CIST 1400, Introduction to Computer Programming
Programming Assignment

Program 4 : 25 points

Overview of Assignment

This program will extend what you have done before for the ROT13 encryption program earlier this semester.

Instead of converting a single letter, you will encrypt an entire sentence that is entered by the user.

So, for instance, if you entered `MEET ME IN THE NORTHERN COURTYARD`, your program would return `ZRRGZ RVAGU RABEG UREAP BHEGL NEQ`, with letters in the ciphertext being grouped by fives.

Using the `String` Data Type

You may have encountered this before in lab or elsewhere, but we can now officially introduce the `String` data type, as you'll need this in order to store an entire sentence of text from the user. `String` will be discussed in greater depth later this semester.

A `String` variable is declared like other variables, but using it for input and to get at particular characters is a little different.

To read an entire sentence from the keyboard into a `String` variable, you would use the `ScannerName.nextLine()` format of your `Scanner`. This reads multiple words/chunks of text from the user until the `ENTER` or `RETURN` key is used

Strings are collections of `char` character values, and each character in the String can be accessed by its position in the `String` variable using `variableName.charAt(X)`, where `X` is the position of the character you want.

You can access the total number of characters in a `String` by accessing the `variableName.length()` of the `String` variable.

The next page has some example code that demonstrates some different techniques for using a `String`.

Java

```java
import java.util.Scanner;
public class UsingStrings {
    public static void main ( String[] args ) {
        Scanner get = new Scanner ( System.in );
        String sentence;   // declare a String variable to store data

        System.out.print ( "Enter something: " );
        // Read an entire line of text into variable sentence
        sentence = get.nextLine();

        // Do different kinds of output
        System.out.printf ( "This is what you entered:\n%s\n", sentence );
        System.out.println ( "You entered: " + sentence );

        // Use .length() to determine how many characters are in the String
        System.out.printf ( "You entered %d characters\n", sentence.length() );
        int numChars = sentence.length();
        if ( numChars <= 10 ) {
             System.out.println ( "This is a short sentence!" );
        }

        // Explicitly store something into variable sentence.
        sentence = "Hello!";
        // Positions in a String start at 0, so if there are 6 characters in the
        // String, they will be numbered in positions 0 through 5.  To access the
        // first, third and fifth characters positionally in the String:
        System.out.printf ( "1st character: %c\n", sentence.charAt(0) ); // H
        System.out.printf ( "3rd character: %c\n", sentence.charAt(2) ); // l
        System.out.printf ( "5th character: %c\n", sentence.charAt(4) ); // o

        // Positions in a String start at 0, so if there are 6 characters in the
        // String, they will be numbered in positions 0 through 5.  To access the
        // second, fourth and sixth characters positionally in the String:
        int pos = 1;
        System.out.printf ( "2nd character: %c\n", sentence.charAt(pos) ); // e
        pos = pos + 2;
        System.out.printf ( "4th character: %c\n", sentence.charAt(pos) ); // l
        System.out.printf ( "6th character: %c\n", sentence.charAt(pos + 2) ); // !
    }
}
```

Goals

- Repurpose code you've already written and enhance its functionality. It's possible you may need to rewrite portions of your existing codebase.
- Use the new data type `String` to store a sentence to process and use some new features of `Scanner` and `String` to process `String` data.

Class and File Naming

Name your class `NetID_Sentence` and source file `NetID_Sentence.java`, using your UNO NetID. For example, account `jsmith` would create a class `jsmith_Sentence` in a file named `jsmith_Sentence.java`.

Points to Think About

- This time, you're going to input an entire sentence of any length and convert it from beginning to end.
- A counter-controlled loop will be useful to work through the entire sentence.
- `variable.length()` will be useful to get the number of characters in the `String`
- `variable.charAt(X)` will be useful to get an individual character in the `String`
- Make sure to handle both upper and lfowercase inputs and provide your ciphertext conversion in uppercase. In other words, both `f` and `F` as input would produce capital `S` as output.
- How will you handle outputting 5-character chunks of output? In other words, how will you keep track of outputting a space after displaying five characters of ciphertext?

Testing Your Program

- One way to test your program would be to enter a "sentence" with the alphabet, such as `ABCD EFG HIJKLKMN PQRS TUVWX YZ` and see if it outputs the ROT13 conversions in five-character chunks, which would be `NOPQR STUVW XYXZA CDEFG HIJKL M`.

What About Spaces and Punctuation in the Input?

For this assignment, the `CANNOT CONVERT` portion of your program is no longer useful. If your program encounters text that it cannot process, it should ignore it.

