COMP-165-0837

Merlin Newcomb

Loops Assignment

Problem #1

Code:

/\*

\* This program continuously prompts the user for a telephone number in letters, and translates it to digits accordingly.

\* If more than 7 letters are entered, only the first 7 letters are used.

\* If less than 7 letters are entered, a malformed phone number is returned.

\* Spaces in phone-number input are stripped, and all user input is converted to uppercase.

\*/

#include <iostream>

#include <string>

#include <cctype>

#include <algorithm>

void readToUpper(std::string &userInput); // std::getline(std::cin, userInput), converting to uppercase. One line at a time.

int main(void)

{

std::string userInput;

std::string resultNumber;

while (true)

{

resultNumber.clear();

std::cout << "Enter Y/y to convert a telephone number from letters to digits." << std::endl;

std::cout << "Use any other letter to terminate the program." << std::endl;

readToUpper(userInput);

if (userInput != "Y")

{

break;

}

std::cout << "Enter a telephone number using letters: " << std::endl;

readToUpper(userInput);

userInput.erase(std::remove(userInput.begin(), userInput.end(), ' '), userInput.end()); // Strips space characters

std::cout << std::endl;

for (long unsigned int i = 0; i < userInput.size(); i++) // Make i "long unsigned int" so compiler stops complaining

{

if (i >= 7) // Break if seven letters are already converted

{

break;

}

if (i == 3) // Add hyphen when appropriate

{

resultNumber.append("-");

}

switch (userInput.at(i)) // This... would be much more tedious with if/else branches.

{

case 'A':

case 'B':

case 'C':

resultNumber.append("2");

break;

case 'D':

case 'E':

case 'F':

resultNumber.append("3");

break;

case 'G':

case 'H':

case 'I':

resultNumber.append("4");

break;

case 'J':

case 'K':

case 'L':

resultNumber.append("5");

break;

case 'M':

case 'N':

case 'O':

resultNumber.append("6");

break;

case 'P':

case 'Q':

case 'R':

case 'S':

resultNumber.append("7");

break;

case 'T':

case 'U':

case 'V':

resultNumber.append("8");

break;

case 'W':

case 'X':

case 'Y':

case 'Z':

resultNumber.append("9");

break;

default:

resultNumber.append("?");

break;

}

}

std::cout << "The corresponding telephone number is:" << std::endl;

std::cout << resultNumber << std::endl;

std::cout << std::endl;

}

return 0;

}

void readToUpper(std::string &userInput)

{

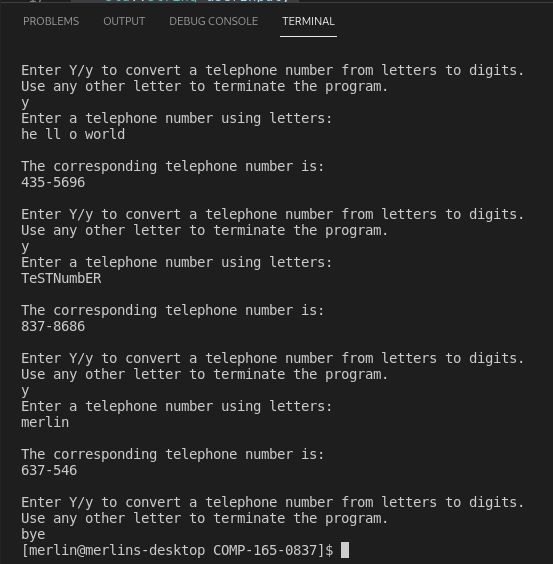
std::getline(std::cin, userInput);

std::transform(userInput.begin(), userInput.end(), userInput.begin(), toupper); // Converts to uppercase

return;

}

Output:



Problem #2

Code:

/\*

\* Aggregates and displays rainfall data provided by user.

\*

\* Input Validation:

\* User cannot enter less than 1 year.

\* User cannot enter negative rainfall per month.

\*/

#define forever for (;;) // :)

#include <iostream>

int main(void)

{

int userYears = 0;

int totalMonths = 0;

double userRainfall = 0.0;

double totalRainfall = 0.0;

double avgRainfall = 0.0;

forever // :))

{

std::cout << "Enter number of years: " << std::flush;

std::cin >> userYears;

if (userYears > 0)

{

break;

}

std::cout << "Invalid input. Number of years must be 1 or greater." << std::endl;

}

for (int year = 0; year < userYears; year++)

{

for (int month = 0; month < 12; month++)

{

forever // Now I'm just being consistent!

{

std::cout << "Enter rainfall (inches) for month " << (month + 1) << ": " << std::flush;

std::cin >> userRainfall;

if (userRainfall >= 0.0)

{

totalRainfall += userRainfall;

break;

}

std::cout << "Invalid input. Rainfall for month most be non-negative." << std::endl;

}

}

}

totalMonths = userYears \* 12;

avgRainfall = totalRainfall / totalMonths;

std::cout << "Total months: " << totalMonths << std::endl;

std::cout << "Total rainfall: " << totalRainfall << " inches" << std::endl;

std::cout << "Avg. rainfall per month: " << avgRainfall << " inches" << std::endl;

return 0;

}

Output:

