Programming assignments 3  25 Points

(Do NOT work in groups for programming assignments.)

A cryptarithm is a puzzle in which letters are substituted for numbers in an equation. We assume that no two letters have the same value. The term was coined by Maurice Vatriquant in 'Sphinx' magazine in 1931. The following are examples of cryptarithms:

\[
\begin{align*}
\text{SEND} & \quad 5478 \\
\text{MORE} & \quad 1624 \\
\text{GOLD} & \quad 9638 \\
\text{MONEY} & \quad 16740 \\
\text{VENUS} & \quad 54739 \\
\text{EARTH} & \quad 46120 \\
\text{MARS} & \quad 8619
\end{align*}
\]

In this assignment, you are to use Ruby to find the solution for cryptarithms. Our puzzles will be stated as a string. For example

```
puzzle = "SEND+MORE+GOLD==MONEY";
```

You will write the solution as:

```
5478+1624+9638==16740
```

Here are puzzles to try:
NO+GUN+NO==HUNT
WINTER+IS+WINDIER+SUMMER+IS == SUNNIER
FIVE+FIVE+NINE+ELEVEN==THIRTY

**Part 1 Test a specific solution**

**Input:** (test for five problems shown above)

- puzzle: “SEND+MORE+GOLD==MONEY”;
- solution String: YMRLESONDG

**Output:**

Original Puzzle: “SEND+MORE+GOLD==MONEY”
Solution String: YMRLESONDG
Puzzle with substitution: "5478+1624+9638==16740"  CORRECT  (or possibly NOT CORRECT)

Note: The solution string is the ordering of the letters such that the first character in the string is 0, the second is 1, etc. So, for “SEND+MORE+GOLD==MONEY”, the solution string is "YMRLESONDG", as Y is a 0, M is a 1, R is a 2, etc. If your puzzle doesn't use all ten letters, you will need to insert blanks for unused values.

Hints:
Use the ruby command tr to replace the characters of the puzzle with numbers.
  ```ruby
  puzzle.tr!("YMRLESONDG","0123456789")
  ```

The syntax of `tr` is:
```ruby
str.tr!(from_str, to_str) → new_str
```
Returns a copy of `str` with the characters in `from_str` replaced by the corresponding characters in `to_str`. If `to_str` is shorter than `from_str`, it is padded with its last character.

Use the Ruby command eval to evaluation the puzzle (after substitution)
  ```ruby
  res = eval(puzzle)
  ```

If `eval` returns true, we have found a correct solution.

**Part 2 Computer finds the solution**

Finish the assignment by finding a solution to a provided puzzle. You can do this any way you like. Find all solutions, but only list each solution once. The following is only a suggestion. You do not need to solve the problem this way.

**Suggestion:**
Find all the unique letters of your puzzle.

Write a recursive function to permute the unique letters of the puzzle so that all possible permutations are found. Try them (one at a time) as a solution string.

Note, that you do not expect that finding all permutations will run quickly as the number of permutations is ten factorial! **Show that your permutation routine works by showing all permutations of “ABCD”.** (This is a good testing technique. Try with something small so you can follow the recursion and it doesn’t take forever.)

The following is a C++ version of the permute algorithm.

```cpp
void permute(int v[], int start, int n) {
    if (start == n-1) {
        print(v, n); // This is an ordering
    } else {
```
for (int i = start; i < n; i++) {
    int tmp = v[i];
    v[i] = v[start];
    v[start] = tmp;
    permute(v, start+1, n);
    v[start] = v[i];
    v[i] = tmp;
}

Try each permutation of your unique letters until a solution is found or no solution is possible. This is slow, so put your feet up and take a nap while you churn out the answer. During debugging, I found myself wanting some feedback from the brute force trials. I did this via a statement which printed some feedback every 500 calls.

Hints

I had trouble with string assignment doing a shallow copy. 
s = String.new(puzzle) guarantees a copy is made

A number that begins with a zero causes a problem in eval (because it thinks it is an octal number). Remove leading zeros from your solution before using eval (or come up with another solution) or it will complain about numbers that begin with 0 but can't be interpreted as octal numbers.

When you have fewer than ten variables, the same solution will be considered multiple times (as the blanks are permuted).