Lab #06

***Alexander Fan***

***100229106***

CPSC 1150 - 003

Instructor: H. Darbandi

Lab Title: Cipher Encoding and Decoding

Date Completed: Jun 11, 2020

Department: CSIS

Program: Cipher Encoding and Decoding

File Name: Lab06.java

Purpose: Cipher Encoding and Decoding

Technical Information:

(You should fill the following information based on compiler and computer you are using).

Compiler: Java SDK version 14

Computer: AMD Ryzen 5 2600 3.40 GHz, 16 GB ram, 64-bit processor, Java SDK 14

Operating System: Windows 10

Language: Java

Program Logic (Pseudocode)

Algorithm:

**START**

* 1. final static int asciiNum = 10;
  2. final static int asciiLetter = 26

**Main Body**

1. Char [] message
2. Int cipherKey = 0;
3. String ans = “”;
4. Accept user inputted message and cipher Key
5. For each character in char[] message //length of char array message
   1. ans += cipher(msg[i], key);

**String AsciiToBinary(int n)**

1. String binary = “” //initialize as blank string
2. Int binaryCount = 0, temp //binary numbers are 8 digits, so this allows for each binary representation to be in the form of xxxxxxxx
3. While binaryCount < 8
   1. Int temp = n % 2
   2. Binary = temp + binary
   3. n /= 2
   4. binaryCount += 1
4. Return Binary

**String cipher(char c, int key)**

1. String cipher = “”;
2. if((int) **c** >= 48 && (int) **c** <= 57) //if in the range of 48-57 for ascii numbers
   1. return cipher = cipherNum(c, key);
3. else if ((int) c >= 65 && (int) c <= 90) //if in the range of 65 – 90 for ascii Uppercase Letters
   1. return cipher = cipherUpper(c, key);
4. else if ((int) c >= 97 && (int) c <= 122) //if in the range of 65 – 90 for ascii Uppercase Letters
   1. return cipher = cipherLower(c, key);
5. else
   1. return cipher = asciiToBinary((int) c);

**String cipherNum(char c, int key)**

1. Int ascii = (int) c;
2. String cipherNum = “”;
3. if(ascii + key > 57)
   1. ascii = ascii – (asciiNum – key)
   2. return cipherNum = asciiToBinary(ascii);
4. else
   1. ascii = ascii + key
   2. return cipherNum = asciiToBinary(ascii);

**String cipherUpper(char c, int key)**

1. Int ascii = (int) c;
2. String cipherUpper = “”;
3. if(ascii + key > 90)
   1. ascii = ascii – (asciiLetter – key)
   2. return cipherUpper = asciiToBinary(ascii);
4. else
   1. ascii = ascii + key
   2. return cipherUpper = asciiToBinary(ascii);

**String cipherLower(char c, int key)**

1. Int ascii = (int) c;
2. String cipherLower = “”;
3. if(ascii + key > 122)
   1. ascii = ascii – (asciiLetter – key)
   2. return cipherLower = asciiToBinary(ascii);
4. else
   1. ascii = ascii + key
   2. return cipherLower = asciiToBinary(ascii);

**END**

Generate your test cases based on the specifications in your lab assignment. Follow following format for each test case: (Refer to external document of your previous lab)

*purpose*

*input*

*output*

*expected value*

*passed or failed*

**Test Cases:**

**Test Case 1**

*Input:* CPSC1150 checking in!

Cipher Key: 4

*Expected Output:*

(hidden ciphered text) GTWG5594 gligomrk mr!

01000111 01010100 01010111 01000111 00110101 00110101 00111001 00110100 00100000 01000111 01001100 01001001 01000111 01001111 01001101 01010010 01001011 00100000 01001101 01010010 00100001

*Output: (spaced out for readability)*

01000111 01010100 01010111 01000111 00110101 00110101 00111001 00110100 00100000 01100111 01101100 01101001 01100111 01101111 01101101 01110010 01101011 00100000 01101101 01110010 00100001

Passed

**Test Case 2**

*Input:* This is a test!

Cipher Key: 4

*Expected Output:*

(hidden ciphered text) Xlmw mw e xiwx!

01011000 01101100 01101101 01110111 00100000 01101101 01110111 00100000 01100101 00100000 01111000 01101001 01110111 01111000 00100001

*Output:* (spaced for readability)

01011000 01101100 01101101 01110111 00100000 01101101 01110111 00100000 01100101 00100000 01111000 01101001 01110111 01111000 00100001

Passed

**Test Case 3**

*Input:* Trying out !!! different messages and ciphers?

Cipher Key: 3

*Expected Output:*

(hidden ciphered text) Wu4lqj rxw !!! gliihuhqw phvvdjhv dqg flskhuv?

01010111 01110101 00110100 01101100 01110001 01101010 00100000 01110010 01111000 01110111 00100000 00100001 00100001 00100001 00100000 01100111 01101100 01101001 01101001 01101000 01110101 01101000 01110001 01110111 00100000 01110000 01101000 01110110 01110110 01100100 01101010 01101000 01110110 00100000 01100100 01110001 01100111 00100000 01100110 01101100 01110011 01101011 01101000 01110101 01110110 00111111

*Output:*

01010111 01110101 01110010 01101100 01110001 01101010 00100000 01110010 01111000 01110111 00100000 00100001 00100001 00100001 00100000 01100111 01101100 01101001 01101001 01101000 01110101 01101000 01110001 01110111 00100000 01110000 01101000 01110110 01110110 01100100 01101010 01101000 01110110 00100000 01100100 01110001 01100111 00100000 01100110 01101100 01110011 01101011 01101000 01110101 01110110 00111111

Passed

**Test Case 4**

*Input:* This is great.

Cipher Key: 3

*Expected Output:*

(hidden ciphered text) Wklv lv juhdw.

01010111 01101011 01101100 01110110 00100000 01101100 01110110 00100000 01101010 01110101 01101000 01100100 01110111 00101110

*Output:*

01010111 01101011 01101100 01110110 00100000 01101100 01110110 00100000 01101010 01110101 01101000 01100100 01110111 00101110

Passed