Phys4051: C Lecture 4 & 5

- Review Switch Statement
- Bitwise Operator Examples
- Loops
- Arrays and Array Declaration
- Multidimensional Arrays

Branching Instructions: Switch Statement Syntax

```c
switch( variable ) {
    case CONSTANT :
        statements ;
        break;
}
```

Bitwise Operators: Bit Masking

Example: Write a function that returns 1 if the nth bit of a number is HI, 0 if it's LO.

Bitwise Operators: Bit Toggling

Example: Write a function that returns its argument except that the nth bit of its argument has been “toggled.”

Bitwise Operators: Combining Bits

Example: Combine 4 bits (D0 (LSB), D1, D2, D3 (MSB)) into a byte.

Version 1:
```c
X = 8*D3 + 4*D2 + 2*D1 + D0;
```

Version 2a:
```c
X = (D3<<3)+(D2<<2)+(D1<<1) + D0;
```

Version 2b:
```c
X = (((((D3<<1)+D2)<<1)+D1)<<1)+D0;
```

Repetitions and Loops (1) General

- Loop body
- Infinite loop
- Nested loops
Repetitions & Loops (2)
Types of loops in C:

※ Pre-execution (checking) loops:
  (may never execute)
  for loop
  while loop

※ Post-execution (checking) loop:
  (executes at least once)
  do-while loop

Repetitions & Loops (3)
For Loop Syntax

※ for( init. cond. ; logic cond. ; increment )
  {
    statement(s) ;
  }

※ Example:
  for( x = 34; x < 1104; x++)
    printf("looping "); //shortcut notation

Repetitions & Loops (4)
While Loop Syntax

※ while ( logic condition )
  {
    statement(s) ;
  }

※ Example:
  short x = 0;
  while( x > 5)
    x --; //shortcut notation

Repetitions & Loops (5)
Do-While Loop Syntax

※ do
  {
    statement(s) ;
  }
  while ( logic condition ) ;

※ Example:
  do
    x = x + 123; //shortcut notation
  while( x < 12345 );

Example 1: Bitwise Operators & Loop (1)

void char2bin(unsigned char num) 
{
  unsigned char i, mask = 0x80;
  for( i = 1; i < 9; i++)
    {
      if( num & mask)
        printf("1");
      else
        printf("0");
      mask = mask >> 1;
    }
}

Example 1: Bitwise Operators & Loop (2)

※ For the following statements:

  unsigned char val = 13;
  char2bin( val );

  what is the corresponding output ?
Example 2: Bitwise Operators & Loop (3)

val = 13; // 0000 1101

<table>
<thead>
<tr>
<th>i</th>
<th>mask</th>
<th>mask &amp; num</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1000 0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Midterm 96: Pseudo Random Number Generator

(7 Points) The picture on the right is from H&H p. 656. It shows a hardware implementation of a 4 bit pseudo random number generator using a shift register and an XOR gate.

For example, if Q3, Q2, Q1 and Q0 are all HI then after one clock cycle, i.e. after all the registers are shifted right, Q = 0111; the next clock cycles produces, Q = 0011, etc.

You will write a C function called MyRand( ) that implements its hardware function.

Pseudo Random Number Generator: Truth Table

<table>
<thead>
<tr>
<th>cycle</th>
<th>D_in</th>
<th>Q3</th>
<th>Q2</th>
<th>Q1</th>
<th>Q0</th>
<th>Q3Q2Q1Q0 (Decimal)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>

Pseudo Random Number Generator: Flowchart

short MyRand( short Q) {

Pseudo Random Number Generator: Solution

Same thing:

short MyRand( short Q )
{
    return((Q | ((Q&1) <<4) ^ ((Q&2)<<3) )) >>1 );
}

Arrays

Definition:

A collection of variables which are all of the same type

Examples:

short val[ 200 ]; //declaration
val[ 12 ] = 5; //assignment
Arrays: Declaration & Data Types

* Arrays have the same data types as variables, i.e., short, long, float etc.
* Arrays are similar to variables: they can either be declared global or local

Arrays: Number of Elements

* If an array y has \( n \) elements, then:
  - the "first" element is: \( y[0] \)
  - the last is: \( y[n-1] \).
* Example:
  - \( \text{short } y[4]; \)
  - contains the following elements:
  - Note: \( y[4] \) is \textbf{NOT} part of the array!!

