

# Bài 32: PHÁT SINH SỐ NGẪU NHIÊN TRONG C++ (RANDOM NUMBER GENERATION)

Xem bài học trên website để ủng hộ Kteam: [Phát sinh số ngẫu nhiên trong C++ \(Random number generation\)](#)

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage [How Kteam](#) nhé!

## Dẫn nhập

Ở bài học trước, mình đã chia sẻ cho các bạn về [TỪ KHÓA BREAK & CONTINUE](#). Từ khóa **break** được dùng để kết thúc các vòng lặp **while**, **do-while**, **for** và câu điều kiện **switch**. Từ khóa **continue** sẽ **nhảy đến cuối vòng lặp hiện tại**, và **thực thi lần lặp tiếp theo**.

Trong bài hôm nay, mình sẽ giới thiệu cho các bạn về phương pháp **Phát sinh số ngẫu nhiên trong C++ (Random number generation)**. Bạn có thể ứng dụng nó vào những chương trình cần phát sinh số ngẫu nhiên, các trò chơi, hoặc để ứng dụng vào những bài học về mảng tiếp theo.

---

## Nội dung

Để đọc hiểu bài này tốt nhất các bạn nên có kiến thức cơ bản về:

- [VÒNG LẶP DO WHILE TRONG C++ \(Do while statements\)](#)

- [VÒNG LẶP FOR TRONG C++ \(For statements\)](#)

Trong bài ta sẽ cùng tìm hiểu các vấn đề:

- Tổng quan về phát sinh số ngẫu nhiên
- Phát sinh số ngẫu nhiên trong C++
- Phát sinh số ngẫu nhiên trong C++ 11

## Tổng quan về phát sinh số ngẫu nhiên

**Phát sinh các số ngẫu nhiên** được ứng dụng **rất nhiều** trong lập trình, đặc biệt là trong các trò chơi, các chương trình cần dữ liệu ngẫu nhiên, ...

**Ví dụ** về trò chơi bắn máy bay, nếu nó không có những sự kiện ngẫu nhiên, những máy bay sẽ luôn luôn xuất hiện cùng 1 vị trí, tấn công bạn theo cùng một cách, những vật thể xuất hiện trên đường không bao giờ thay đổi, vv ... và đó không phải là một trò chơi hay.



## Vậy số ngẫu nhiên được tạo ra bằng cách nào?

Trong cuộc sống, bạn thường tạo ra số ngẫu nhiên bằng cách như lắc 1 con xúc xắc, rút 1 lá thăm, tung 1 đồng xu, ... và rất nhiều vấn đề ngẫu nhiên trong cuộc sống khác.

Trong lập trình, mọi thứ đều được tạo nên từ 2 số 0 và 1, chỉ có đúng hoặc sai, không có trường hợp ở giữa. Máy tính không thể lắc 1 con xúc xắc, rút 1 lá thăm, tung 1 đồng xu, ... Kết quả mà nó đưa ra, luôn là kết quả có thể dự đoán trước, ví dụ  $1 + 1$  luôn luôn là 2, không thể là 1 giá trị khác.

Vì vậy, các máy tính không có khả năng tạo ra số ngẫu nhiên. Muốn tạo số ngẫu nhiên, lập trình viên phải tự xây dựng 1 hệ thống phát sinh số ngẫu nhiên.

Phát sinh số ngẫu nhiên là 1 vấn đề rất cần thiết trong lập trình, để đáp ứng nhu cầu đó, C++ đã xây dựng sẵn 1 số thuật toán phát sinh số ngẫu nhiên.

---

## Phát sinh số ngẫu nhiên trong C++

Ngôn ngữ C++ cung cấp 2 hàm có chức năng khởi tạo và phát sinh số ngẫu nhiên, 2 hàm này thuộc thư viện **cstdlib**:

Khởi tạo số ngẫu nhiên (initialize random number generator)

Để khởi tạo số ngẫu nhiên, bạn sử dụng hàm **srand()** thuộc thư viện **cstdlib**:

```
void srand(unsigned int seed);
```

### Lưu ý:

- Hàm **srand()** nhận vào một đối số kiểu số nguyên không dấu, được gọi là **seed** (hạt giống).
- Với mỗi **seed** khác nhau, hàm **srand()** sẽ tạo ra những bộ số ngẫu nhiên khác nhau. Những số ngẫu nhiên này sẽ được lấy ra bởi hàm **rand()**.
- Hai khởi tạo số ngẫu nhiên khác nhau với cùng một **seed** sẽ tạo ra cùng một kết quả.

