Ders 22

```
int main()
{
    srand(35u);

    for (int i = 0; i < 10; ++i) {
        printf("%d\n", rand());
    }
```

35 tohum değeri ile rand sayı üretti

```
int main()
{
    srand((unsigned)time(0));

    for (int i = 0; i < 10; ++i) {
        printf("%d\n", rand());
    }

}
```

Unsigned time deme nededimiz long long olan time fonksiyonunu srand argümanı için unsigned yapmak .

Gecen sürede değişecek

ÖRNEK MÜLAKAT SORUSU SRAND

```c
#define MIN_PSW_LEN         5
#define MAX_PSW_LEN         11

void print_random_psw(void)
{
    int len;

    len = rand() % (MAX_PSW_LEN - MIN_PSW_LEN + 1) + MIN_PSW_LEN;
    for (int i = 0; i < len; ++i) {
        putchar((rand() % 2 ? 'A' : 'a') + (rand() % 26));
    }
    putchar('\n');
}

int main()
{
    srand((unsigned)time(0));

    for (int i = 0; i < 10; ++i) {
        print_random_psw();
    }
}
```

Her seferinde farklı parolalar gelmekte

```c
#define MIN_PSW_LEN         5
#define MAX_PSW_LEN         11

void print_random_psw(void)
{
    int len;

    srand((unsigned)time(0));
    len = rand() % (MAX_PSW_LEN - MIN_PSW_LEN + 1) + MIN_PSW_LEN;
    for (int i = 0; i < len; ++i) {
        putchar((rand() % 2 ? 'A' : 'a') + (rand() % 26));
    }
    putchar('\n');
}

int main()
{

    for (int i = 0; i < 10; ++i) {
        print_random_psw();
    }
}
```

Srand fonksiyon içinde yazdım ve paralolar her seferinde aynı gelmeye başladı.

Fonksiyon çok hızlı çağırılıyor ve her seferinde aynı saniye değeri ile tohum değer

Veriliyor sık yapılan hata

Örnek iç içe döngü

```c
#define _CRT_SECURE_NO_WARNINGS

#include <stdio.h>
#include <stdlib.h>
#include <time.h>


int main()
{
    for (int i = 1; i < 100; ++i) {
        srand(i);
        printf("random number chain for seed value %d\n", i);
        for (int k = 0; k < 20; ++k) {
            printf("%d\n", rand());
        }
        (void)getchar();
        system("cls");
    }

}
```

Buradaiç içe döngüde srand ile tohum değerilerini değiştirerek farklı 20 sayı ürettik.

Daha sonra bu işlemi sıra ile yapmak için getchar kullandık.

Her entera basdığımıda bir sonraki sıraya gidiyor.

System(cls) ekranı siliyoruz daha görmedik.

Örnek pi sayısına yaklaşma kare daire

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

int main()
{
    //double x = 0 - 1 araliginda rastgele gercek sayi
    //double x = 0 - 1 araliginda rastgele gercek sayi

    /*
        if (x * x + y * y <= 1.)
            ++inside_counter;

    */


}
```

Koordinatlar için 2 değişken

İf içinde dairenin içindemi değilmi

Kodu yazmaya başlayalım

```cpp
#include <random>

#define    NPOINTS          1000000000

int main()
{
    std::mt19937 eng;
    std::uniform_real_distribution dist{ 0., 1. };

    int inside_count = 0;

    for (int i = 0; i < NPOINTS; ++i) {
        double x = dist(eng);
        double y = dist(eng);
        if (x * x + y * y <= 1.)
            ++inside_count;
    }

    printf("pi ~= %.12f\n", 4. * inside_count / NPOINTS);


}
```

Printf içindeki ifadede neden dönüşüm yapmadık

Çünkü solda sağa giderken ilk işlem double olur sonra kalan ifade double bölü int olcağı için örtülü double dönüşümü olur.

Veri yapıları ve algoritma

```cpp
for (int i = 0; i < n; ++i) {

}
```

Doğru orantılı ise o(n)

```cpp
for (int i = 0; i < n; ++i) {
    for (int i = 0; i < n; ++i) {
}
```

O(n2)

Algoritma karmaşıklığı

Algorithm	Best Time Complexity	Average Time Complexity	Worst Time Complexity	Worst Space Complexity
Linear Search	O(1)	O(n)	O(n)	O(1)
Binary Search	O(1)	O(log n)	O(log n)	O(1)
Bubble Sort	O(n)	O(n^2)	O(n^2)	O(1)
Selection Sort	O(n^2)	O(n^2)	O(n^2)	O(1)
Insertion Sort	O(n)	O(n^2)	O(n^2)	O(1)
Merge Sort	O(nlogn)	O(nlogn)	O(nlogn)	O(n)
Quick Sort	O(nlogn)	O(nlogn)	O(n^2)	O(log n)
Heap Sort	O(nlogn)	O(nlogn)	O(nlogn)	O(n)
Bucket Sort	O(n+k)	O(n+k)	O(n^2)	O(n)
Radix Sort	O(nk)	O(nk)	O(nk)	O(n+k)
Tim Sort	O(n)	O(nlogn)	O(nlogn)	O(n)
Shell Sort	O(n)	O((nlog(n))^2)	O((nlog(n))^2)	O(1)