NATIONAL UNIVERSITY OF SINGAPORE

1

CS1101S — PROGRAMMING METHODOLOGY

(AY2018/2019 SEMESTER 1)

READING ASSESSMENT 2, REDACTED EDITION OF 2020/21

Time Allowed: **45 Minutes**

INSTRUCTIONS

1. This question paper comprises **NINE (9)** printed pages, including this page.

2. You are also provided with **one OCR Form** to write your answers.

3. Clearly **write** and **shade** your **STUDENT NUMBER** on your **OCR Form** using a **2B PENCIL**.

4. There are **17** multiple-choice questions. Each question has one correct answer. **1 mark** is awarded for each correct answer and there is no penalty for a wrong answer.

5. The full score is **17 marks**.

6. Answer **ALL** questions.

7. Use only a **2B PENCIL** to **shade** your answers on your **OCR Form**.

8. This is a **CLOSED BOOK** assessment, but you are allowed to bring in one A4 sheet of notes (handwritten or printed on both sides).

9. **Submit only the OCR Form**.

(1) What is the single-digit **number** at the **top-right corner** on the **front page** of this question paper? (**Important**: Please make sure your answer is correct because it determines how we mark your answers to all the subsequent questions.)

A. 1
B. 2
C. 3
D. 4
E. 5

Consider the following Source program for the next 3 questions:

Program A:

```
x => x + 1;

function funA(n) {
    return n <= 1 ? n : funA(n - 1) + funA(n - 2);
}
const aa = funA(4);
```

(2) How many bindings appear in the program environment frame?

A. 0
B. 1
C. 2
D. 3
E. More than 3

(3) How many environment frames get created during the evaluation of Program A? (Do not count the global environment frame.)

A. 0
B. 1
C. 4
D. 10
E. None of the above

(4) Of the environment frames that get created during the evaluation of Program A, how many extend the program environment *directly*?

A. 0
B. 1
C. 4
D. 9
E. None of the above

(5) How many environment frames get created during the evaluation of the following program? (Do not count the global environment frame.)

```
function funB(n) {
    if (n <= 1) {
        return n;
    } else {
        let xx = funB(n - 1);
        let yy = funB(n - 2);
        return xx + yy;
    }
}
funB(4);
```

A. 14
B. 9
C. 4
D. 1
E. None of the above

(6) What is the result of evaluating the following program?

```
let xx = 0;
let yy = 0;

function funC(n) {
    if (n <= 1) {
        return n;
    } else {
        xx = funC(n - 1);
        yy = funC(n - 2);
        return xx + yy;
    }
}
funC(12);
```

A. 1
B. 6
C. 21
D. 144
E. None of the above

Consider the following Source program for the next 4 questions:

Program D:

```
function ff(xx) {
    function gg(yy) {
        return xx * yy;
    }
    return gg;
}
const aa = ff(1)(6);
const bb = ff(2)(5);
ff(aa)(bb);
```

(7) How many function objects get created during the evaluation of Program D?

A. 6
B. 4
C. 3
D. 2
E. 1

(8) How many environment frames get created during the evaluation of Program D? (Do not count the global environment frame.)

A. 1
B. 3
C. 7
D. 9
E. None of the above

(9) Of the environment frames that get created during the evaluation of Program D, how many extend the program environment *directly*?

A. 1
B. 3
C. 6
D. 9
E. None of the above

(10) During the evaluation of Program D, some of the environment frames contain the name `xx`. Which other name do these frames also contain?

A. `gg`
B. `ff`
C. `yy`
D. `aa`
E. None of the above

Consider the following Source program for the next 4 questions:

Program E:

```
function dest_map(fun, xs) {
    if (! is_null(xs)) {
        const h = head(xs);
        set_head(xs, fun(h));
        dest_map(fun, tail(xs));
    } else { }
}
const L = list(1, 2, 3);
dest_map(x => y => x + y, L);
```

(11) How many function objects get created during the evaluation of Program E?

- **A.** 1
- **B.** 2
- **C.** 4
- **D.** 5
- **E.** None of the above

(12) How many environment frames get created during the evaluation of Program E? (Do not count the global environment frame. We assume that the application of a primitive function does not create any frame. See Appendix for a list of primitive functions.)

- **A.** 7
- **B.** 3
- **C.** 11
- **D.** 4
- **E.** None of the above

(13) Of the environment frames that get created during the evaluation of Program E, how many extend the program environment *directly*?

- **A.** 7
- **B.** 3
- **C.** 2
- **D.** 4
- **E.** None of the above

(14) Which of the following statements, when added to the end of Program E, will change the value of `L` back to `list(1, 2, 3)`?

A. `dest_map(x => y => x - y, L);`
B. `dest_map(x => x - y, L);`
C. `dest_map(x => x(0), L);`
D. `dest_map(x => y => x(-y), L);`
E. None of the above

Consider the following Source program for the next 3 questions:

Program F:

```
function mystery(A) {
    const len = array_length(A);
    let i = len - 1;
    while (i >= 1) {
        let j = 1;
        while (j <= i) {
            const temp = A[j - 1];
            if (A[j - 1] > A[j]) {
                A[j - 1] = A[j];
                A[j] = temp;
            } else { }
            j = j + 1;
        }
        i = i - 1;
    }
}
const aa = [7, 3, 10, 4, 9, 8, 1, 5, 2, 6];
mystery(aa);
aa;
```

(15) How many environment frames get created during the evaluation of Program F? (Do not count the global environment frame. We assume that the application of a primitive function does not create any frame. See Appendix for a list of primitive functions.)

A. 91
B. 56
C. 46
D. 19
E. None of the above

(16) Of the environment frames that get created during the evaluation of Program F, how many are created for the *inner* **`while`**-loop in the function `mystery`?

A. 0
B. 9
C. 45
D. 81
E. None of the above

(17) What is the value of `aa` at the end of the evaluation of Program F?

A. `[1, 3, 5, 7, 9, 10, 2, 4, 6, 8]`
B. `[3, 7, 4, 9, 8, 1, 5, 2, 6, 10]`
C. `[7, 7, 7, 7, 7, 7, 7, 7, 7, 7]`
D. `[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]`
E. `[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]`

——— END OF QUESTIONS ———

Appendix

Primitive Functions

The following are some of the primitive functions in Source §3:

- `display(a)`
- `pair(x, y)`
- `is_pair(x)`
- `head(x)`
- `tail(x)`
- `is_null(xs)`
- `list(x1, x2,..., xn)`
- `set_head(p, x)`
- `set_tail(p, x)`
- `array_length(x)`

Pre-declared Functions

Some of the pre-declared functions in Source §3 are declared as follows:

```
function length(xs) {
    return is_null(xs)
        ? 0
        : 1 + length(tail(xs));
}

function map(f, xs) {
    return is_null(xs)
        ? xs
        : pair(f(head(xs)), map(f, tail(xs)));
}

function filter(pred, xs) {
    return is_null(xs)
        ? xs
        : pred(head(xs))
            ? pair(head(xs), filter(pred, tail(xs)))
            : filter(pred, tail(xs));
}

function accumulate(op, initial, xs) {
    return is_null(xs)
        ? initial
        : op(head(xs), accumulate(op, initial, tail(xs)));
}
```

(blank page)