- Chỉ nên gọi hàm **srand()** 1 lần trước khi phát sinh số ngẫu nhiên.

Kết quả phát sinh số ngẫu nhiên của hàm **rand()** phụ thuộc vào giá trị của **seed** (hạt giống), nếu mỗi lần khởi tạo đều sử dụng cùng 1 **seed**, các số ngẫu nhiên nhận được sẽ là như nhau.

Vì vậy, giá trị của **seed** (hạt giống) cũng phải là 1 số ngẫu nhiên trong mỗi lần truyền vào hàm **srand()**. Nghe có vẻ khá mâu thuẫn, chúng ta đang cần 1 số ngẫu nhiên để tạo ra các số ngẫu nhiên. Vậy, vấn đề là người ta phải tìm ra 1 số thay đổi mỗi khi chương trình được chạy, không phải là số do người dùng chọn.

Một giải pháp cho vấn đề này là **dựa trên thời gian hệ thống**. Mỗi lần chương trình được chạy, thời gian sẽ khác nhau. Nên người ta lấy giá trị thời gian hệ thống làm **seed** (hạt giống), kết quả sẽ là những số ngẫu nhiên khác nhau trong mỗi lần chạy chương trình.

Để lấy được thời gian từ hệ thống, bạn có thể sử dụng hàm **time()** thuộc thư viện **ctime**. Hàm này sẽ trả về số giây từ 00:00 giờ, ngày 01 tháng 1 năm 1970.

### Ví dụ:

```
#include <iostream>
#include <cstdlib> // for rand() and srand()
#include <ctime> // for time()
using namespace std;

int main()
{
    // initialize random number generator
    srand(time(0)); // set initial seed value to system clock

    // generate random number
    // ...

    return 0;
}
```

Ví dụ trên chỉ mới khởi tạo số ngẫu nhiên từ thời gian hệ thống, cách phát sinh các số ngẫu nhiên sẽ được giới thiệu ở phần tiếp theo.

# Phát sinh số ngẫu nhiên (generate random number)

Để phát sinh 1 số ngẫu nhiên, bạn sử dụng hàm **rand()** thuộc thư viện **cstdlib**:

```
int rand(void);
```

```
int rand(void);
```

## Lưu ý:

- Hàm **rand()** trả về 1 số nguyên ngẫu nhiên trong khoảng từ **0** đến **RAND\_MAX**.
- RAND\_MAX** là 1 hằng số có giá trị 32767, được định nghĩa trong thư viện **cstdlib**.

**Ví dụ** về phát sinh số ngẫu nhiên:

```
int v1 = rand();           // v1 in the range 0 to 32767
int v2 = rand() % 100;      // v2 in the range 0 to 99
int v3 = rand() % 100 + 1; // v3 in the range 1 to 100
int v4 = rand() % 30 + 1985; // v4 in the range 1985-2014
```

Một số ví dụ về phát sinh số ngẫu nhiên

**Ví dụ** về trò chơi đoán số từ 1 số phát sinh ngẫu nhiên:

```
#include <iostream>
#include <cstdlib> // for rand() and srand()
#include <ctime>   // for time()
using namespace std;

int main()
{
    int nSecret, nGuess;

    // initialize random seed
```

```
    srand(time(NULL));

    // generate secret number between 1 and 10
    nSecret = rand() % 10 + 1;

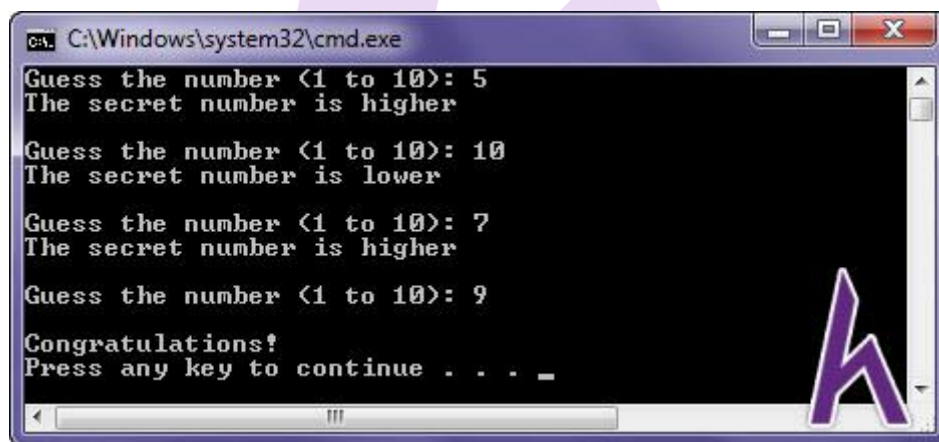
    do {
        cout << "Guess the number (1 to 10): ";
        cin >> nGuess;

        if (nSecret < nGuess)
            cout << "The secret number is lower" << endl;
        else if (nSecret > nGuess)
            cout << "The secret number is higher" << endl;
    } while (nSecret != nGuess);

    cout << "Congratulations!" << endl;

    return 0;
}
```

