CIS37A-Lab5

Due: July-22-2017, 11:59PM

Name: Subal Pant

**Problem 1:** Write a program that reverses the order of the bits in an unsigned int value. The program should input the value from the user and call function reverseBits to print the bits in reverse order. Print the value in bits both before and after the bits are reversed to confirm that the bits are reversed properly.  
#include <stdio.h>

#include <conio.h>

void displayBits(unsigned int a)

{

unsigned int displayMask = 1 << 31;

printf("%u (Forward Sequence) = ", a);

for (size\_t i = 1; i <= 32; i++)

{

putchar(a & displayMask ? '1' : '0');

a <<= 1;

if (i % 8 == 0)

{

putchar(' ');

}

}

}

void reverseBits(unsigned int a)

{

unsigned int displayMask = 1;

printf("%u (Reverse Sequence) = ", a);

for (size\_t i = 1; i <= 32; i++)

{

putchar(a & displayMask ? '1' : '0');

a >>= 1;

if (i % 8 == 0)

{

putchar(' ');

}

}

}

int main(void)

{

unsigned int x;

for (size\_t i = 0; i < 1; i++)

{

printf("Enter an Non-Negative Integer Value: ");

scanf("%d", &x);

}

displayBits(x);

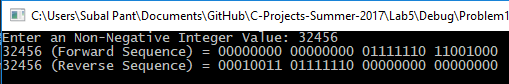
printf("\n");

reverseBits(x);

printf("\n");

getch();

}



**Problem 2:** Write a program to read an integer. Pass this integer to a function called f. The function must find the number of 1s in this integer. Then it should check if the number of 1s is odd or even. If the number of 1s is even, the function must return 1. If the number of 1s is even, the function must return 0. After you call this function in main, display the bits of this number in groups of 8 and display the bit value that the function returns.

#include <stdio.h>

#include <conio.h>

unsigned int f(unsigned int a)

{

unsigned int displayMask = 1 << 31;

unsigned int count = 0;

for (size\_t i = 1; i <= 32; i++)

{

if (a & displayMask > 0)

{

count++;

}

a <<= 1;

}

if (count % 2 == 0)

{

return 1;

}

else

{

return 0;

}

}

void displayBits(unsigned int a)

{

unsigned int displayMask = 1 << 31;

printf("%u = ", a);

for (size\_t i = 1; i <= 32; i++)

{

putchar(a & displayMask ? '1' : '0');

a <<= 1;

if (i % 8 == 0)

{

putchar(' ');

}

}

}

int main(void)

{

unsigned int x;

for (size\_t i = 0; i < 1; i++)

{

printf("Enter an Non-Negative Integer Value: ");

scanf("%d", &x);

}

printf("\n");

displayBits(x);

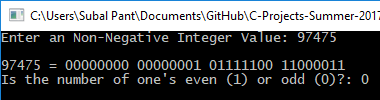
printf("\n");

unsigned int even = f(x);

printf("Is the number of one's even (1) or odd (0)?: %u\n", even);

getch();

}



**Problem 3:** Write a program to create an integer variable dynamically. Place the address of this variable into an integer pointer. Read a value into this variable. Pass the address of this variable to a void function that multiplies the content of it by 2. Display the value of this variable after the function call in main. Delete the variable and display the message that the dynamic variable is deleted. Explain why we delete dynamic variable after we are done with them.

#include <stdio.h>

#include <conio.h>

void mult(int \*pMultVar)

{

\*pMultVar = \*pMultVar \* 2;

}

int main(void)

{

int \*pVal = malloc(sizeof(int));

\*pVal = 5;

printf("Value Before Mult = %d\n", \*pVal);

mult(pVal);

printf("Value After Mult = %d\n", \*pVal);

free(pVal);

pVal = NULL;

if (pVal == NULL)

{

printf("pVal dynamic variable was deleted.\n");

}

getch();

}

/\*

We delete dynamic variables after we are done with them because if the variable is not deleted, it will cause a memory leak.

The Windows OS will reclaim the memory after the program quits.

\*/

