White Belt Exam

Your name: _________________________  Email ID: _________________________

This exam is your first opportunity to earn a white belt. Students who pass it will be ready to move on to earning a yellow belt; if you do not pass it, you will have other opportunities to earn the white belt.

Closed Resources, No Help. For this exam you should work on your own and are not permitted to use any resources other than your own mind and body, and a simple writing implement.

Answer well. Answer all 5 questions. Your answers should be clear, correct, and concise.

Required Questions

1. What is a computer?

2. What are the primitives in the invented English word “embiggen”?

3. How many different strings can the following BNF grammar produce?

   Sentence ::= I Emotion Food
   Emotion ::= love
   Emotion ::= hate
   Food ::= Bagels
   Food ::= Donuts
4. Consider this excerpt from the Python language grammar (similar to Lesson 1 of cs101):

\[
\text{Expression ::= Expression Operator Expression} \\
\text{Expression ::= Number} \\
\text{Operator ::= +} \\
\text{Operator ::= *} \\
\text{Number = 0, 1, 2}
\]

Circle all the strings below that can be produced by Expression from just this grammar:

- 0
- 1 + 1 + 2 + 0
- 1 1
- + 2

5. Write a sequence of Python statements that will result in the variable sindex holding the index of the second occurrence of the string given in the variable match in string given in the variable quote.

```python
quote = 'In general we are least aware of what our minds do best. (Marvin Minsky)'
match = 'are'

# write the code that goes here
```

After your code executes, the value in sindex should be the index in the original quote string where the second occurrence of match occurs. The given values of quote and match here are just examples; your code should work for any starting string values in these variables. (For this, you don't need to worry about what happens if the quote does not include two occurrences of match.)

For reference, here are the descriptions of the string find functions (from the cs101 notes):

- **find**: `<Search String>.find(<Target String>)` returns a `Number`.
  - Returns a number giving the position in `<Search String>` where `<Target String>` first appears. If there is no occurrence of `<Target String>` in `<Search String>`, returns -1.

- **find after**: `<Search String>.find(<Target String>, <Start Index>)` returns a `<Number>`
  - Returns a number giving the position in `<Search String>` where `<Target String>` first appears that is at or after the position given by `<Start Index>`. If there is no occurrence of `<Target String>` in `<Search String>` at or after `<Start Number>`, outputs -1.