Writing Libraries: Caesar Cipher

16 Oct 2009
CMPT140
Dr. Sean Ho
Trinity Western University

TRINITY WESTERN UNIVERSITY

Computing & Society Paper

- Computing scientist as **G**odly **C**hristian **L**eader:
  - Not just knowledge about tools, but
  - Wisdom of how to use tools
    - To serve others and
    - To give glory to God
- Write a short essay on a topic of your choosing about computers and society:
    - ~ 5 pages typed double-spaced 12pt 1in margins
    - Submit half-page topic by Fri 6Nov
    - Paper due near end of semester (Wed 2Dec)
      - Electronic submission (email, eCourses)

Sample paper topics

- Censorship and free speech
  - Pornography, gambling, hate groups, etc.
- Violence in video games (Columbine etc.)
- Privacy: online banking, ID theft, etc.
- Blogs: effect on politics, social interaction, etc.
- File sharing: Napster, Gnutella, etc.
- Artificial intelligence: the nature of sentience
- Online dating (e.g. eHarmony): pros/cons
- Equity of access / rural digital divide
  - ..... or come up with your own topic!

Tips for essay writing

- Your essay should be a position paper:
  - Topic should have at least two sides (e.g. pro/con)
  - You should state (in the introductory paragraph) what your position is (thesis)
  - You should have at least 2-3 points, each, both for and against your position
    - It is not necessary to rebut every point that contradicts your position:
    - Be honest about faults/limitations of your thesis
  - Summary intro/conclusion paragraphs
  - Proper English (spelling, grammar) is important!

Prime numbers in the news

- 15 Oct 2009 news article: "12-million-digit prime number sets record, nets $100,000 prize"
- 45th Mersenne prime found so far: $2^{43,112,609} - 1$
- Found by UCLA math dept computers as part of the GIMPS project (Great Internet Mersenne Prime Search)
- Sysadmin Edson Smith installed Prime95 program as screensaver in their 75-seat computer lab
- Full list of 12.9M digits is over 7MB!

Library modules vs. programs

- So far we've been writing Python programs (e.g., helloworld.py)
- Our programs have used library modules (e.g., import math)

- Libraries group related code for reuse (import)
  - Only need to define cos() once
  - Libraries are not intended to be executed (called), unlike programs
- We can create our own libraries for others to use

Designing libraries

- In creating a library, we need to decide what the public interface is: how programs can use it
  - Functions, types, constants, etc. for public use
  - Think about pre-/post-conditions
- We can hide implementation details
  - Certain functions may be for internal use only
- Car: how to use it vs. how it works
  - Owner's manual vs. shop manual
  - A driver doesn't need to understand how the engine works, variable valve timing/lift, etc.

Definition vs. implementation

- In M2, each library has a definition file and an implementation file:
  - DEF: declares types and procedures
    - Tells programs how to invoke its procedures
    - No bodies to the procedures
  - IMP: implements the procedures
    - Parameter lists must match those in DEF file
- In C/C++, definition files are called header files (.h, .H, .hpp)
- In Python, everything is in one .py file

Null-termination in strings

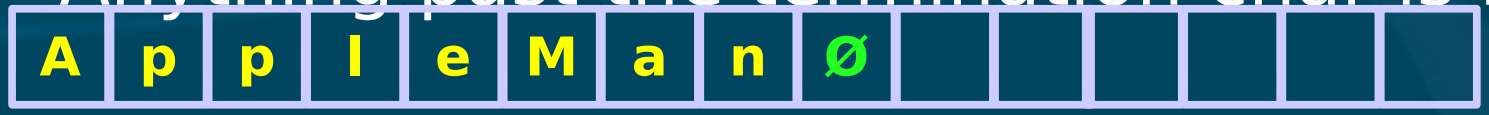
- In Python, strings are a basic type
- But in M2/C, strings are fixed-len arrays of CHAR:

```
VAR myName : ARRAY [0..14] OF CHAR;
```

- But the array is not always completely filled:

```
myName := “AppleMan”;
```

- How to know where the string ends?
- Strings are null-terminated:
  - The null character chr(0) is added to the end
  - Anything past the termination char is ignored



Cryptography example

- Cæsar substitution cipher:
  - Key: e.g., QAZXSWEDCVFRTGBNHYUJMKIOLP
  - Cleartext: input text to encrypt
  - Ciphertext: output encrypted text
  - Encoding: replace each letter in source with corresponding letter from code key
  - Decoding: same, using the decode key
- ROT13 was an example of a substitution cipher
  - Key: NOPQRSTUVWXYZABCDEFGHIJKLM

Write a Substitution library

- Design a public interface for the library?

```
def encode (src, key):
    """Encode the source string using the given
      codestring.
    Returns the encoded string.
    pre: src must be a string;
    key must be a permutation of the 26 letters."""
def decode (src, key):
    """Decode the source string using the given
      codestring.
    Returns the decoded string.
    pre: src must be a string;
    key must be a permutation of the 26 letters."""
```

Internal helper functions

- In the implementation it is handy to have some helper functions for internal use:

```
def isalpha (ch):
    """Return true if ch is a letter."""
def alpha_pos (ch):
    """Return index of a letter in the range 0 .. 25"""
def decode_key (enckey):
    """Create a decode key from an encoding key"""
```

- How to implement these?
  - isalpha() is built-in: ch.isalpha()

Implementing Substitution

- Main function to encode strings:

```
def encode(src, key):
    """Encode the source string using the given
      codestring.
    Returns the encoded string.
    pre: src must be a string;
    key must be a permutation of the 26 letters.
    """
    dst = ""
    for ch in src:
        if ch.isalpha():
            dst += key[alpha_pos(ch)]
        else:
            dst += ch
    return dst
```

Implementing decode()

- Decoding is just encoding using a reverse key:

```
def decode (src, key):
    """Decode the source string using the given
      codestring.
    Returns the decoded string.
    pre: src must be a string;
    key must be a permutation of the 26 letters.
    """
    return encode(src, decode_key(key))
```

- **Library: http://twu.seanho.com/python/substitution.py**
- **Testbed: http://twu.seanho.com/python/caesartest.py**