COS284 Assignment 2

University of Pretoria
Department of Computer Science

Online Publication: 21 August 2023

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Plagiarism Policy

- All work submitted must be your own. You may not copy from classmates, textbooks, or other resources unless explicitly allowed.
- You may discuss the problem with classmates, but you may not write or debug code together.
- If you use any advanced material, you must cite these sources.

Quest Brief: Decrypt the Ancient Scroll

For this quest, you are tasked with mastering the ancient Vigenere cipher. This will allow you to send encrypted messages to allies and decipher the secrets of old. To guide you, consult the tome of knowledge: `https://en.wikipedia.org/wiki/Vigen%C3%A8re_cipher`. Remember, in this cipher, spaces vanish like mist, so "HELLO WORLD" becomes "HELLOWORLD" before you cast your encryption spell. Assume your incantations will always be in uppercase runes without any mystical symbols.

Challenge 1: Conjure the Matrix [15%]

For this challenge, you must craft an assembly spell named
`char** populateMatrix()` that will conjure a 26x26 matrix of runes. Each row of this matrix will have the alphabet shifted left based on the row's mystical index (i.e., row 0 remains unchanged, row 1 shifts left by 1, row 2 shifts left by 2, and so on). The spell must return a pointer to this matrix. Your matrix would look something like if implemented correctly:

```
’A’ ’B’ ’C’ ’D’ ’E’ ’F’ ’G’ ’H’ ’I’ ’J’ ’K’ ’L’ ’M’ ’N’ ’O’ ’P’ ’Q’ ’R’ ’S’ ’T’ ’U’ ’V’ ’W’ ’X’ ’Y’ ’Z’
’B’ ’C’ ’D’ ’E’ ’F’ ’G’ ’H’ ’I’ ’J’ ’K’ ’L’ ’M’ ’N’ ’O’ ’P’ ’Q’ ’R’ ’S’ ’T’ ’U’ ’V’ ’W’ ’X’ ’Y’ ’Z’ ’A’
’C’ ’D’ ’E’ ’F’ ’G’ ’H’ ’I’ ’J’ ’K’ ’L’ ’M’ ’N’ ’O’ ’P’ ’Q’ ’R’ ’S’ ’T’ ’U’ ’V’ ’W’ ’X’ ’Y’ ’Z’ ’A’ ’B’
’D’ ’E’ ’F’ ’G’ ’H’ ’I’ ’J’ ’K’ ’L’ ’M’ ’N’ ’O’ ’P’ ’Q’ ’R’ ’S’ ’T’ ’U’ ’V’ ’W’ ’X’ ’Y’ ’Z’ ’A’ ’B’ ’C’
’E’ ’F’ ’G’ ’H’ ’I’ ’J’ ’K’ ’L’ ’M’ ’N’ ’O’ ’P’ ’Q’ ’R’ ’S’ ’T’ ’U’ ’V’ ’W’ ’X’ ’Y’ ’Z’ ’A’ ’B’ ’C’ ’D’
’F’ ’G’ ’H’ ’I’ ’J’ ’K’ ’L’ ’M’ ’N’ ’O’ ’P’ ’Q’ ’R’ ’S’ ’T’ ’U’ ’V’ ’W’ ’X’ ’Y’ ’Z’ ’A’ ’B’ ’C’ ’D’ ’E’
