**Renee Thomas**

**CIS170C\_Lab04**

**2/2/14**

**Lab # CIS CIS170C-A4**

// ---------------------------------------------------------------

// Programming Assignment: LAB4

// Developer: Renee Thomas

// Date Written: 2/2/14

// Purpose: Functions - The phone number game

// ---------------------------------------------------------------

# include <iostream>

# include <iomanip>

# include <string>

# include <cctype>

using namespace std;

int readDials(char &d1, char &d2, char &d3, char &d4, char &d5, char &d6, char &d7, char &d8); // initialize readDials function (function prototype) step 1

int toDigit(char &d); // initialize toDigit function

void acknowledgeCall(char d1, char d2, char d3, char d4, char d5, char d6, char d7, char d8); // initialize acknowledgeCall funnction

// main function

int main()

{

// Welcome and Intro to program/ game

char d1, d2, d3, d4, d5, d6, d7, d8;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"welcome to the phone number check game."<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl<<endl;

while(true)

{

// create variable to hold return of readDials function (call readDials) - step 3

int code = readDials(d1, d2, d3, d4, d5, d6, d7, d8);

// check for invalid codes to output error messages

if (code== -5)

break;

if (code == -1)

cout<<"ERROR - An invalid character was entered."<< endl;

if (code == -2)

cout<<"ERROR - Phone number cannot begin with 0."<< endl;

if (code == -3)

cout<<"ERROR - Phone number cannot begin with 555."<< endl;

if (code == -4)

cout<<"ERROR - Hyphen is not in the correct position."<< endl;

if (code == 0)

acknowledgeCall(d1, d2, d3, d4, d5, d6, d7, d8);

}

// after user presses Q, output message to thank user for playing the game (I just lost the game!!! ugh!!)

cout<<"\n\nThank you for playing the phone number check game."<< endl;

cin.ignore(2);

return 0;

}

// create readDials function - setp 2

int readDials(char &d1, char &d2, char &d3, char &d4, char &d5, char &d6, char &d7, char &d8)

{

cout<< "\nEnter a phone number (Press \"Q\" to quit): ";

cin>> d1;

// check to see if first digit is a Q if so break

d1 = toupper(d1);

if (d1 == 'Q')

return -5;

cin>>d2>>d3>>d4>>d5>>d6>>d7>>d8;

// call the toDigit function for each character entered

toDigit(d1);

toDigit(d2);

toDigit(d3);

toDigit(d5);

toDigit(d6);

toDigit(d7);

toDigit(d8);

// if any digits are symbols return -1

if (toDigit(d1) == -1 ||

toDigit(d2) == -1 ||

toDigit(d3) == -1 ||

toDigit(d5) == -1 ||

toDigit(d6) == -1 ||

toDigit(d7) == -1 ||

toDigit(d8) == -1)

return -1;

// if the first character entered is 0 return -2

if (d1 == '0')

return -2;

// if the first three characters entered equal 5, return -3

if(d1=='5' && d2 =='5' && d3=='5')

return -3;

// if the hyphen is in the wrong place, return -4

if(d4 != '-')

return -4;

// if there are no errors, return 0

return 0;

}

// create the toDigit function to check for symbols and to convert letters to numbers

int toDigit(char &d)

{

d = toupper(d);

switch(d)

{

// use the table in the lab instructions to create a switch statment to convert all letters to numbers

case '1': d='1'; break;

case '0': d='0'; break;

case 'A': case 'B': case 'C': case '2': d='2'; break;

case 'D': case 'E': case 'F': case '3': d='3'; break;

case 'G': case 'H': case 'I': case '4': d='4'; break;

case 'J': case 'K': case 'L': case '5': d='5'; break;

case 'M': case 'N': case 'O': case '6': d='6'; break;

case 'P': case 'Q': case 'R': case 'S':case '7': d='7'; break;

case 'T': case 'U': case 'V': case '8': d='8'; break;

case 'W': case 'X': case 'Y': case 'Z': case '9': d='9'; break;

// if user enters anything other than a number or letter, return -1

default: return -1;

}

return 0;

}

// create the function to output good number

void acknowledgeCall(char d1, char d2, char d3, char d4, char d5, char d6, char d7, char d8)

{

cout<<"The phone number you dialed: "<< d1<<d2<<d3<<d4<<d5<<d6<<d7<<d8<< endl;

}

