More Arrays

Last Time

- We
  - Learnt how to pass arrays to functions
- Today we will
  - Start talking about sorting arrays

Histogram

- Write a program that will output the contents of the array in the form of a histogram

```
Element      Value    Histogram

      0          7    *******

      1         12    ************

      2          3    ***

      3          5    *****
```


Random Number Generation

- The library <cstdlib> contains a function for generating random numbers
- For example, the statement used to produce integers in the range 0 - 5 is
  - `int x = rand() % 6;`
- To simulate the role of a dice we would use the statement
  - `int x = 1 + rand() % 6`

Seeded Random Numbers

- Function rand generates pseudo-random numbers
- The function produces the same numbers every time the program runs
- Use the function srand to produce true random numbers
- srand needs an integer argument to seed the rand function to produce a different sequence of numbers for each program execution

What Does the Program Do?

```
int main()
{
  unsigned seed;
  cout << "Enter seed" << endl;
  cin >> seed;
  srand(seed);
  for( int i=1; i<=10; i++ )
    cout << setw(10) << (1+rand()%6);
  return 0;
}
```

Randomizing

- Is there a way of finding a true random number without asking the user for a seed?
- Best thing is to use the calendar time. This uses the date and the time to produce a unique unsigned int
- Need to include **<ctime>** and the function **time(0)**
  - `srand( time( 0 ) );`

Random Number Generation

- Write a program that will simulate the roll of a dice 6000 times and show the frequency in which each side appeared

Face	Frequency
1	1003
2	1017
3	983
4	994
5	1004
6	999

Sorting Arrays

- Bubble sort
- Not suitable for large arrays
- Smaller values gradually bubble their way upward to the top of the array

Bubble Sort

```
for( int pass=0; pass < size - 1; pass ++ )
  for( int j=0; j < size - 1; j++ )
    if( a[j] > a[j+1] )
    {
      hold = a[j];
      a[j] = a[j+1];
      a[j+1] = hold;
    }
```


Summary

- In today's lecture we covered
  - Random numbers
  - Sorting arrays (bubble sort)
- Readings
  - P. 276 - p. 278: Bubble sort
  - P. 262 - p. 264: Histograms
  - P. 182 - p. 188: Random number generation