

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.
- **Do not attach or remove any pages.** Questions must be answered only on the provided pages.

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	10
3	5
4	5
5	15
6	10
7	10
8	10
9	5
10	5
11	5
12	10
13	5
14	5
total	100

2. (10 pts.) Using *Standard ML*, define a function `fi` that takes a pair of lists as argument and returns a list whose elements are the elements of the two lists picked alternately in order, starting with the first element of the first list. If the lists are of unequal lengths then the remaining elements from the longer list appear at the end (without any interleaving elements). The following examples illustrate the desired behavior of `fi`.

```
1   fi ([10, 11, 12, 13], [20, 21, 22, 23]) =  
    [10,20,11,21,12,22,13,23]  
2  
3   fi ([10, 11, 12, 13], [20, 21]) = [10,20,11,21,12,13]  
4  
5   fi ([10, 11], [20, 21, 22, 23]) = [10,20,11,21,22,23]  
6  
7   fi (["Hello,", "World!"], ["I", "prefer", "to", "be"]) = ["Hello,",  
    "I", "World!", "prefer", "to", "be"]  
8  
9   fi (["I", "prefer", "to", "be"], ["Hello,", "World!"]) = ["I", "  
    Hello,", "prefer", "World!", "to", "be"]
```

3. (5 pts.) Explain as precisely as possible why your SML definition of Question 2 is correct.
4. (5 pts.) Trace the operation of your SML code of Question 2 on last example provided there.

[additional space for earlier material]

5. (15 pts.) Provide a **complete JCoCo assembly language program** that
- Reads a single newline-terminated string from *standard input*. (Everything up to but excluding the newline is the input string, which may include spaces.)
 - Writes a single integer n followed by a newline to *standard output*, where n is the number of spaces (character count) in the input string.
6. (10 pts.) Explain why your program of Question 5 is correct using a suitable combination of inline comments and separate text.
7. (10 pts.) Trace the operation of your program of Question 5 when the string `I Hi! y o` (followed by a newline) appears on standard input. (The string has 11 characters and spaces are depicted as `␣` for clarity.)

[additional space for earlier material]

[additional space for earlier material]

8. (10 pts.) Consider the following grammar (using *yacc/PLY* syntax):

```
S : A S B | c
A : a c a | a c A a
B : B b | b | b S
```

Does the sentence `acacacbacacb` belong to the language of this grammar? If it does then provide a corresponding **leftmost derivation** and a **parse tree**, both *using class and textbook conventions*; otherwise, provide a **proof** (as precise as possible) that it does not.

9. (5 pts.) Consider the following regular expression. (The regular expression uses the textbook's conventions. Tokens `a`, `b`, and `c` are character literals.)

`a.b.(c.(a + b))*`

- (a) What is the minimum length of a string that matches this regular expression?
 - (b) Why is that the minimum length?
 - (c) Provide such a string.
10. (5 pts.) Provide two distinct strings of length 3 that match the regular expression of Question 12, or explain as precisely as possible why no such strings exist.
11. (5 pts.) Repeat Question 10 for length 4 instead of 3.

12. (10 pts.) Provide a context-free grammar (CFG) for the language defined by the following regular expression. (The set of strings that can be derived using the CFG must be the same as the set of strings matching the regular expression.) **Explain** briefly why your answer is correct.

13. (5 pts.) For each string of Question 10, either provide a leftmost derivation using the grammar of Question 12 or explain as precisely as possible why no such derivation exists.

14. (5 pts.) Repeat Question 13 for the strings of Question 11 instead of those of Question 10.

[additional space for earlier material]