

Name: _____

Solutions

1. (1 pt.)

- **Read all material carefully.**
- *If in doubt whether something is allowed, ask, don't assume.*
- You may refer to your books, papers, and notes during this test.
- E-books may be used.
- Computers are permitted but discouraged.
- **Electronic and network resources must only be used as a passive library.**
- Write, and draw, carefully. Ambiguous or cryptic answers receive zero credit.
- Use class and textbook conventions for notation, algorithmic options, etc.

Write your name in the space provided above.

Do not write anything else on this page.

2. (14 pts.) Consider the JCoCo assembly language program listed below.

- (a) (7 pts.) **Explain** what the program does as precisely as possible. (Recall recent classroom discussion of similar questions.) In particular, describe its output as a function of its input.
- (b) (7 pts.) Provide a **complete JCoCo assembly language program** that exhibits the same input-output behavior as this one but whose code is shorter by at least one instruction, or explain why no such shorter program is possible. If your shorter program reuses parts of this program then you may indicate so instead of rewriting those parts **but only if** the result is completely obvious and unambiguous. **Explain why your answer is correct.**

```
Function: main/0
Constants: ''
Locals: x
Globals: print, input, split
BEGIN
    LOAD_GLOBAL 1
    LOAD_CONST 0
    CALL_FUNCTION 1
    DUP_TOP
    LOAD_ATTR 2
    CALL_FUNCTION 0
```

```

        DUP_TOP
        SETUP_LOOP label2
        GET_ITER
label10: FOR_ITER label11
        LOAD_GLOBAL 0
        ROT_TWO
        CALL_FUNCTION 1
        POP_TOP
        JUMP_ABSOLUTE label10
label11: POP_BLOCK
        BREAK_POINT
label12: RETURN_VALUE
END

```

Ⓐ (Outline) The program reads a single line (string) from standard input and then creates a list whose elements are the whitespace-separated components of that string. These elements are then written to standard output, one per line.

There are several possibilities for shortening the program. Two of them are: (1) delete the `DUP_TOP` instruction (line 4 after the `BEGIN`) and (2) delete the `BREAK_POINT`.

3. (15 pts.) Provide a **complete JCoCo assembly language program** that
- Reads a newline-terminated string from *standard input*.
 - Writes a single integer n followed by a newline to *standard output*, where n is the length (in characters) of the input string.

Explain why your program is correct.

Ⓐ (Outline; the explanation should include the action of each instruction.)

```

Function: main/0
Constants: ""
Globals: print, input, len
BEGIN
        LOAD_GLOBAL          0
        LOAD_GLOBAL          1
        LOAD_CONST           0
        CALL_FUNCTION        1
        LOAD_GLOBAL          2
        ROT_TWO
        CALL_FUNCTION        1
        CALL_FUNCTION        1
        RETURN_VALUE
END

```