Some Examples Using Arrays

- Sorting
- Fibonacci Sequence
- Coin Tossing
- Decimal to Binary Conversion
- Binary to Decimal Conversion

Source: Suffolk Maths

Suppose a newly-born pair of rabbits, one male, one female, are put in a field.

Rabbits are able to mate at the age of one month. So at the end of its second month a female can produce another pair of rabbits.

Suppose that our rabbits never die. And the female always produces one new pair (one male, one female) every month from the second month on.

The puzzle that I posed was...

How many pairs will there be in one year?

Pairs
1 pair

At the end of the first month there is still only one pair

Pairs

1 pair

End first month… only one pair

1 pair

At the end of the second month the female produces a new pair, so now there are 2 pairs of rabbits

2 pairs

Pairs

1 pair

End first month… only one pair

1 pair

End second month… 2 pairs of rabbits

2 pairs

At the end of the third month, the original female produces a second pair, making 3 pairs in all in the field.

3 pairs

Pairs

1 pair

End first month… only one pair

1 pair

End second month… 2 pairs of rabbits

2 pairs

3 pairs

End third month… 3 pairs

5 pairs

At the end of the fourth month, the first pair produces yet another new pair, and the female born two months ago produces her first pair of rabbits also, making 5 pairs.

Recursive Definition

- Fibonacci sequence
  - F[0] = 0, F[1] = 1, F[n] = F[n-1] + F[n-2]
  - 0 1 1 2 3 5 8 13 21 34 55 89 144 …
- Lucas sequence
  - L[0] = 2, L[1] = 1, L[n] = L[n-1] + L[n-2]
  - 2 1 3 4 7 11 18 29 47 76 123 199 …
- You may write a program to verify
  - L[n] == F[n+2] - F[n-2]

Expected Result

```
Fibonacci sequence:
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181
Lucas sequence:
2 1 3 4 7 11 18 29 47 76 123 199 322 521 843 1364 2207 3571 5778 9349
========================
L[ 2]=   3      F[ 4]=   3      F[ 0]=  0       3 == 3 - 0
L[ 3]=   4      F[ 5]=   5      F[ 1]=  1       4 == 5 - 1
L[ 4]=   7      F[ 6]=   8      F[ 2]=  1       7 == 8 - 1
L[ 5]=  11      F[ 7]=  13      F[ 3]=  2       11 == 13 - 2
L[ 6]=  18      F[ 8]=  21      F[ 4]=  3       18 == 21 - 3
L[ 7]=  29      F[ 9]=  34      F[ 5]=  5       29 == 34 - 5
L[ 8]=  47      F[10]=  55      F[ 6]=  8       47 == 55 - 8
L[ 9]=  76      F[11]=  89      F[ 7]= 13       76 == 89 - 13
L[10]= 123      F[12]= 144      F[ 8]= 21       123 == 144 - 21
L[11]= 199      F[13]= 233      F[ 9]= 34       199 == 233 - 34
L[12]= 322      F[14]= 377      F[10]= 55       322 == 377 - 55
L[13]= 521      F[15]= 610      F[11]= 89       521 == 610 - 89
L[14]= 843      F[16]= 987      F[12]=144       843 == 987 - 144
L[15]=1364      F[17]=1597      F[13]=233       1364 == 1597 - 233
L[16]=2207      F[18]=2584      F[14]=377       2207 == 2584 - 377
L[17]=3571      F[19]=4181      F[15]=610       3571 == 4181 - 610
```

Sample Code (fibonacci.cpp)

```
int n = 0;
const int M = 20;
int L[M] = {2, 1};
int F[M] = {0, 1};

for (n=2; n<M; n++)
{
     L[n] = L[n-1] + L[n-2];
     F[n] = F[n-1] + F[n-2];
}

cout << "Fibonacci sequence: " << endl;
for (n=0; n<M; n++)
     cout << F[n] << " ";
cout << endl;

cout << "Lucas sequence: " << endl;
for (n=0; n<M; n++)
     cout << L[n] << " ";

cout << endl << "=========================" << endl;

 for (n=2; n<M-2; n++)
     cout << "L[" << setw(2) << n << "]=" << setw(4) << L[n] << "\t"
             << "F[" << setw(2) << n+2 << "]=" << setw(4) << F[n+2] << "\t"
             << "F[" << setw(2) << n-2 << "]=" << setw(3) << F[n-2] << "\t"
             << L[n] << (L[n]==F[n+2]-F[n-2]?" == ":" != ")
             << F[n+2] << " - " << F[n-2]
             << endl;
```

Sample Code (exchange_sort.cpp)

```
const int MAX = 4;
int data[MAX] = { 9, 5, 7, 2};
int i, j, temp;

cout << "Before soring:\t";
for (i=0; i<MAX; i++)
        cout << data[i] << '\t';

for (j=0; j<=MAX-2; j++)
{
        cout << endl << "Round " << j+1 << ":\t";
        for (i=j+1; i<=MAX-1; i++)
        {
                if (data[j] > data[i])
                {
                        temp = data[j]; data[j]=data[i]; data[i]=temp;
                }
        }
        for (i=0; i<MAX; i++)
                cout << data[i] << '\t';
}
```

Example: Coin Tossing

- A coin has two sides – Head/Tail
  - 0/1
- Repeat tossing the coin 20 times
- Count the occurrences of Head and Tail, respectively.

Random Number Generator

- `rand()`
  - The rand function returns a pseudorandom integer in the range 0 to RAND_MAX (32767)

```
// Print 5 random numbers.
for (int i = 0; i < 5; i++ )
    cout << rand() << endl;
```

Seed of `rand()`

- With the same seed, the program will get the same result at each execution.
- Use `srand()` and choose the current time as the seed.

```
#include <time.h>
srand((unsigned) time(NULL));
for (int i = 0; i < 5; i++ )
    cout << rand() << endl;
```

Sample Code (coin_tossing.cpp)

```
#include <iostream>
#include <time.h>
using std::cout;
using std::endl;

int main()
{
   int toss[2] = { 0 };        // 0 for Head; 1 for Tail
   int i;                      // coin tossing
   srand((unsigned) time(NULL));

   for (int k = 0; k < 20; k++ )
   {
        i = rand() % 2;
        toss[i] += 1;
   }

   cout << toss[0] << " Heads and " << toss[1] << " Tails.\n";
   return 0;
}
```

Decimal to Binary Conversions

- dec2bin.cpp

- Octal
  - dec2oct.cpp
- How about septenary (七進位)?
- Hexadecimal
  - dec2hex.cpp
  - dec2hex_v2.cpp

Binary to Decimal Conversion

```
// bin2dec.cpp
#include <iostream>
using std::endl;
using std::cout;
using std::cin;

int main()
{
    short i;
    char b[9];
    cout << "Please input a binary string -- ";
    cin >> b;
    for (i=0; i<8; i++)
        b[i] -= 48;       // '0' = 48
    for (i=0; i<=6; i++)
        b[i+1] = b[i] * 2 + b[i+1];

    cout << static_cast<short>(b[7]) << endl;
    return 0;
}
```

'0'	'0'	'0'	'0'	'0'	'1'	'0'	'1'
48	48	48	48	48	49	48	49
0	0	0	0	0	1	0	1

Exercise & Homework

- Exercise:
  - Improve bin2dec.cpp so that it can accept shorter binary digits such as 1101.
  - Bonus: Users can input binary strings such as "1101 1100" or simply "101". Invalid input such as "1201 1100" or "1101@1111" will be rejected with a warning message.
- Homework
  - oct2dec.cpp
  - hex2dec.cpp