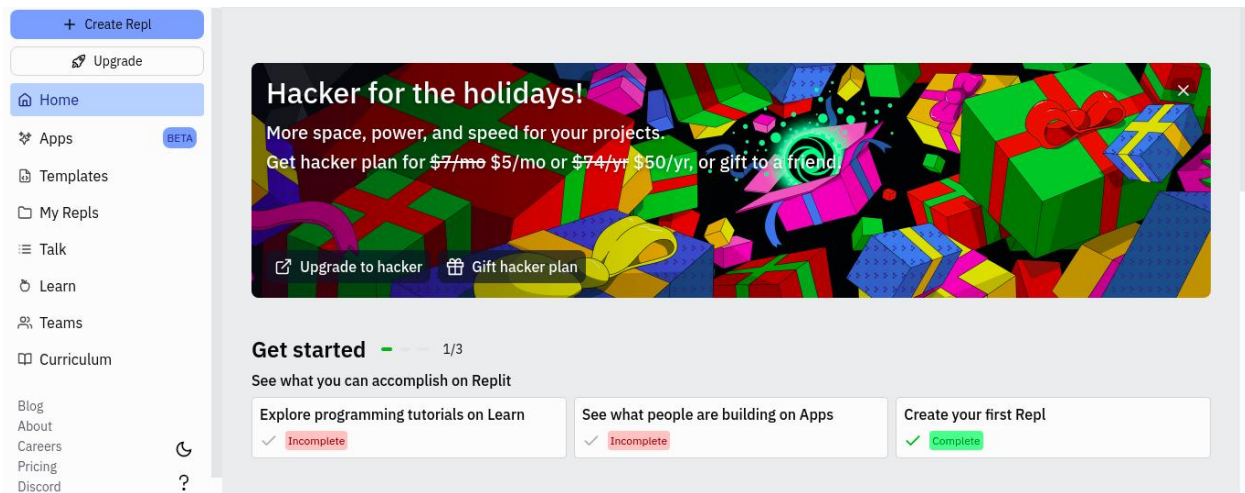
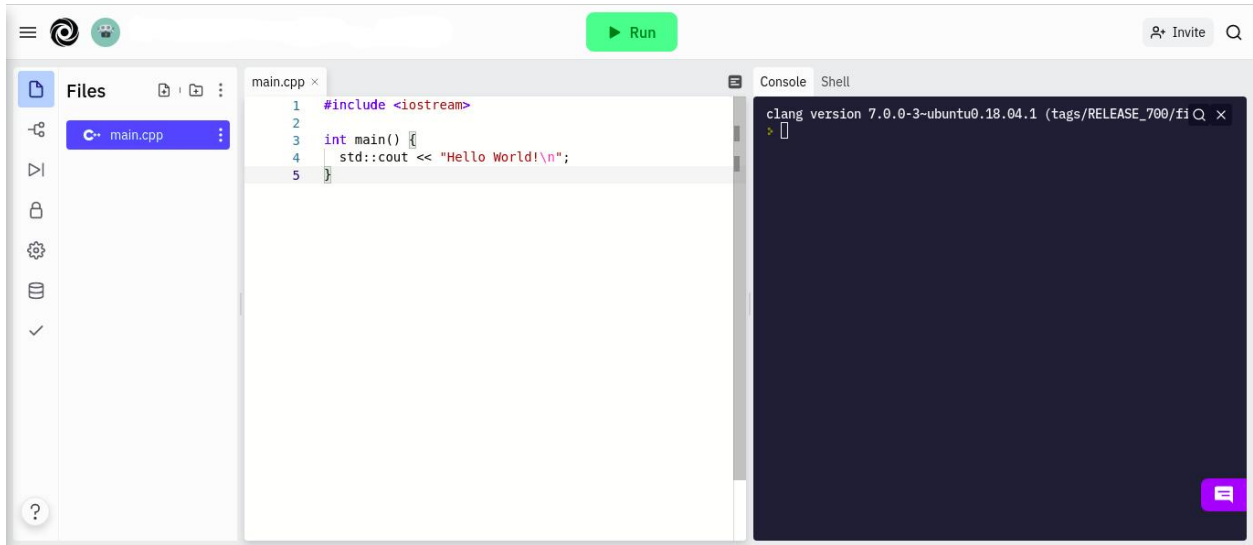


Ձանգվածների մշակում C++ լեզվով

Խնդիրներում գրված կոդը փորձարկելու համար պետք է այցելել <https://replit.com/> կայք, ստեղծել նոր **Repl**՝ ընտրելով **C++** լեզուն որպես հիմնական լեզու:





Երրորդ նկարի միջին սյունակում կոդի համար նախատեսված դաշտն է, իսկ երրորդ սյունակում **terminal**-ը: Վերևի **Run** կոճակը կմիացնի մեր կոդմից գրված ծրագիրը:

Արարատ Մատինյան

Խնդիր 1

```
#include <iostream>
#include <vector>
using namespace std;

int random(int max) {
    int min = -max;

    return min + rand() % (( max + 1 ) - min);
}

int main() {
    int n;

    vector<int> X;
    vector<int> Y;

    vector<int> X_vector_negatives;
    vector<int> X_vector_zeros;
    vector<int> X_vetor_positives;

    // Ներմուծում ենք ենք n-ը
    do {
        cout << "Type n: ";
        cin >> n;
    } while (n <= 0);

    // X վեկտորը լցնում ենք թվերով
    for (int i = 0; i < n; i++) X.push_back(random(n));

    // Ստուգում ենք X-ի տարրերը
    for (int i: X) {
        if (i < 0) {
            X_vector_negatives.push_back(i);
            continue;
        }

        if (i == 0) {
            X_vector_zeros.push_back(i);
            continue;
        }

        X_vetor_positives.push_back(i);
    }

    // Լցնում ենք Y վեկտորը
    // Սկզբից X վեկտորի բացասական տարրերով
    // Հետո զրոներով
    // Հետո X վեկտորի դրական տարրերով
    for (int i: X_vector_negatives) Y.push_back(i);
    for (int i: X_vector_zeros) Y.push_back(i);
    for (int i: X_vetor_positives) Y.push_back(i);

    // Արտածում ենք Y վեկտորի տարրերը
    for (int i: Y) cout << i << ", ";

    return 0;
}
```

Խնդիր 2

```

#include <iostream>
#include <cmath>
using namespace std;

int random(int max) {
    int min = -max;

    return min + rand() % (( max + 1 ) - min);
}

int main() {
    int n;
    int sum = 0;
    int result;

    // Ստանում ենք n-ը
    do {
        cout << "Type n: ";
        cin >> n;
    } while (n <= 0);

    int numbers[n][n];

    // Լցնում ենք numbers զանգվածը թվերով
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) numbers[i][j] = random(n);
    }

    // Ստանում ենք 0-ից մեծ տարրերի գումարը
    for (int i = 0; i < n; i++) {
        int elem = numbers[i][i];

        if (elem > 0) sum += pow(elem, 2);
    }

    // Հաշվում ենք քառակուսի արմատը
    result = sqrt(sum / n);

    // Արտածում ենք պատասխանը
    cout << "Result: " << result << endl;

    return 0;
}
```