

Abstract:

I wrote a code that allow the user to manipulate an input in one of 3 different ways. The ways are as follows: convert a sting that has been input in to a different string via manipulation by a caesar cipher, convert a sting into the corresponding ascii values and out put them in a list, and lastly turn any input number into its same value in binary. The code works very well and a code that does exactly what mine does with the same capabilities does not exist as far as I am concerned.

Intro:

Sometimes whether it be for professional or private reasons I have to send a message across a public system such as facebook messenger. Although it is secure due to the recent events that have taken place I feel like this is ever more prevalent. When sending sensitive info across these channels it can be a good idea to encrypt the info so that it is not read by people you don't want reading it. Along with encryption sometimes when it comes to coding you need the ascii value of a character and don't

know what it is. The conversion of decimal numbers to binary ones is also a very useful tool that can be used in multiple different scenarios.

Detailed system description:

For the caesar code the user inputs what message they would like to have encoded or decoded. The system then asks them what type of encoding they would like to use. Then they are prompted to enter a number between 0 and 25 to encode the message. The code then finds the letter of the message that is stored in an array and replaces it with the number that comes however many places after it working in round robin style. The encoded message is then printed out to the user.

For the ascii code the user inputs what string they want the ascii characters for. The code takes the characters and runs them against ascii values and prints out the one that matches what was requested.

The decimal to binary code takes an input of a positive integer and divides it until there is only a remainder of 1 or zero. That value is then stored in an array. That array is then converted to a string and output for the user to view.

The code caller that calls all of the other codes into one code prompts the user to input a number between 1 and three while showing what these numbers correspond with. That code that corresponds with the chosen number is then run.

Caesar2
alphaLength: ints asciiShift: char chphershift:int
+cipher(s string,k int): string +cipher(c char, k int): char

toascii
Ascii: char
+main(String[]):void

decimaltobinary
Remainder: int Number: int
+printBinaryform(int): void

caller
Choice: int Key: int Password: string Number: int
+main(String[]):void +printBinaryform:(int)

Requirements:

Many people across the world value privacy. This has become very hard to come across in this digital era and the only way to have true secrecy is to encode messages so that only you know how to decode it as well as the person receiving the message. The program I am writing allows for the users to safely encrypt and decrypt messages easily without having to spend time decoding by hand the old fashioned way.

Literature survey:

Many other systems like this exist but many of them are either single encoding systems that can only process one type of cipher or they can only encode or decode but not both. The system is far from complete and will need much more work as making encoders and decoders from scratch is difficult and prone to messing up. All the code will be accessed via a centralized class that will use the other codes.

User manual:

The code runs from the compiler in eclipse or whatever compiler you desire to use. To use the encoder first run the class. You will be asked to enter a message. Once the message has been entered click over to the area asking for a number between 0 and 25. After entering that number hit enter and your message will be encoded based

upon what number you selected. That number is important because it will also be used to decode the message when it needs to be.

Conclusion:

In its current state the system can encode and decode a caesar cipher with 100% accuracy. The ascii code now works and is able to translate anything put into the scanner and output a list of the related ascii values in descending order. It can also translate any decimal number inputted into its binary form with 100% accuracy for most numbers. Each one of these individual codes is brought together through the actions of the caller function. It asks the user which one they would like to use and then it runs that code and then it outputs the result all in one code.

References/Bibliography:

Lyons, james. "Crypto." *Practical Cryptography*, Practical Cryptography, 2012,
practicalcryptography.com/ciphers/.