Lecture 13:
Random Numbers

PIC 10A
Todd Wittman

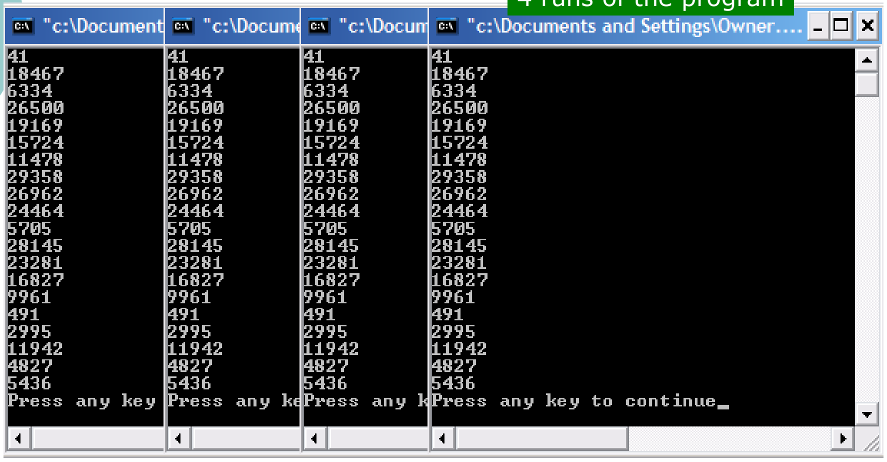
Sec 3.11: Random Numbers

- The rand() function creates a random integer between 0 and RAND_MAX, a constant about 32,000.
- To generate a random integer in the range [a,b] use

```
a + rand()%(b-a+1);
```

- Ex Simulate the roll of a die.
- Ex Generate a random even integer between 2 and 10.

```
int i = 1;
while (i <= 20) {
    int rand_num = rand();
    cout << rand_num << "\n";
    i++;
}
```



```
41             41          41         41
18467          18467       18467      18467
6334           6334        6334       6334
26500          26500       26500      26500
19169          19169       19169      19169
15724          15724       15724      15724
11478          11478       11478      11478
29358          29358       29358      29358
26962          26962       26962      26962
24464          24464       24464      24464
5705           5705        5705       5705
28145          28145       28145      28145
23281          23281       23281      23281
16827          16827       16827      16827
9961           9961        9961       9961
491            491         491        491
2995           2995        2995       2995
11942          11942       11942      11942
4827           4827        4827       4827
5436           5436        5436       5436
Press any key  Press any kePress any kPress any key to continue_
```

4 runs of the program

Gives the same output every time we run it!

Seeding Your Randomness

- The problem is that the rand() function doesn’t really generate a random number.
- It just reads a number from a really long built-in list of integers.
- So it always reads the same list every time we restart the program.

- To make it look more random, we should start reading the list from a different spot.
- This is called *seeding* the random number generator.

Seeding Your Randomness

- The srand(int) function starts reading the list at the seed position specified by the passed int.

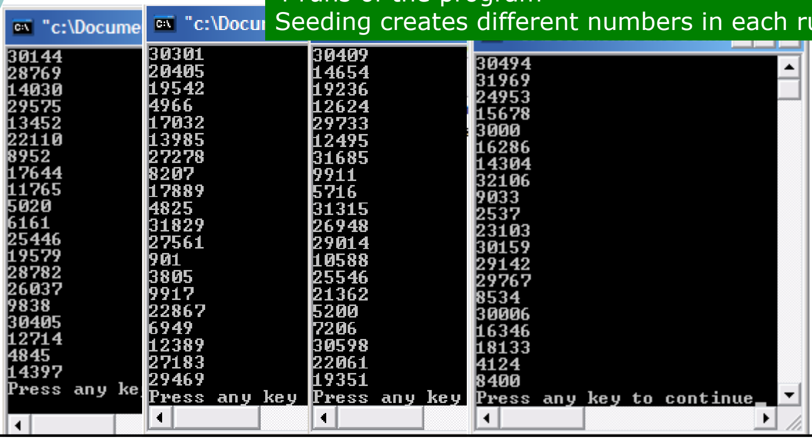
    srand(10);

- Starts reading at position 10.
- Ideally, we should change the seed every time. An easy way to do this is to use the current time.
- If we #include <ctime>, we can use the time(int) function.
- time(0) returns the number of seconds since January 1, 1970.

    srand((int) time(0));

- We need to cast it to int because time returns a special number of type time_t.

```
srand( (int) time(0) );
int i = 1;
while (i <= 20) {
    int rand_num = rand();
    cout << rand_num << "\n";
    i++;
}
```



```
30144
28769
14030
29575
13452
22110
8952
17644
11765
5020
6161
25446
19579
28782
26037
9838
30405
12714
4845
14397
Press any ke
```

```
30301
20405
19542
4966
17032
13985
27278
8207
17889
4825
31829
27561
901
3805
9917
22867
6949
12389
27183
29469
Press any key
```

```
30409
14654
19236
12624
29733
12495
31685
9911
5716
31315
26948
29014
10588
25546
21362
5200
7206
30598
22061
19351
Press any key
```

