#include <stdio.h>

#include <math.h>

#include <stdlib.h>

// Project Includes

#include <cmsc257-s17-assign1-support.h>

//

// Functions

////////////////////////////////////////////////////////////////////////////////

//

// Function : main

// Description : The main function for the CMSC257 assignment #1

//

// Inputs : argc - the number of command line parameters

// argv - the parameters

// Outputs : 0 if successful test, -1 if failure

int main(int argc, char \*argv[]) {

// Local variables

// NOTE: this is where you will want to add some new variables

int numOfinput = 20;

float f\_array[numOfinput];

int tmp, i, m\_array[25];

int i\_array[numOfinput];

int mask;

// Step #1 - read in the float numbers to process

for (i=0; i<numOfinput; i++) {

scanf("%f", &f\_array[i]);

// printf("%f\n",f\_array[i] );

}

// float\_display\_array(f\_array);

// Step #2 - convert the floating point values to integers

for(int i = 0; i<numOfinput; i++){

tmp= (int) f\_array[i];

//Absolute value bitwise

mask = tmp>>31 ;

// mask = mask ^ tmp;

tmp = (mask ^ tmp) - mask;

tmp = tmp & 15 ; // temp mod 16

i\_array[i] = tmp;

}

// Step #3 - print out the integer and floating point arrays

printf("===== print the orginal input ======\n");

float\_display\_array(f\_array, numOfinput);

printf("===== print the integer array ======\n");

integer\_display\_array(i\_array, numOfinput);

// Step #4 - print out the number of '1's in each integer

printf("===== print out the number of '1's in each integer ======\n");

for (i=0 ; i<numOfinput ; i++){

printf("%d : %d \n",i\_array[i],countBits(i\_array[i]) );

}

// Step #5 - Sort the integer array using QuickSort and print the sorted array

integerQuickSort(i\_array,0,numOfinput-1); // sort the integer array

printf("===== print out the sorted integer array ======\n");

integer\_display\_array(i\_array, numOfinput);

// Step #6 - figure out number of evens in each array (float and sorted integer arrays)

printf("===== figure out number of evens in each array ======\n");

printf("even integers %d : even floats %d \n",float\_even(f\_array,numOfinput),integer\_even (i\_array,numOfinput) );

// Step #7 - print the most frequent values from the sorted integer array

printf("===== print the most frequent values from the sorted integer array ======\n");

printf("Most Valuse:\n");

most\_values(i\_array,numOfinput,16);

// Step #8 - print the integers corresponding to the reverse bit string from integer array

printf("===== print the integers corresponding to the reverse bit string from integer array ======\n");

int size = 17; // size of the string

char binary[size]; // define the buffer string

unsigned short reversed; //placeholder for reversed integer

for(i=0;i<numOfinput;i++){

reversed = reverseBits((unsigned short) i\_array[i] ); //reverse the integer

binaryString(binary,17,(unsigned short) i\_array[i] );

printf("orginal: %d == %s ",i\_array[i],binary);

binaryString(binary,17, reversed);

printf("rervese: %d == %s\n",reversed ,binary);

}

// Return successfully

return(0);

}