Pseudocode for **encrypt** function – would be void, not returning a value.

REQUIRED VARIABLES:

String : **source, cipher**

Char : oldChar, newChar

Output to console “Type in a message, and it will be encrypted.”

Input from console – sentence 🡪 **source**

Initialise **cipher** to be “”

FOR **c**=0 to number of chars in **source** (+1 each time)

**oldChar** 🡨 **c**th character in **source**

**newChar** 🡨 “”

IF (**oldChar** is between ‘a’ and ‘z’)

**ascii** 🡨 ascii code for **OldChar**

**pos** 🡨 **ascii** – ascii code for ‘a’

**code** 🡨 ascii code for ‘z’ - **pos**

**newChar** 🡨 ascii character **code**

ELSE

**newChar** 🡨 **oldChar**

ENDIF

**cipher** 🡨 **newChar** on the front of old version of **cipher**;

ENDFOR

Output to console **cipher**;

**TESTING PLAN**

|  |  |  |  |
| --- | --- | --- | --- |
| Test No | Data to be input | Expected Result | Actual result |
| 1 | abc | zyx |  |
| 2 | No! | !lN |  |
| 3 | hello2 | 2loovs |  |

**Development into code**

**Linking with other code :**

**Calls :** no other code

**Called from :** GetPassword(), SetPassword() within User class

public void encrypt()

{

String source,cipher;

char oldChar, newChar;

output(“Type in a message, and it will be encrypted.”);

source = input(“”);

cipher = “”;

for (int c=0; c<source.length(); c=c+1 )

{

// for each iteration of loop, extract cth character

oldChar = source.charAt(c);

newChar = ‘ ‘;

if ( (oldChar >= ‘a’) && (oldChar <= ‘z’) )

{

// convert each character into encrypted one

int ascii = (int)oldChar;

int pos = ascii - (int)’a’;

int code = ((int)’z’) - pos;

newChar = (char)code;

}

else

{

// for non a-z characters, let them through

newChar = oldChar;

}

// add newly encrypted character to start of cypher string

cipher = newChar + cipher;

}

output(cipher);

}