

NATIONAL UNIVERSITY OF SINGAPORE
CS1101S — PROGRAMMING METHODOLOGY

AY2022/2023 SEMESTER 1

READING ASSESSMENT 1

Time Allowed: **45 Minutes**

INSTRUCTIONS

1. This assessment contains **19 Multiple-Choice Questions** in **4 Sections**.
2. Each question has one correct answer. **1 mark** is awarded for each correct answer and there is no penalty for a wrong answer.
3. The full score of this assessment is **19 marks**.
4. Answer **all questions**.
5. This is a **Closed-Book** assessment, but you are allowed one double-sided **A4 / foolscap / letter-sized sheet** of handwritten or printed **notes**.
6. You are allowed to use up to **4 sheets** of **blank A4 / foolscap / letter-sized** paper as **scratch paper**.
7. **Follow the instructions of your invigilator or the module coordinator to submit your answers.**

Section A

(1) What is the result of evaluating the following Source §1 program?

```
const u = 2;
const v = 3;
const w = 4;

function fun(w) {
    const v = 5;
    return (100 * u) + (10 * v) + w;
}
fun(v);
```

- A. 233
- B. 234
- C. 253
- D. 254
- E. Error: one or more names is/are redeclared
- F. Error: one or more names is/are not declared before being used

(2) What is the result of evaluating the following Source §1 program?

```
function fun(u) {
    const x = 1;

    function gee(x) {
        const y = 6;
        return u + x;
    }

    return gee(u) + y;
}

const x = 3;
const y = 4;
const u = 5;
fun(2 + u);
```

- A. 20
- B. 18
- C. 16
- D. 14
- E. 12
- F. 10
- G. Error: one or more names is/are redeclared
- H. Error: one or more names is/are not declared before being used

(3) Consider the following Source §1 function fun:

```
function fun(x) {  
    return t => (u, v) => (x => 2 * x)(t + u + v + x);  
}
```

Which of the following statements shows a correct statement of using the fun function?

- A. fun(1)(2)(3)(4);
- B. fun(1)(2)(3)(4)(5);
- C. fun(1)(2, 3)(4);
- D. fun(1)(2, 3)(4)(5);
- E. fun(1)(2)(3, 4);
- F. fun(1)(2)(3, 4)(5);
- G. None of the other options is a correct statement

(4) What is the result of evaluating the following Source §1 program?

```
const w = 3;  
const x = 5;  
const y = 7;  
(w => (x, y) => (w => (y => y + x)(w))(100 * w + 10 * y))(y)(w, x);
```

- A. 357
- B. 375
- C. 537
- D. 573
- E. 735
- F. 753
- G. Error: one or more names is/are redeclared
- H. Error: wrong kind of argument(s) or wrong number of argument(s)

Section B

IMPORTANT

All programs in this section are written in the **Source R2L** programming language, which is identical to **Source §1**, except the following two differences:

- (1) In Source R2L, the right operand of a binary operation is evaluated before the left operand, and
- (2) In Source R2L, when there are multiple arguments in a function application, the argument expressions are evaluated from right to left, i.e. the rightmost argument expression is evaluated first, and the leftmost last.

In some of the following questions, the pre-declared `display` function is used in the programs. The `display` function displays/prints the value of its input argument in the REPL, *and* returns the value of its input argument. For example, `display(2 * 5)` prints `10` and returns `10`; and `display(1 > 2)` prints `false` and returns `false`.

- (5) What is the sequence of values printed by the `display` function when the following Source R2L program is evaluated?

```
function fun(n) {  
  if (n <= 1) {  
    return 1; // note that it is 1, not n  
  } else {  
    display(n);  
    return fun(n - 1) + fun(n - 2);  
  }  
}  
fun(5);
```

- A. 5 4 3 2 1 0 1 2 1 0 3 2 1 0 1
B. 5 4 3 2 2 3 2
C. 2 3 2 2 3 4 5
D. 5 3 2 4 2 3 2
E. 5 3 4 2 2 3 2
F. 5 4 3 3 2 2 2
G. None of the other options is the correct answer

- (6) What is the sequence of values printed by the `display` function when the following Source R2L program is evaluated?

```
function fun(n) {  
  if (n <= 1) {  
    return 1; // note that it is 1, not n  
  } else {  
    const x = fun(n - 1) + fun(n - 2);  
    display(x);  
    return x;  
  }  
}  
fun(5);
```

