Name: ____

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.

Wait until instructed to continue to remaining questions.

(Do not view any other pages.)

Do not write on this page.

(It is for use in grading only. Anything written here will be ignored.)

Q	Full	Score
1	1	
2	15	
3	15	
4	6	
5	8	
total	45	

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, len
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
label0: FOR_ITER label1
        LOAD_CONST 0
        BUILD_LIST 2
        LOAD_GLOBAL 0
        ROT_TWO
        CALL_FUNCTION 1
        POP_TOP
        JUMP_ABSOLUTE label0
label1: POP_BLOCK
        BREAK_POINT
label2: RETURN_VALUE
END
```

[additional space for earlier material]

[additional space for earlier material]

3. (15 pts.) Provide a **complete JCoCo assembly language program** that

   (a) Reads two newline-terminated string from *standard input* (one string per line).

   (b) Writes a single integer n followed by a newline to *standard output*, where n is the product of the lengths (in characters) of the two input strings.

   Explain why your program is correct.

[additional space for earlier material]

4. (6 pts.) For each of the following *Standard ML* expressions, provide the response when that expression is evaluated by the `sml` REPL (read-eval-print loop). Assume that the expressions are evaluated in the order listed. In your response, *draw a **box** around the **type** and `oval` around the **value**.* (If there is an error then clearly explain the error.)

   (a) (2 pts.) `42.42 / 2.0;`

   (b) (2 pts.) `42 / 2;`

   (c) (2 pts.) `42 / 2.0;`

5. (8 pts.) For each of the following SML expressions: (1) **State** its type and (2) **explain** how that type is inferred (using the recent classroom discussion type inference as a model).

   (a) (4 pts.)

   ```
   fun f101(x) = x + 101;
   ```

   (b) (4 pts.)

   ```
   fun f301 (i, j) =
       if i < j then
           2 * i + j + 1
       else
           j * f301(i - 1, j + 1);
   ```

[additional space for earlier material]