**Talen Phillips**

**EE 107-01**

**Homework 7**

**Due 8 Apr 13**

1) Write a program containing three functions including the main() function. Name the other two functions fn\_even and fn\_odd. The task of fn\_even() is to print the word "EVEN" to the screen. The task of fn\_odd() is to print the word "ODD" to the screen. The main() function requests a positive integer from the user and uses a switch statement to call the appropriate function (i.e. fn\_even() or fn\_odd()) based on the number entered.

/\* Talen Phillips

\* EE107-01

\* 08APR2013

\* "Even or Odd"

\*/

#include <stdio.h>

/\*declaring each additional function (since they come after the function in which they're called)\*/

int fn\_even(int i);

int fn\_odd(int i);

int main (void){

int i;

printf("Enter a number: ");

scanf("%d",&i);

if ((i%2)==0) fn\_even(i);

else fn\_odd(i);

return 0;

}

/\*defining each additional function\*/

int fn\_even(int i){

printf("%i is even.\n",i);

return 0;

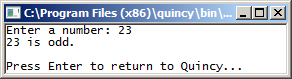
}

int fn\_odd(int i){

printf("%i is odd.\n",i);

return 0;

}



2) Write a program which requests the user to enter an integer with magnitude less than 1,000. Then store the number in a structure such that it contains a variable for the ones digit, tens digit and hundreds digit as well as a variable for the sign of the entered number. For example, assume your variable with the structure type described is called 'A', then if 766 is entered by the user, 'A' would contain the following:

A.sign = 0 (\*Zero for positive, one for negative.)

A.ones = 6

A.tens = 6

A.hundreds = 7

/\* Talen Phillips

\* EE107-01

\* 08APR2013

\* "Structures"

\*/

#include <stdio.h>

int main (void){

int i;

/\*initialize structure\*/

struct A{

int sign;

int hundreds;

int tens;

int ones;

};

struct A number;

printf("Enter a number from -999 to 999: ");

scanf("%d",&i);

/\*absolute value function, which also adjusts the sign\*/

if (i<0){

i=-i;

number.sign = 1;

}

else number.sign = 0;

/\*determine the value of each place\*/

number.hundreds = (i/100)%10;

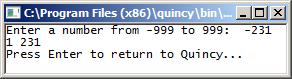
number.tens = (i/10)%10;

number.ones = i%10;

/\* print the resultant values (included for bug detection).\*/

printf("%i %i%i%i",number.sign,number.hundreds,number.tens,number.ones);

}



3) Write a program to request a positive integer less than 256 and print the binary equivalent to the screen. Do not print leading zeros. For example if the user enters '7', the program should print

111 to the screen. Your program should use bit shift operators.

/\* Talen Phillips

\* EE107-01

\* 08APR2013

\* "Decimal to Binary Converter"

\*/

#include <stdio.h>

int main (void)

{

unsigned short int dec\_num, i;

/\*shift 1 left by 7 (into the most significant place)\*/

unsigned short int bin\_num = 1 << (sizeof(unsigned short int)\*4 - 1);

printf("Enter a decimal number between 1 and 255: ");

scanf("%d",&dec\_num);

printf("The binary version of your number is:\t ");

/\*for each place, shift the 1 from the declaration left to the current place, and compare it to the user input. Print the truth of the logic statement in the printf function.\*/

for(i=0; i<sizeof(short int)\*4; i++){

printf("%hu",(dec\_num & (bin\_num >> i))!=0);

}

return 0;

}