- A. 2 3 2 2 3 5 8
 - B. 2 3 2 5 2 3 8
 - C. 2 3 2 4 2 3 5
 - D. 2 3 2 2 3 4 5
 - E. 8 5 3 2 2 3 2
 - F. 8 3 2 5 2 3 2
 - G. None of the other options is the correct answer
- (7) What is the sequence of values printed by the `display` function when the following Source R2L program is evaluated?

```
const D = display;  
  
function fun(x, y, z) {  
  return D(z) * D(y) + D(x); // z * y + x  
}  
  
fun( D(2), D(3), D(4) );  
// equivalent to fun(2, 3, 4);
```

- A. 4 3 2 3 4 2
- B. 4 3 2 2 4 3
- C. 4 3 2 2 3 4
- D. 4 3 2 4 3 2
- E. 2 3 4 3 4 2
- F. 2 3 4 4 3 2
- G. None of the other options is the correct answer

- (8) What is the sequence of values printed by the `display` function when the following Source R2L program is evaluated?

```
const D = display;

function fun(x, y) {
  return y + x;
}

D( fun( D( D(2) * D(3) ),
        D( D(4) + D(5) )
      )
  );
// equivalent to fun(2 * 3, 4 + 5);
```

- A. 2 3 6 4 5 9 15
 - B. 4 5 9 2 3 6 15
 - C. 15 6 2 3 9 4 5
 - D. 3 2 6 5 4 9 15
 - E. 5 4 3 2 9 6 15
 - F. 5 4 9 3 2 6 15
 - G. None of the other options is the correct answer
- (9) What is the sequence of values printed by the `display` function when the following Source R2L program is evaluated?

```
function D(m, x) {
  display(m);
  return x;
}

const fun = D( "4", () => D( "3", D("1", 1) + D("2", 2) ) );
// equivalent to const fun = () => 1 + 2;

D("5", 5);
D("6", fun());
```

- A. "2" "1" "3" "4" "5" "6"
- B. "4" "5" "2" "1" "3" "6"
- C. "2" "1" "3" "4" "5" "2" "1" "3" "6"
- D. "2" "1" "3" "4" "5" "6" "2" "1" "3"
- E. "4" "3" "5" "2" "1" "6"
- F. "4" "5" "1" "2" "3" "6"
- G. None of the other options is the correct answer

(10) In Source R2L, the binary logical operators **&&** and **||** are implemented as follows:

The expression

expressionL && expressionR

is equivalent to

expressionR ? expressionL : false

Similarly, the expression

expressionL || expressionR

is equivalent to

expressionR ? true : expressionL

What is the sequence of values printed by the `display` function when the following Source R2L program is evaluated?

```
function D(m, x) {  
    display(m);  
    return x;  
}  
  
function fun(p, q, r, s) {  
    return ( D("P", p) && D("Q", q) ) && ( D("R", r) || D("S", s) );  
    // equivalent to return (p && q) && (r || s);  
}  
  
fun(true, false, false, true);
```

- A. "S" "R" "Q" "P"
- B. "R" "S" "P" "Q"
- C. "P" "Q" "R" "S"
- D. "P" "Q"
- E. "S" "Q"
- F. "S" "R" "Q"
- G. None of the other options is the correct answer

Section C

- (11) What kind of process does the following function give rise to for any *integer* argument $n > 1$?

```
function fun(n) {  
    return n < 1  
        ? 1  
        : fun(n / 2) + fun(n / 4);  
}
```

- A. An iterative process
 - B. A recursive process
 - C. A process that is neither iterative nor recursive
 - D. A substitution process
 - E. An infinite process
- (12) What kind of process does the following function give rise to for any *integer* arguments $a > 0$ and $b > 0$?

```
function fun(a, b) {  
    return a <= b ? a + b : fun(a - 1, b);  
}
```

- A. An iterative process
- B. A recursive process
- C. A process that is neither iterative nor recursive
- D. A substitution process
- E. An infinite process

- (13) What kind of process does the following function give rise to for any *integer* argument $x > 1$?

```
function fun(x) {  
    return x <= 1  
        ? 1  
        : x === 2  
        ? fun(x - 1)  
        : x / fun(x - 1);  
}
```