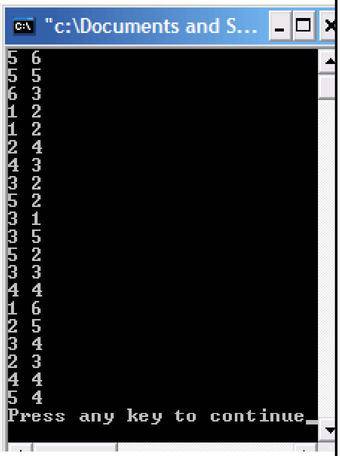
```
30494
31969
24953
15678
3000
16286
14304
32106
9033
2537
23103
30159
29142
29767
8534
30006
16346
18133
4124
8400
Press any key to continue
```

4 runs of the program
Seeding creates different numbers in each run.

Rolling Dice

- To simulate rolling dice 20 times, we should create two random numbers between 1 and 6.

```
srand( (int) time(0) );
int k = 1;
while (k <= 20) {
     int die1 = 1+rand()%6;
     int die2 = 1+rand()%6;
     cout << die1 <<" "
          << die2 <<"\n";
     k++;
}
```



```
5 6
5 5
6 3
1 2
1 2
2 4
4 3
3 2
5 2
3 1
3 5
5 2
3 3
4 4
1 6
2 5
3 4
2 3
4 4
5 4
Press any key to continue_
```

Playing Craps

- In the casino game of craps, a shooter rolls until a sum of 7 is rolled.
- Then the dice are passed to the next player.

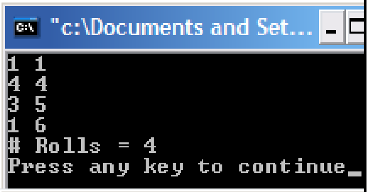
- On average, how many rolls does a player's turn last?
- This is a very difficult mathematical question, but we can try to answer it by simulating it numerically.

Simulating One Run

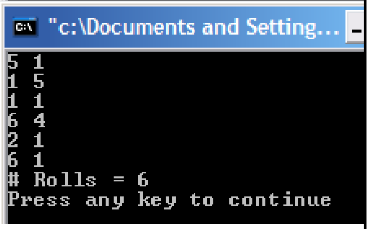
- Roll until the sum is 7.

```
srand( (int) time(0) );
int rolls = 0;
int sum = 0;
while (sum != 7) {
   int die1 = 1+rand()%6;
   int die2 = 1+rand()%6;
   cout << die1 <<" "<< die2 <<"\n";
   rolls++;
   sum = die1 + die2;
}
cout << "# Rolls = " << rolls << "\n";
```

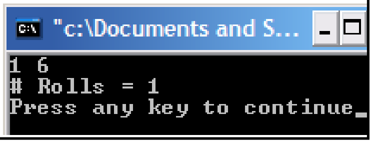
- But this doesn’t answer our question!



```
1 1
4 4
3 5
1 6
# Rolls = 4
Press any key to continue_
```



```
5 1
1 5
1 1
6 4
2 1
6 1
# Rolls = 6
Press any key to continue
```

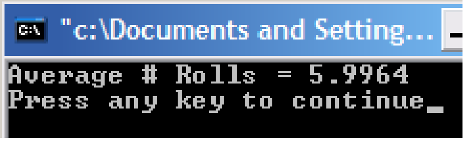


```
1 6
# Rolls = 1
Press any key to continue_
```

Averaging Over Many Runs

- Average the last program over 10,000 runs.

```
srand( (int) time(0) );
int total_rolls = 0;
int run = 1;
while (run <= 10000) {
   int sum = 0;
   while (sum != 7) {
         int die1 = 1+rand()%6;
         int die2 = 1+rand()%6;
         total_rolls++;
         sum = die1 + die2;
   }
   run++;
}
cout << "Average # Rolls = "
      << (double) total_rolls/10000 << "\n";
```



```
Average # Rolls = 5.9964
Press any key to continue_
```

- It appears the average craps turn lasts 6 rolls.

Drawing Cards

- To simulate drawing a random card, we need to generate a suit (Clubs, Diamonds, Hearts, Spades) and rank (Ace, One, Two, ..., Queen, King).
- Here’s how we can pick a random suit.

```
string suit;
int suit_number = 1+rand()%4;
switch(suit_number) {
   case 1:  suit = "Clubs";  break;
   case 2:  suit = "Diamonds";  break;
   case 3:  suit = "Hearts";  break;
   case 4:  suit = "Spades";  break;
}
```

- I’ll let you figure out how to pick a random rank.