Arrays: Memory Allocation

```
short x = 2;
short y[4];
y[x] = 12345;
```

(The memory addresses are arbitrarily chosen)

Example 4: Calculate Average

```
float average( void )
{
    short j, x[200];
    float accum = 0;
    for( j = 0; j < 200; j++)
        x[ j ] = rand(); //assign value
    for( j = 0; j < 200; j++)
        accum = accum + x[ j ];
    return (accum / 200);
}
```

Example 5: Bitwise Operators Combining Bits (1)

* Example: Combine 8 bits (D[0] (LSB), D[1]..., D[7] (MSB) into a byte:
* Previous Version Without Loop:
  \[ X = (D3<<3) + (D2<<2) + (D1<<1) + D0; \]
* Which is identical to:
  \[ X = D0; \]
  \[ X = X + (D1<<1); \]
  \[ X = X + (D2<<2); \]
  \[ X = X + (D3<<3); \]

Example 5: Bitwise Operators Combining Bits (2)

* New Version with Loop:
  \[ X = D[0]; \]
  \[ for( k = 1; k < 8; k++) \]
  \[ \{ \]
  \[ \} \]
Arrays: Initialization

* In global arrays, all elements are initialized to zero
* In local arrays, the elements are NOT initialized!

Arrays: Declaration & Initialization

* Various examples to declare and initialize arrays:
  
  short x[12] = {1, 3, 88};
  double w[ ] = {3.14, 127.0, 22};

  Strings:
  char a_name[10] = "Hello";
  char name[20] = {'H', 'e', 'l', 'l', 'o'};

Array Size

* Once declared, it's very difficult to change the size of an array!
* The number of array elements must be declared with a constant. It can NEVER be specified with a variable!
  Example:
  
  short x = 200;
  short y[x]; // WRONG!!!!!!!

Constants (Pre-Processors)

* Constant Declaration Example:
  
  #define ACONSTANT 300.0
  #define TRUE 1
  #define FALSE 0

  Note: NO semicolon at the end!
  * C-convention: the name of a constant is usually written in uppercase.

Example 6: Constant & Average Program

```
#define MAX 300
short v, x[ MAX ];
main(){
  float accum = 0, average;
  for(v = 0; v < MAX; v++)
    x[v] = rand();
  for(v = 0; v < MAX; v++)
    accum += x[v];
  average = accum / MAX;
}
```

Example 7:

```
#define MAX 200
short x, volt[MAX], temp;

/*... Various statements that assign values to array volt[] have been omitted*/
for(x = 0; x < MAX / 2; x++)
{
  temp = volt[x];
  volt[x] = volt[MAX - x -1];
  volt[MAX - 1 - x] = temp;
}
```


Example 7: (Program Segment)

```c
for( y = 0; y < MAX - 1; y++)
    { for(x = 0; x < MAX - 1 - y; x++)
        { if( volt[x] > volt[x + 1] )
            { temp = volt[x];
            volt[x] = volt[x + 1];
            volt[x + 1] = temp;
            }
        }
    }
```

Example 7: Content of Array “volt”

<table>
<thead>
<tr>
<th>Array Index</th>
<th>x=0</th>
<th>x=MAX-1</th>
<th>x=MAX-2</th>
<th>Finally</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Arrays: Memory and Size Considerations (1)

- Global arrays are always “static”
- Local arrays are always “dynamic”
- The “static” memory space is usually larger than the space allocated to “dynamic” variables and arrays!
- To make a local array “static”, declare it so explicitly: `static short x[200];`

Arrays: Memory and Size Considerations (2)

- Keep track of array size in Bytes!
  ```c
  char c[10000]; //10000 bytes
double d[10000]; // 80000 bytes
  ```
- Beware of the 64 kB boundary in some compilers!

Arrays: Memory and Size Considerations (3)

- When in doubt use the `sizeof` keyword:
  ```c
  char char;
double double;
  printf("%ld \n", sizeof( char ));
  printf("%ld \n", sizeof( double ));
  ```

  Output:
  ```
  1
  8
  ```

Multi-Dimensional Arrays

- 2D Array Example:
  ```c
  short daysofyear[2][13] = {
    { 0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 },
    { 0, 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 }}
  ```

- Usage:
  ```c
  days = daysofyear[leapyear][month]
  ```
Loops: Example 1

What do rabbits, the Golden Mean and the size of sheets of paper have in common?

Calculate the Golden Mean from the first 20 (and 21) terms of the Fibonacci Series.

How many terms are needed for Fibonacci Series till > 100000?