Grading Notes

- You must have an Honor Pledge on file for the course in order for this program to be graded.
- You must have header documentation at the top of your program. Header documentation is worth 5% of your grade.
- You must use a `switch` statement to do the actual conversion of plaintext to ciphertext.
- You may only use material through lecture 11 of the class. Using material beyond lecture 11 will result in points being deducted from your assignment or having the assignment returned as not acceptable.
- Make sure your output includes groupings of five letters each separated by at least one space. The last group of output may fewer than five characters.

Working With New Concepts / Developing This Program

When starting to work with new programming concepts, it's a good idea to write small programs to test those concepts before trying to use them in larger, more complex programs.

For this program, since you're just starting to work with the `String` data type, you may wish to write a small program that allows you enter a sentence and then does some maniuplation with the `String` without worrying about converting it with ROT13.

For instance, can you input a string such as `Hello there!` and then first output the entire sentence and then output that sentence with a space between each character, such as `H e l l o t h e r e !`?

Can you remove all of the spaces in your output, so that `Hello there!` is just `Hellothere!`?

Can you modify your code so you just output the letters, so that `Hello, how are you doing today? I am fine, thanks for asking!` becomes `HellohowareyoudoingtodayIamfinethanksforasking`?

Can you enhance that code to output just the letters in 5-character chunks? So, if your input was `Why, you stuck up, half-witted, scruffy-looking Nerfherder!`, can you produce the output `Whyyo ustuc kupha lfwit tedsc ruffy looki ngNer fherd er`?

If you can do all of those things, then you'll have a good grasp of how to work with `String` data and most of the tasks you need to be able to peform for this assignment.

You can then move on to incorporate the ROT13 conversion of the characters so that `Why, you stuck up, half-witted, scruffy-looking Nerfherder!` becomes `JULLB HFGHP XHCUN YSJVG GRQFP EHSSL YBBXV ATARE SUREQ RE`.

Sample Run

```
Enter a sentence to ROT13: meet me in the northern courtyard
ORIG:  meet me in the northern courtyard
ROT13: ZRRGZ RVAGU RABEG UREAP BHEGL NEQ
```

Sample Run

```
Enter a sentence to ROT13: ZRRGZ RVAGU RABEG UREAP BHEGL NEQ
ORIG:  ZRRGZ RVAGU RABEG UREAP BHEGL NEQ
ROT13: MEETM EINTH ENORT HERNC OURTY ARD
```

Sample Run

```
Enter a sentence to ROT13: Supercalifragilisticexpialidocious
ORIG:  Supercalifragilisticexpialidocious
ROT13: FHCRE PNYVS ENTVY VFGVP RKCVN YVQBP VBHF
```

Sample Run

```
Enter a sentence to ROT13: FHCRE PNYVS ENTVY VFGVP RKCVN YVQBP VBHF
ORIG:  FHCRE PNYVS ENTVY VFGVP RKCVN YVQBP VBHF
ROT13: SUPER CALIF RAGIL ISTIC EXPIA LIDOC IOUS
```

Sample Run

```
Enter sentence to ROT13: All we have to decide is what to do with the time that is given us.
ORIG:   All we have to decide is what to do with the time that is given us.
ROT13:  NYYJR UNIRG BQRPV QRVFJ UNGGB QBJVG UGURG VZRGU NGVFT VIRAH F
```

Sample Run

```
Enter a sentence to ROT13: NYYJR UNIRG BQRPV QRVFJ UNGGB QBJVG UGURG VZRGU NGVFT VIRAH F
ORIG:  NYYJR UNIRG BQRPV QRVFJ UNGGB QBJVG UGURG VZRGU NGVFT VIRAH F
ROT13: ALLWE HAVET ODECI DEISW HATTO DOWIT HTHET IMETH ATISG IVENU S
```

Sample Run

```
Enter a sentence to ROT13: Who?  What?  When?  Where?  Why!?
ORIG:  Who?  What?  When?  Where?  Why!?
ROT13: JUBJU NGJUR AJURE RJUL
```

Sample Run

```
Enter a sentence to ROT13: JUBJU NGJUR AJURE RJUL
ORIG:  JUBJU NGJUR AJURE RJUL
ROT13: WHOWH ATWHE NWHER EWHY
```

Sample Run

```
Enter a sentence to ROT13: Why, you stuck up, half-witted, scruffy-looking Nerfherder!
ORIG:  Why, you stuck up, half-witted, scruffy-looking Nerfherder!
ROT13: JULLB HFGHP XHCUN YSJVG GRQFP EHSSL YBBXV ATARE SUREQ RE
```