**Project 3: Bitwise Ops**

**Brandon Horner**

**February 22, 2013**

**CS 200**

**Purpose:**

The purpose of this project was to practice C++ by performing multiple bitwise operations on two inputted integers. The code had to be output in decimal, hexadecimal and binary form.

**Approach:**

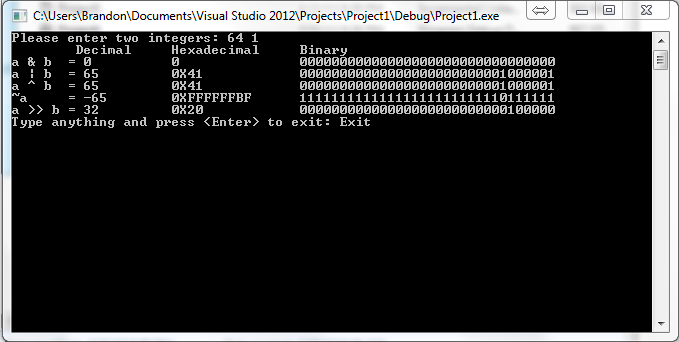
After figuring out how to build and look at my program I played around with simple input output and learned how the syntax of multiple C++ functions. I tried taking input and performed the multiple operations asked by the project description, simply displaying the results. Then I found the *printf()* function and decided I would try to format my output with it. Within the *printf()* C++ page I found that it was very simple to print out the hexadecimal form.

Finding the decimal to binary conversion proved challenging. And after much searching I found a user named Malcolm McLean on bytes.com that answered a question regarding converting decimal to binary. The code ended up leaving me with a lot of leading 0’s but no way of knowing how to get rid of them. After much fiddling I made the output presentable, finishing my project. I also added an a + b line of output to make sure my integers were behaving correctly in the first place.

**Conclusion:**

This project was more challenging than I thought to convert decimal to binary. Though I do think it was achievable even without knowing any C++. More or less the hardest part was figuring out the ins and outs of the *printf()* function, but I probably could have formatted this with *cout* as well. I understand the concept of creating a decimal to binary converter, but I do not have enough experience with C++ to be comfortable attempting to do so yet.

**Sample Output:**

****

**Source Code:**

#include <iostream>

#include <string>

using namespace std;

/\*

convert machine number to human-readable binary string.

Returns: pointer to static string overwritten with each call.

\*/

char \*itob(int x)

{

static char buff[sizeof(int) \* CHAR\_BIT + 1];

int i;

int j = sizeof(int) \* CHAR\_BIT - 1;

buff[j] = 0;

for(i=0;i<sizeof(int) \* CHAR\_BIT; i++)

{

if(x & (1 << i))

buff[j] = '1';

else

buff[j] = '0';

j--;

}

return buff;

}

/\*

Author: Brandon Horner

Takes two integers as input and applies various operation results in decimal, hexadecimal, and binary form.

\*/

int main(){

// initializing variables

cout << "Please enter two integers: ";

unsigned int a;

unsigned int b;

string exit;

// takes input unsigned int a and unsigned int b

cin >> a >> b;

unsigned int c;

printf("%15s %15s %10s \n", "Decimal", "Hexadecimal", "Binary");

// sets the unsigned variable c to the result of a & b

c = a & b;

// prints the formatted results

printf("%-8s %-10d %-#15X %-20s \n", "a & b =", c, c, itob(c));

c = a | b;

printf("%-8s %-10d %-#15X %-20s \n", "a | b =", c, c, itob(c));

c = a ^ b;

printf("%-8s %-10d %-#15X %-20s \n", "a ^ b =", c, c, itob(c));

c = ~a;

printf("%-8s %-10d %-#15X %-20s \n", "~a =", c, c, itob(c));

c = a >> b;

printf("%-8s %-10d %-#15X %-20s \n", "a >> b =", c, c, itob(c));

// code to keep console up and exit console

cout << "Type anything and press <Enter> to exit: ";

cin >> exit;

return 0;

}