’G’ ’H’ ’I’ ’J’ ’K’ ’L’ ’M’ ’N’ ’O’ ’P’ ’Q’ ’R’ ’S’ ’T’ ’U’ ’V’ ’W’ ’X’ ’Y’ ’Z’ ’A’ ’B’ ’C’ ’D’ ’E’ ’F’
’H’ ’I’ ’J’ ’K’ ’L’ ’M’ ’N’ ’O’ ’P’ ’Q’ ’R’ ’S’ ’T’ ’U’ ’V’ ’W’ ’X’ ’Y’ ’Z’ ’A’ ’B’ ’C’ ’D’ ’E’ ’F’ ’G’
’I’ ’J’ ’K’ ’L’ ’M’ ’N’ ’O’ ’P’ ’Q’ ’R’ ’S’ ’T’ ’U’ ’V’ ’W’ ’X’ ’Y’ ’Z’ ’A’ ’B’ ’C’ ’D’ ’E’ ’F’ ’G’ ’H’
’J’ ’K’ ’L’ ’M’ ’N’ ’O’ ’P’ ’Q’ ’R’ ’S’ ’T’ ’U’ ’V’ ’W’ ’X’ ’Y’ ’Z’ ’A’ ’B’ ’C’ ’D’ ’E’ ’F’ ’G’ ’H’ ’I’
’K’ ’L’ ’M’ ’N’ ’O’ ’P’ ’Q’ ’R’ ’S’ ’T’ ’U’ ’V’ ’W’ ’X’ ’Y’ ’Z’ ’A’ ’B’ ’C’ ’D’ ’E’ ’F’ ’G’ ’H’ ’I’ ’J’
’L’ ’M’ ’N’ ’O’ ’P’ ’Q’ ’R’ ’S’ ’T’ ’U’ ’V’ ’W’ ’X’ ’Y’ ’Z’ ’A’ ’B’ ’C’ ’D’ ’E’ ’F’ ’G’ ’H’ ’I’ ’J’ ’K’
’M’ ’N’ ’O’ ’P’ ’Q’ ’R’ ’S’ ’T’ ’U’ ’V’ ’W’ ’X’ ’Y’ ’Z’ ’A’ ’B’ ’C’ ’D’ ’E’ ’F’ ’G’ ’H’ ’I’ ’J’ ’K’ ’L’
’N’ ’O’ ’P’ ’Q’ ’R’ ’S’ ’T’ ’U’ ’V’ ’W’ ’X’ ’Y’ ’Z’ ’A’ ’B’ ’C’ ’D’ ’E’ ’F’ ’G’ ’H’ ’I’ ’J’ ’K’ ’L’ ’M’
’O’ ’P’ ’Q’ ’R’ ’S’ ’T’ ’U’ ’V’ ’W’ ’X’ ’Y’ ’Z’ ’A’ ’B’ ’C’ ’D’ ’E’ ’F’ ’G’ ’H’ ’I’ ’J’ ’K’ ’L’ ’M’ ’N’
’P’ ’Q’ ’R’ ’S’ ’T’ ’U’ ’V’ ’W’ ’X’ ’Y’ ’Z’ ’A’ ’B’ ’C’ ’D’ ’E’ ’F’ ’G’ ’H’ ’I’ ’J’ ’K’ ’L’ ’M’ ’N’ ’O’
’Q’ ’R’ ’S’ ’T’ ’U’ ’V’ ’W’ ’X’ ’Y’ ’Z’ ’A’ ’B’ ’C’ ’D’ ’E’ ’F’ ’G’ ’H’ ’I’ ’J’ ’K’ ’L’ ’M’ ’N’ ’O’ ’P’
’R’ ’S’ ’T’ ’U’ ’V’ ’W’ ’X’ ’Y’ ’Z’ ’A’ ’B’ ’C’ ’D’ ’E’ ’F’ ’G’ ’H’ ’I’ ’J’ ’K’ ’L’ ’M’ ’N’ ’O’ ’P’ ’Q’
’S’ ’T’ ’U’ ’V’ ’W’ ’X’ ’Y’ ’Z’ ’A’ ’B’ ’C’ ’D’ ’E’ ’F’ ’G’ ’H’ ’I’ ’J’ ’K’ ’L’ ’M’ ’N’ ’O’ ’P’ ’Q’ ’R’
’T’ ’U’ ’V’ ’W’ ’X’ ’Y’ ’Z’ ’A’ ’B’ ’C’ ’D’ ’E’ ’F’ ’G’ ’H’ ’I’ ’J’ ’K’ ’L’ ’M’ ’N’ ’O’ ’P’ ’Q’ ’R’ ’S’
’U’ ’V’ ’W’ ’X’ ’Y’ ’Z’ ’A’ ’B’ ’C’ ’D’ ’E’ ’F’ ’G’ ’H’ ’I’ ’J’ ’K’ ’L’ ’M’ ’N’ ’O’ ’P’ ’Q’ ’R’ ’S’ ’T’
’V’ ’W’ ’X’ ’Y’ ’Z’ ’A’ ’B’ ’C’ ’D’ ’E’ ’F’ ’G’ ’H’ ’I’ ’J’ ’K’ ’L’ ’M’ ’N’ ’O’ ’P’ ’Q’ ’R’ ’S’ ’T’ ’U’
’W’ ’X’ ’Y’ ’Z’ ’A’ ’B’ ’C’ ’D’ ’E’ ’F’ ’G’ ’H’ ’I’ ’J’ ’K’ ’L’ ’M’ ’N’ ’O’ ’P’ ’Q’ ’R’ ’S’ ’T’ ’U’ ’V’
’X’ ’Y’ ’Z’ ’A’ ’B’ ’C’ ’D’ ’E’ ’F’ ’G’ ’H’ ’I’ ’J’ ’K’ ’L’ ’M’ ’N’ ’O’ ’P’ ’Q’ ’R’ ’S’ ’T’ ’U’ ’V’ ’W’
’Y’ ’Z’ ’A’ ’B’ ’C’ ’D’ ’E’ ’F’ ’G’ ’H’ ’I’ ’J’ ’K’ ’L’ ’M’ ’N’ ’O’ ’P’ ’Q’ ’R’ ’S’ ’T’ ’U’ ’V’ ’W’ ’X’
’Z’ ’A’ ’B’ ’C’ ’D’ ’E’ ’F’ ’G’ ’H’ ’I’ ’J’ ’K’ ’L’ ’M’ ’N’ ’O’ ’P’ ’Q’ ’R’ ’S’ ’T’ ’U’ ’V’ ’W’ ’X’ ’Y’
```