- A. An iterative process
- B. A recursive process
- C. A process that is neither iterative nor recursive
- D. A substitution process
- E. An infinite process

- (14) What kind of process does the following function give rise to for any *integer* arguments $s > 0$ and $t > 0$?

```
function fun(s, t) {  
    return s === 0  
        ? 0  
        : s < t  
        ? 1 + fun(s - 1, t)  
        : fun(s - 1, t);  
}
```

- A. An iterative process
- B. A recursive process
- C. A process that is neither iterative nor recursive
- D. A substitution process
- E. An infinite process

(15) What kind of process does the following function give rise to for any *integer* argument $x > 0$?

```
function hoo(x, fun) {  
  return x === 0  
    ? x  
    : x % 2 === 0  
      ? fun(hoo(x - 1, fun))  
      : hoo(x - 1, fun);  
}
```

- A. An iterative process
- B. A recursive process
- C. A process that is neither iterative nor recursive
- D. A substitution process
- E. An infinite process

Section D

- (16) We specify that function *AA*, when applied to any two *integer* arguments, should return the sum of all integers *between* the two arguments, *inclusive of* the two arguments. Consider the following implementation:

```
function AA(start, end) {  
    return start > end ? 0 : start + AA(start + 1, end);  
}
```

Which one of the following statements is correct?

- A. The function *AA* meets the specification.
 - B. The function *AA* does not meet the specification because it can be applied to non-integer arguments.
 - C. The function *AA* does not meet the specification because it will only return the correct result when `start <= end`.
 - D. The function *AA* does not meet the specification because it is an inefficient way to compute the result.
 - E. The function *AA* does not meet the specification because it will only return the correct result when `start > end`.
- (17) We specify that function *BB*, when applied to any three *number* arguments, should return `true` if *exactly one* of the three arguments is negative and return `false` otherwise. Consider the following implementation:

```
function BB(x, y, z) {  
    return x * y * z > 0  
        ? false  
        : x * y > 0 && x * z > 0  
        ? false  
        : true;  
}
```

Which one of the following statements is correct?

- A. The function *BB* meets the specification.
- B. The function *BB* does not meet the specification because it can be applied to non-integer arguments.
- C. The function *BB* does not meet the specification because there is a syntax error.
- D. The function *BB* does not meet the specification because it is an inefficient way to compute the result.
- E. The function *BB* does not meet the specification because it does not work correctly for some valid inputs.

- (18) We specify that function *CC*, when applied to two *positive integer* arguments should return the greatest common divisor of the two arguments. Consider the following implementation:

```
function CC(x, y) {  
    function helper(small, large) {  
        return large % small === 0  
            ? small  
            : helper(small - 1, large);  
    }  
    return x > y ? helper(y, x) : helper(x, y);  
}
```

Which one of the following statements is correct?

- A. The function *CC* meets the specification.
- B. The function *CC* does not meet the specification because it can be applied to non-integer arguments.
- C. The function *CC* does not meet the specification because it will lead to an infinite loop.
- D. The function *CC* does not meet the specification because it is an inefficient way to compute the result.
- E. The function *CC* does not meet the specification because it does not work correctly for some valid inputs.

- (19) We specify that function *DD*, when applied to a *positive integer* argument *n*, should return a function, which when applied to a *positive integer* argument *x*, returns the result $\text{floor}(x / n)$, where, given any number *y*, $\text{floor}(y)$ is the greatest integer number that is less than or equal to *y*.

Consider the following implementation:

```
function E(n) {  
  function helper(x) {  
    return x < n ? 0 : 1 + helper(x - n);  
  }  
  return x => helper(x);  
}  
const DD = n => E(n);
```

Which one of the following statements is correct?

- A. The function *DD* meets the specification.
- B. The function *DD* does not meet the specification because one of the lambda expressions cannot be evaluated.
- C. The function *DD* does not meet the specification because the *helper* function does not check if the argument *x* is greater than 0.
- D. The function *DD* does not meet the specification because it is an inefficient way to compute the result.
- E. The function *DD* does not meet the specification because it does not work correctly for some valid arguments.

————— **END OF QUESTIONS** —————