/\*--------------------------------------------------------------\*/

/\* Sam Lee \*/

/\* Lab8 Spring 2018 \*/

#include <stdio.h>

#include <stdlib.h>

/\* function prototypes \*/

void bitprint (unsigned num);

int rotate\_left(unsigned num, int n);

/\*-----------------------------------------\*/

int main (void)

{

int left\_count;

unsigned num; /\* the starting number \*/

unsigned shifted\_num;

printf("\nSam Lee. Lab 8.\n");

do

{

/\* read a unsigned integer \*/

printf("\nEnter an unsigned integer value (0 to stop): ");

scanf("%u", &num);

if (num != 0)

{

printf("\nEnter an integer value for the left shift: ");

scanf("%d", &left\_count);

printf("\nOriginal is %u \n", num);

bitprint(num);

shifted\_num = rotate\_left(num, left\_count);

bitprint(shifted\_num);

printf("Shifted it is %u \n", shifted\_num);

}

} while (num != 0);

printf("\n\n");

return EXIT\_SUCCESS;

}

/\*--------------------------------------------------------------\*/

void bitprint (unsigned num)

{

unsigned mask;

int bit, count, nbits;

/\* determine the word size in bits and set the initial mask \*/

nbits = 8 \* sizeof(unsigned); /\* finds number of bytes in an unsigned w

number and changes it to bits \*/

mask = 0x1 << (nbits - 1); /\* place 1 in left most position

starting place for the mask \*/

for(count = 1; count <= nbits; count++)

{

bit = (num & mask) ? 1: 0; /\* set display bit on or off \*/

printf("%x", bit); /\* print display bit \*/

if(count %4 == 0)

printf(" "); /\* blank space after every 4th digit \*/

mask >>= 1; /\* shift mask 1 position to the right \*/

}

printf("\n");

return;

}

/\*--------------------------------------------------------------\*/

int rotate\_left(unsigned num, int n)

{

int count, bit, nbits;

unsigned mask;

nbits = 8 \* sizeof(unsigned); /\* finds number of bytes in an int

unsigned number and changes it to bits \*/

mask = 0x1 << (nbits - 1); /\* place 1 in left most position

starting place for the mask \*/

// put the loop here and then the return

for (count =1; count <= n; count++)

{

bit = (num & mask) ? 1: 0;

num = num << 1;

num = num + bit;

}

return num;

}

/\*--------------------------------------------------------------\*/