// CIS2541

// Hong Zhang

// Program: use loop to generate a table of

// decimal numbers, as well as the binary,

// octal and hexadecimal equivalents of

// decimal numbers between 1 and 256

#include "stdafx.h"

#include <iostream>

#include <iomanip>

#include <cmath>

using namespace std;

int \_tmain(int argc, \_TCHAR\* argv[])

{

int min, max, number;

// for binary conversion

int remainder, temp;

int corrected\_binary = 0;

int reversed\_binary = 0;

int count=0;

// for octal conversion

int corrected\_octal = 0;

int reversed\_octal = 0;

int counter = 0;

//for hexadecimal conversion

int i = 0;

int j;

char hex[5];

cout << "Enter the low number: \n";

cin >> min;

// Validate the input

while (min < 1 || min > 256)

{

cout << "Invalid Entry!\n"

<< "Entry a number between 1 and 256: ";

cin >> min;

}

cout << "Enter the high number: \n";

cin >> max;

// Validate the input

while (max < 1 || max > 256 || max < min)

{

cout << "Invalid Entry!\n"

<< "Entry a number between 1 and 256: ";

cin >> max;

}

cout << left;

cout << setw(8)<<"Decimal"<<setw(12)<<"Binary"<<setw(12)<<"Octal"<<setw(12)<<"Hexadecimal"<<endl;

for (number = min; number <= max; number++)

{

//intialize varibles

int corrected\_binary = 0;

int reversed\_binary = 0;

int count=0;

int corrected\_octal = 0;

int reversed\_octal = 0;

int counter = 0;

int i = 0;

// binary conversion

temp = number;

while (temp)

{

remainder = temp % 2;

temp = temp / 2;

reversed\_binary = reversed\_binary \*10 + remainder;

count++;

}

while (count)

{

remainder = reversed\_binary % 10;

corrected\_binary = corrected\_binary \*10 +remainder;

reversed\_binary = reversed\_binary / 10;

count--;

}

// octal conversion

temp = number;

for (counter=0; counter<4 ;counter++)

{

remainder = temp % 8;

temp = temp/8;

reversed\_octal = reversed\_octal \*10 +remainder;

}

for (counter=3; counter >= 0 ;counter--)

{

remainder = reversed\_octal % 10;

corrected\_octal = corrected\_octal\*10 +remainder;

reversed\_octal = reversed\_octal / 10;

}

//hexadecimal conversion

temp = number;

do

{

remainder=temp%16;

if (remainder>9) //To print A instead of 10, we have to add 55 to remainder

remainder=remainder+55;

else

remainder=remainder+48; //To print 0,1,2 etc as a char, we have to add 48 to remainder

hex[i++]=remainder;

temp=temp/16;

}

while (temp != 0);

cout << left;

cout << setw(8)<<number<<setw(12)<<corrected\_binary<<setw(12)<<corrected\_octal;

for (j=i-1;j>=0;j--) //To print our array in reverse!

cout<< hex[j];

cout <<"\n";

}

cout <<"\n\n";

cout << "\*\*\*\*Resetting low and high\*\*\*\*\n";

cout << "Enter the low number: \n";

cin >> min;

// Validate the input

while (min < 1 || min > 256)

{

cout << "Invalid Entry!\n"

<< "Entry a number between 1 and 256: ";

cin >> min;

}

cout << "Enter the high number: \n";

cin >> max;

// Validate the input

while (max < 1 || max > 256 || max < min)

{

cout << "Invalid Entry!\n"

<< "Entry a number between 1 and 256: ";

cin >> max;

}

cout << left;

cout << setw(8)<<"Decimal"<<setw(12)<<"Binary"<<setw(12)<<"Octal"<<setw(12)<<"Hexadecimal"<<endl;

for (number = min; number <= max; number++)

{

//intialize varibles

int corrected\_binary = 0;

int reversed\_binary = 0;

int count=0;

int corrected\_octal = 0;

int reversed\_octal = 0;

int counter = 0;

int i = 0;

// binary conversion

temp = number;

while (temp)

{

remainder = temp % 2;

temp = temp / 2;

reversed\_binary = reversed\_binary \*10 + remainder;

count++;

}

while (count)

{

remainder = reversed\_binary % 10;

corrected\_binary = corrected\_binary \*10 +remainder;

reversed\_binary = reversed\_binary / 10;

count--;

}

// octal conversion

temp = number;

for (counter=0; counter<4 ;counter++)

{

remainder = temp % 8;

temp = temp/8;

reversed\_octal = reversed\_octal \*10 +remainder;

}

for (counter=3; counter >= 0 ;counter--)

{

remainder = reversed\_octal % 10;

corrected\_octal = corrected\_octal\*10 +remainder;

reversed\_octal = reversed\_octal / 10;

}

//hexadecimal conversion

temp = number;

do

{

remainder=temp%16;

if (remainder>9) //To print A instead of 10, we have to add 55 to remainder

remainder=remainder+55;

else

remainder=remainder+48; //To print 0,1,2 etc as a char, we have to add 48 to remainder

hex[i++]=remainder;

temp=temp/16;

}

while (temp != 0);

cout << left;

cout << setw(8)<<number<<setw(12)<<corrected\_binary<<setw(12)<<corrected\_octal;

for (j=i-1;j>=0;j--) //To print our array in reverse!

cout<< hex[j];

cout <<"\n";

}

cout <<"\n\n";

return 0;

}