### Output:



Chương trình trên phát sinh một số ngẫu nhiên từ 1 đến 10, và yêu cầu người dùng lặp lại việc chọn 1 số sao cho trùng với số ngẫu nhiên của hệ thống.

**Ví dụ** phát sinh một dãy 10 chữ số ngẫu nhiên:

```
#include <iostream>
#include <cstdlib> // for rand() and srand()
#include <ctime> // for time()
```

```
using namespace std;

int main()
{
    srand(time(0)); // set initial seed value to system clock

    for (int count = 0; count < 10; ++count)
    {
        cout << rand() << "\t";

        // If we've printed 5 numbers, start a new row
        if ((count + 1) % 5 == 0)
            cout << endl;
    }

    return 0;
}
```

### Output:



## Phát sinh số ngẫu nhiên trong C++ 11

**C++ 11** cung cấp thêm rất nhiều thuật toán phát sinh số ngẫu nhiên thuộc thư viện **random**.

**Ví dụ** về 1 thuật toán phát sinh số ngẫu nhiên Mersenne Twister thường được sử dụng:

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

int main()
{
    random_device rd;          // only used once to initialize (seed) engine
    mt19937 rng(rd());          // random-number engine used (Mersenne-
    Twister in this case)

    // output 10 random number
    for (int i = 0; i < 10; i++)
    {
        auto random_integer = rng();
        cout << random_integer << "\t";

        if ((i + 1) % 5 == 0)
            cout << endl;
    }

    // output 10 random number in the range 1 to 100
    uniform_int_distribution<int> uni(1, 100);
    for (int i = 0; i < 10; i++)
    {
        auto random_integer = uni(rng);
        cout << random_integer << "\t";

        if ((i + 1) % 5 == 0)
            cout << endl;
    }

    return 0;
}
```

**Output:**





```
C:\Windows\system32\cmd.exe
1464330171 3890418801 3780955317 40949
2042117381 3422918438 2231036862 18787
91 99 18 54 34
70 7 45 13 85
Press any key to continue . . .
```

Thuật toán trên tạo ra các **số nguyên không dấu 32 bit**, nên sẽ có phạm vi lớn hơn rất nhiều so với sử dụng hàm **rand()**. Bạn cũng có thể sử dụng kiểu **mt19937\_64** cho biến **rng** để có phạm vi số lớn hơn (**64 bit**).

Ví dụ trên có sử dụng từ khóa **auto**, chi tiết về nó sẽ được hướng dẫn trong bài [TỪ KHÓA AUTO TRONG C++ \(The auto keyword\)](#).

Vẫn còn rất nhiều thuật toán phát sinh số ngẫu nhiên khác, các bạn có thể tìm hiểu thêm và chia sẻ lại cho mọi người nhé.

---

## Kết luận

Qua bài học này, bạn đã hiểu rõ về phương pháp **Phát sinh số ngẫu nhiên trong C++ (Random number generation)**. Bạn có thể ứng dụng nó vào những chương trình cần phát sinh số ngẫu nhiên, các trò chơi, hoặc để ứng dụng vào những bài học về mảng tiếp theo.

Trong bài tiếp theo, mình sẽ giới thiệu cho các bạn về [MẢNG 1 CHIỀU TRONG C++ \(Arrays\)](#).

Cảm ơn các bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên **"Luyện tập – Thử thách – Không ngại khó"**.