



```
#include <cstdlib>
#include <iostream>

using namespace std;

int main ()
{
    srand(time(nullptr));

    cout << "RAND_MAX: " << RAND_MAX << '\n';

    for (int i = 0; i < 5; ++i)
    {
        cout << rand() << '\n';
    }

    for (int i = 0; i < 5; ++i)
    {
        cout << rand() % 100 << '\n';
    }

    for (int i = 0; i < 5; ++i)
    {
        cout << rand() / (double) RAND_MAX << '\n';
    }
}
```

```
#include <random>
#include <iostream>

using namespace std;

int muito_errado ()
{
    mt19937 gerador; // Demorado e sempre começa igual!

    return gerador();
}

int melhor (mt19937 &gerador)
{
    return gerador(); // Ok, mas podia chamar logo diretamente.
}

class MeuGerador
{
    mt19937 gerador;

public:
    MeuGerador () /* : gerador{} */ { }

    int numero () // Não precisa de argumento.
    {
        return gerador();
    }
};

int main ()
{
    mt19937 gerador;

    for (int i = 0; i < 10; ++i) cout << ' ' << gerador();
    cout << "\n\n";

    for (int i = 0; i < 10; ++i) cout << ' ' << muito_errado();
    cout << "\n\n";

    for (int i = 0; i < 10; ++i) cout << ' ' << melhor(gerador);
    cout << "\n\n";

    MeuGerador meu;

    for (int i = 0; i < 10; ++i) cout << ' ' << meu.numero();
    cout << "\n\n";
}
```

```
#include <random>
#include <iostream>

using namespace std;

class MeuGerador
{
    mt19937 gerador;
    uniform_int_distribution<int> distr;

public:
    MeuGerador () : gerador{}, distr{100,200} { }

    int operator() () // int numero ()
    {
        return distr(gerador);
    }
};

int main ()
{
    mt19937 gerador;

    uniform_int_distribution distr(100,200);

    for (int i = 0; i < 10; ++i)
    {
        cout << ' ' << distr(gerador);
    }
    cout << "\n\n";

    uniform_real_distribution dr(5.0,10.0);

    for (int i = 0; i < 10; ++i)
    {
        cout << ' ' << dr(gerador);
    }
    cout << "\n\n";

    MeuGerador g;

    for (int i = 0; i < 10; ++i)
    {
        cout << ' ' << g(); // g.numero();
    }
    cout << "\n\n";
}
```

```
#include <random>
#include <iostream>

using namespace std;

int main ()
{
    random_device disp_aleat;

    mt19937 gerador{ disp_aleat() }; // Semente Aleatória

    uniform_int_distribution distr{0,10};

    for (int i = 0; i < 10; ++i)
    {
        cout << ' ' << distr(gerador);
    }

    cout << '\n';
}
```

4 Sementes.cpp 8,1 Tudo

```
#include <chrono>
#include <iostream>

using namespace std;
using namespace chrono;

int main ()
{
    int n; cout << "n: ";

    auto inicio = steady_clock::now();

    cin >> n;

    auto fim = steady_clock::now();

    duration<double> d = fim - inicio;

    cout << "Você demorou " << d.count()
        << " segundos para digitar.\n";

    duration<double, ratio<60>> min = d;

    cout << "Você demorou " << min.count()
        << " minutos para digitar.\n";

    cout << "Observação: min*60 = "
        << min.count() * 60
        << '\n';
}
```