

## Tekrar:

- Random kütüphanesi modern C++ ile kullanılır.
- Eger resimdeki gibi dağılımını unutmaz ise C diliinde rand() yeterdir
- Dağılım ne random bit üretimi motorları farklı,
- Deterministic / True random generatorlar var  
 $mt19937$        $\downarrow$   
random device

## - Discard Fonksiyonu:

```
int main()
{
    using namespace std;
    std::mt19937 eng;
    eng.discard(_Nskip: 9999);
    cout << eng() << "\n";
}
```

Random generate edilmiş 9888 sayısını  
discard eder.

10000. değerini cout'a verir.

## + Uniform int Distribution:

→ Tüm sayıların gelme olasılığı aynı.

```
int main()
{
    using namespace std;
    std::mt19937 eng;
    uniform_int_distribution<int> dist{ 1, 5};
    dist(eng)
}
```

default argument olarak int  
yazınca da int oluyor  
distribution adı!

CPP 17 ile  
CTAD'in da yorumlanabilirliği

std::mt19937 eng;

//uniform\_int\_distribution<int> dist{ 1, 6};  
//uniform\_int\_distribution<> dist{ 1, 6};  
uniform\_int\_distribution dist{ 1, 6}; CTAD

```
#include <chrono>
#include "nutility.h"
#include <fstream>
#include <map>
```

```
int main()
{
    using namespace std;
    std::mt19937 eng;
    uniform_int_distribution<int> dist{ 1, 6};
    map<int, int> cmap;
    for (int i = 0; i < 10000; ++i) {
        ++cmap[dist(eng)];
    }
    for (const auto [dice, count] : cmap) {
        cout << dice << " " << count << '\n';
    }
}
```

Microsoft Visual Studio Debug Console  
1 1682  
2 1613  
3 1683  
4 1663  
5 1688  
6 1671

C:\hec\1a.exe (process 20024) exited with code 0.  
Press any key to close this window . . .

sayı ortaya da da uniform  
dağılım gösterenmiş.

```

int main()
{
    using namespace std;

    mt19937 eng{ 353234 };

    vector<int> ivec(10000);
    generate(ivec.begin(), ivec.end(), eng);
}

ofstream ofs{ "random.txt" };
// print(ivec, "\n", ofs);

```

generate in 3. parametresi  
 referans sənəti - ne olmır!  
 => Yani mt19937 (3000 byte) lik  
 növ sənəti - ne olmır!

Bunu C02OMÜ  
 Rekurren wrapper!

```

#include <iostream>
#include <random>
#include <chrono>
#include "nutility.h"
#include <fstream>
#include <map>
#include <algorithm>
#include <vector>
#include <functional>

int main()
{
    using namespace std;

    mt19937 eng{ 353234 };

    vector<unsigned int> ivec(10000);

    generate(ivec.begin(), ivec.end(), ref(eng));
}

ofstream ofs{ "random.txt" };
// print(ivec, "\n", ofs);

```

reference  
 wrapper  
 kullanımlı!

```

int main()
{
    using namespace std;

    mt19937 eng{ 353234 };
    uniform_int_distribution dist{ 100, 500 };
    auto f = [&]() {return dist(eng); };

    vector<unsigned int> ivec(10000);

    generate(ivec.begin(), ivec.end(), f);

    ofstream ofs{ "random.txt" };
    // print(ivec, "\n", ofs);

```

Aşağıda uniform  
 distribution eklenmiş hali  
 - aynı bir lambda ifadesi (function object)  
 olup, onu predicate olarak geçmemiz gerekiyor!

```

using namespace std;

mt19937 eng{ 353234 };
uniform_int_distribution dist{ 100, 500 };
auto f = [&]() {return dist(eng); };

vector<int> ivec;
ivec.reserve(1000);

generate_n(ivec.begin(), 1000, f);

```

back inserter ob  
 jektini kullanırsınız.

\* Generate\_n Fonksiyonu Kullanımı: → generate'dan farklı, əsaslıdır. iterator begin, tətbiq olunacaq element sayısi, predicate

\* Shuffle: - BİF VET yapısının içeriğinin sırasını değiştirt.

```
7 #include <algorithm>
8 #include <vector>
9 #include <functional>
10 #include <iterator>
11 #include <string>
12 #include <conio.h>
13
14
15
16 int main()
17 {
18     using namespace std;
19
20     string str{ "volkangundogdu" };
21
22     mt19937 eng{ 345u };
23
24     for (;;) {
25         cout << str;
26         shuffle(str.begin(), str.end(), eng);
27         _getch();
28         system("cls");
29     }
}
```

\* Saat 1.07 ~ gibi universal reference tekrar etti.  
Açık bir done refe

\* Bir container'da random ve unique done olurken注意:

```
int main()
{
    using namespace std;

    //3400 3800
    mt19937 eng{ random_device{}() };
    uniform_int_distribution dist{ 3400, 3800 };

    set<int> myset;
    while (myset.size() != 200) { } } set kullanır
    myset.insert(dist(eng)); } uniquesness done edildi.

    vector<int> ivec{ myset.begin(), myset.end() };
    shuffle(ivec.begin(), ivec.end(), eng);
    print(ivec);
}
```

\* Uniform Real Distribution: - C de gerenksayir rəm distribution yolu

```
18 {
19     using namespace std;
20
21
22     mt19937 eng{ random_device{}() };
23     //uniform_real_distribution<double> dist{ 0., 10. };
24     //uniform_real_distribution<> dist{ 0., 10. };
25     uniform_real_distribution dist{ 0., 10.};
26     auto f = [&](){return dist(eng); };
27
28     vector<double> dvec(100'000);
29     generate(dvec.begin(), dvec.end(), f);
30     std::ofstream ofs{ "random.txt" };
31     if (!ofs) {
32         std::cerr << "out.txt dosyasi olusturulamadi\n";
33         exit(EXIT_FAILURE);
34     }
35
36     ofs << fixed << setprecision(6);
37
38     for (auto dval : dvec)
39         ofs << dval << '\n';
```

\* Bernoulli Distribution: → template değil!