Challenge 2: Rune Transformation [25%]

For this challenge, you must craft an assembly spell named `char encryptChar(char**, char, char)`. This spell will take a `char**` (the matrix from challenge 1) and two `char`s, transforming the first `char` using the matrix and the key rune provided as the second `char` as described in the cipher's ancient scrolls. The spell must return the transformed rune.

Example: (This is not a prophecy, merely an example)

```
Given the rune: A
Given the key rune: L
The spell should produce: L
```

The Operation happening should look something like this:

```
B - Z -> A
Z - Z -> Y
A - L -> L
O - P -> D
```

Challenge 3: Encrypt the Message [35%]

For this challenge, you must craft an assembly spell named `char* encryptString(char**, char*, char*)`. This spell will take the matrix from `populateMatrix()`, and two strings of runes (the message and the keyword). Using the corresponding runes from both strings, it will repeatedly invoke the spell from challenge 2 to encrypt the entire message. The spell must then return the fully encrypted message.

Example:

```
Given the message: ATTACK AT DAWN
Using the keyword: LEMON
The spell should produce: LXFOPVEFRNHR
```

Once your quest is complete, seal your spells in an archive named **populateMatrix.asm, encryptChar.asm, encryptString.asm** and send it through the ethereal portal at ff.cs.up.ac.za, under the **Assignment 2** submission portal.

A scroll on the quest challenges:

For all challenges, the Mage's Guild will provide you with a C-based incantation circle that will handle input and output. You need only to master the spells that will be invoked from within this circle and ensure they produce the correct magical effects.

1 Reward Distribution

Challenge	**Reward**
Challenge 1	15
Challenge 2	25
Challenge 3	35
Total	**75**