ML

Commentary

Once again, your lesson in class was a big help for jump starting these programs. After that class I still needed to figure out how to convert to uppercase, test if a character is a letter, and figure out how to map over a string. Although, there is not a great official documentation for the language, alot of the things I was searching for was already on stack overflow, something I cannot say the same for COBOL. I find it interseting that each of these languages use a different word for function, and then have a different second word for an anonymous function. The program was actually pretty simple to write after I had done it a few times. No really big frustrations on this one. Main things that kept me delayed was just finding the new syntax.

Google Searches

- ML install
- ml install ubuntu
- polyml ubuntu run script
- ml get char code
- ml to uppercase
- ml string map
- ml map over string
- ml enumerate
- ml foreach
- ml run x times

Caesar Implementation

```
fun rot (char, shift) = if ((ord (Char.toUpper char)) >= 65 andalso (ord
(Char.toUpper char)) <= 90) then
        (chr (((ord (Char.toUpper char)) - 65 + shift) mod 26 + 65))
    else
        char;

fun encrypt (word, shift) = (String.map (fn ltr => rot(ltr, shift)) word);
fun decrypt (word, shift) = encrypt(word, 26-shift);
fun solve (word, maxShift) = map
    (fn x => print(Int.toString(x) ^ "\t" ^  (decrypt(word, x)) ^ "\n"))
    (List.tabulate(maxShift, fn x => x)); (* run it maxShift times *)

print (encrypt("ATTACK AT ONCE", 4) ^ "\n");
print (decrypt("EXXEGO EX SRGI", 4)^ "\n");
solve("abcdeFGHIJKLmnopqrstuvwxyz ,?;{[()]}", 26);
```

Output

```
EXXEGO EX SRGI
ATTACK AT ONCE
0       ABCDEFGHIJKLMNOPQRSTUVWXYZ ,?;{[()]}
1       ZABCDEFGHIJKLMNOPQRSTUVWXY ,?;{[()]}
2       YZABCDEFGHIJKLMNOPQRSTUVWX ,?;{[()]}
3       XYZABCDEFGHIJKLMNOPQRSTUVW ,?;{[()]}
4       WXYZABCDEFGHIJKLMNOPQRSTUV ,?;{[()]}
5       VWXYZABCDEFGHIJKLMNOPQRSTU ,?;{[()]}
6       UVWXYZABCDEFGHIJKLMNOPQRST ,?;{[()]}
7       TUVWXYZABCDEFGHIJKLMNOPQRS ,?;{[()]}
8       STUVWXYZABCDEFGHIJKLMNOPQR ,?;{[()]}
9       RSTUVWXYZABCDEFGHIJKLMNOPQ ,?;{[()]}
10      QRSTUVWXYZABCDEFGHIJKLMNOP ,?;{[()]}
11      PQRSTUVWXYZABCDEFGHIJKLMNO ,?;{[()]}
12      OPQRSTUVWXYZABCDEFGHIJKLMN ,?;{[()]}
13      NOPQRSTUVWXYZABCDEFGHIJKLM ,?;{[()]}
14      MNOPQRSTUVWXYZABCDEFGHIJKL ,?;{[()]}
15      LMNOPQRSTUVWXYZABCDEFGHIJK ,?;{[()]}
16      KLMNOPQRSTUVWXYZABCDEFGHIJ ,?;{[()]}
17      JKLMNOPQRSTUVWXYZABCDEFGHI ,?;{[()]}
18      IJKLMNOPQRSTUVWXYZABCDEFGH ,?;{[()]}
19      HIJKLMNOPQRSTUVWXYZABCDEFG ,?;{[()]}
20      GHIJKLMNOPQRSTUVWXYZABCDEF ,?;{[()]}
21      FGHIJKLMNOPQRSTUVWXYZABCDE ,?;{[()]}
22      EFGHIJKLMNOPQRSTUVWXYZABCD ,?;{[()]}
23      DEFGHIJKLMNOPQRSTUVWXYZABC ,?;{[()]}
24      CDEFGHIJKLMNOPQRSTUVWXYZAB ,?;{[()]}
25      BCDEFGHIJKLMNOPQRSTUVWXYZA ,?;{[()]}
```

Log

Estimate: 2 hours

Date	Hours Spent	Accomplishments
4/6	.25	Install local version of ML
4/6	.25	Create rot function
4/6	.25	Create encrypt/decrypt
4/6	.5	Create solve
4/20	.5	Fix special characters

Discrepancy of time

I basically finish all of these programs at the same time period. If I ever get ahead it because I either skipped something or I'm just yet to have the inevitable blocker. For example here, I was done with encrypt and decrypt in 45 minutes, but the solve and getting special characters to work was a much longer process.

Overall Review

Even though ML went pretty well for me, I would only rank the language itself over lisp. There also wasn't a lot of documentation and I would have been stuck for much longer without your basic rundown. Erlang is definitely better and JS and Scala are the only legitimate programming languages in the assignment. Most of the operators are what I would expect them to be, aside from andalso. Who in there right mind came up with that? I think I read that or is referred to as orelse???? Things like this are what lead to a costly language. Erlang, I believe is a worse offender but ML definitely has it's quirks.

Ratings

Readability: 6/10

Writability: 6/10

Ranking: 4/5