→ direkt bin class

→ var olma amaci boolean diger istenmes

```
#include <algorithm>
#include <vector>
#include <functional>
#include <iostream>
#include <string>
#include <conio.h>
#include <set>
#include <iomanip>
#include <random>

int main()
{
    using namespace std;

    mt19937 eng{ random_device{}() };

    bernoulli_distribution bdist, [ ] { constructor a 0,25 gibi
                                         deşur secret
                                         true false bras
                                         verebiliriz! };

    map<bool, int> cmap;
    int true_cnt{};
    int false_cnt{};
    for (int i = 0; i < 1000'000; ++i) {
        if (bdist(eng))
            ++true_cnt;
        else
            ++false_cnt;
    }

    cout << true_cnt << "\n";
    cout << false_cnt << "\n";
}
```

## \* Discrete Distribution:

The screenshot shows a Microsoft Visual Studio interface. On the left is the code editor with the following C++ code:

```
#include <vector>
#include <functional>
#include <iterator>
#include <string>
#include <conio.h>
#include <set>
#include <iomanip>
#include <iostream>

int main()
{
    using namespace std;

    mt19937 eng{ random_device{}() };

    discrete_distribution<int> dist{ 10, 20, 40, 30 };

    map<int, int> cmap;

    for (int i = 0; i < 100'000; ++i) {
        ++cmap[dist(eng)];
    }

    for (const auto [val, cnt] : cmap) {
        cout << val << " " << cnt << '\n';
    }
}
```

A red arrow points from the handwritten note "deegim yozdurur" to the line `discrete_distribution<int> dist{ 10, 20, 40, 30 };`. The Microsoft Visual Studio Debug Console window is open on the right, showing the following output:

```
Microsoft Visual Studio Debug Console
0 10041
1 20007
2 39913
3 30039

C:\nec\ a.exe (process 23528) exited
Press any key to close this window .
```

The screenshot shows a Microsoft Visual Studio interface with the same C++ code as the first screenshot. Handwritten annotations are present:

- A red arrow points from the handwritten note "range constructor" to the line `discrete_distribution<int> dist{begin(weights) ,end(weights)};`.
- A yellow circle highlights the variable `ar[val]` in the line `cout << ar[val] << " " << cnt << '\n';`.

## \* Raw String Literal:

```
#include <conio.h>
#include <set>
#include <iomanip>
#include <iostream>

int main()
{
    std::cout << R"("murat" "eray" \necati\korhan")";
}
```

Okutulma seviye  
kole

escape karakter  
kullanarak sırası koruyor  
\ ve " işin.

```
int main()
{
    const char* p = R"(ahmet aksoy
    eray goksu
    kerim fidan
    volkan gundogdu
    necati ergin
    )";
    puts(p);
}
```

→ Bu şekilde yazılığında alt-ta ve  
hizalı yordur. Farklı oyunculardır.

```
int main()
{
    using namespace std;
    cout << R"123(alii")123" << "\n";
}
```

Sıradı delimiter 123 oldu. Boyalı  
"1" de yordanlobırr.

## \* Ratio Header File:

```
int main()
{
    using namespace std;

    constexpr auto x = ratio<5, -45>::den;
```

• den: 9

• num: -1 our bu örnekte num.

Oranı  
verir.

I

## → Ratio Add:

```
int main()
{
    using namespace std;

    ratio_add<ratio<5, 9>, ratio<4, 7>>::type
```

• type: ratio

• some: 3/9 + 4/7

## → Ratio Multiply:

```
int main()
{
    using namespace std;

    using one_fifth = ratio<1, 5>;
    using two_third = ratio<1, 3>;

    ratio_multiply<one_fifth, two_third>::type::num
```

→ Aşağıda subtractive multiply ver. Birkaçı hatalarla complete formda olan eide edilebilir..

## \* Ratio Yüklenecek Blend Editor Type Names:

```
int main()
{
    using namespace std;

    cout << typeid(milli).name() << "\n";
    cout << typeid(centi).name() << "\n";
    cout << typeid(nano).name() << "\n";
    cout << typeid(micro).name() << "\n";
    cout << typeid(picoc).name() << "\n";
    cout << typeid(kilo).name() << "\n";
    cout << typeid(deca).name() << "\n";
    cout << typeid(deci).name() << "\n";
```

```
Microsoft Visual Studio Debug Console
struct std::ratio<1,1000>
struct std::ratio<1,100>
struct std::ratio<1,1000000000>
struct std::ratio<1,100000>
struct std::ratio<1,1000000000000>
struct std::ratio<1000,1>
struct std::ratio<10,1>
struct std::ratio<1,10>
C:\nec\aa.exe (process 26588) exited with code 0.
Press any key to close this window . . .
```