NATIONAL UNIVERSITY OF SINGAPORE

CS1101S — PROGRAMMING METHODOLOGY

(AY2020/2021 SEMESTER 1)

READING ASSESSMENT 2

Time Allowed: **45 Minutes**

ANSWERS

INSTRUCTIONS

1. This assessment contains **20 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 **20 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. You are allowed access to these online reference pages:
   - *Source §3 pre-declared constants and functions* at `https://source-academy.github.io/source/source_3/global.html`
   - *Specification of Source §3* at `https://source-academy.github.io/source/source_3.pdf`

8. **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 program?

```
function ff(x) {
    x = x + 1;
    return x;
}

function gg(y) {
    yy = y + 1;
    return y;
}

let xx = 1;
let yy = 1;
const k = 1000 * ff(xx) + 100 * gg(yy);
k + 10 * xx + yy;
```

A. 2111
B. 2112 **(answer)**
C. 2121
D. 2122
E. 2211
F. 2212
G. 2221
H. 2222

(2) What is the result (in *list notation*) of evaluating the following Source program?

```
const A = list(list(2), list(4));
const B = map(x => x, A);
set_head(head(B), 3);
set_head(tail(B), 5);
A;
```

A. list(list(2), list(4))
B. list(list(2), list(5))
C. list(list(3), list(4)) **(answer)**
D. list(list(3), list(5))
E. list(list(3), 5)
F. list(3, list(5))
G. list(3, 5)

Section B

For all the questions in this section, consider the following Source program:

Program X:

```
function ff(A, v) {
    let low = 0;
    let high = array_length(A) - 1;

    while (low <= high) {
        const mid = math_floor((low + high) / 2 );
        if (v === A[mid]) {
            break;
        } else if (v < A[mid]) {
            high = mid - 1;
        } else {
            low = mid + 1;
        }
    }
    return (low <= high);
}

const A = [1,2,3,4,5,6,7];
ff(A, 5);
```

(3) How many bindings does the program environment frame contain?

A. 0
B. 1
C. 2 **(answer)**
D. 4
E. 5
F. 6
G. 7

(4) How many environment frames get created during the evaluation of Program X? (Do not count the global environment frame. We assume that the application of the primitive functions `array_length` and `math_floor` do not create any frame.)

A. 3
B. 4
C. 5
D. 6 **(answer)**
E. 8
F. 9

(5) Of the environment frames that get created during the evaluation of Program X, how many extend the program environment *directly*? (Do not count the global environment frame. We assume that the application of the primitive functions `array_length` and `math_floor` do not create any frame.)

A. 1 **(answer)**
B. 2
C. 3
D. 4
E. 5
F. 6

(6) Of the environment frames that get created during the evaluation of Program X, how many *directly* extend the environment whose innermost frame contains the bindings of variables `low` and `high`? (Do not count the global environment frame. We assume that the application of the primitive functions `array_length` and `math_floor` do not create any frame.)

A. 0
B. 1
C. 2
D. 3 **(answer)**
E. 4
F. 5
G. 6

(7) What are the final values of the variables `low` and `high` at the end of the evaluation of Program X?

A. `low`: 2 and `high`: 6
B. `low`: 3 and `high`: 6
C. `low`: 3 and `high`: 5
D. `low`: 4 and `high`: 5
E. `low`: 4 and `high`: 4 **(answer)**
F. `low`: 5 and `high`: 4

Section C

For all the questions in this section, consider the following Source program:

Program Y:

```
let y = 1;

function g(f) {
    y = y + 1;
    return x => 10 * f() + x;
}

const h = g(() => y + 1);
y = y + 1;
h(y);
```

(8) What is the result of evaluating Program Y?

A. 12
B. 13
C. 22
D. 23
E. 32
F. 33
G. 43 **(answer)**

(9) How many bindings does the program environment frame contain?

A. 0
B. 1
C. 2
D. 3 **(answer)**
E. 4
F. 5

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

A. 1
B. 2
C. 3 **(answer)**
D. 4
E. 5
F. 6

(11) Of the environment frames that get created during the evaluation of Program Y, how many extend the program environment *directly*? (Do not count the global environment frame.)

A. 1 **(answer)**
B. 2
C. 3
D. 4
E. 5
F. 6

(12) How many function objects get created during the evaluation of Program Y? (Do not count function objects of primitive and pre-declared functions such as `array_length`, `math_floor`, `pair`, `head`, `tail`, `length`, and `map`.)

A. 0
B. 1
C. 2
D. 3 **(answer)**
E. 4
F. 5

(13) Of the function objects that get created during the evaluation of Program Y, how many have the program environment as their environment (i.e. their right circles point to the program frame)? (Do not count function objects of primitive and pre-declared functions such as `array_length`, `math_floor`, `pair`, `head`, `tail`, `length`, and `map`.)

A. 0
B. 1
C. 2 **(answer)**
D. 3
E. 4
F. 5

Section D

For all the questions in this section, consider the following Source program:

Program Z:

```
let gg = x => (x % 2 === 0);

function fun(ff, xs) {
    gg = x => (x % 2 === 0);

    if (is_null(xs)) {
        return xs;
    } else if (gg(head(xs))) {
        const new_x = ff(head(xs));
        return pair(new_x, fun(ff, tail(xs)));
    } else {
        set_tail(xs, fun(ff, tail(xs)));
        return xs;
    }
}

const L = list(2, 3, 4);
const R = fun(x => 2 * x, L);
```

(14) What is the value of `R` (in *list notation*) at the end of the evaluation of Program Z?

A. `list(2, 3)`
B. `list(4, 3)`
C. `list(2, 3, 4)`
D. `list(4, 3, 4)`
E. `list(2, 3, 8)`
F. `list(4, 3, 8)` **(answer)**

(15) What is the value of `L` (in *list notation*) at the end of the evaluation of Program Z?

A. `list(2, 3)`
B. `list(4, 3)`
C. `list(2, 3, 4)`
D. `list(4, 3, 4)`
E. `list(2, 3, 8)` **(answer)**
F. `list(4, 3, 8)`

(16) How many bindings does the program environment frame contain?

A. 3
B. 4 **(answer)**
C. 5
D. 6
E. 7
F. 8

(17) How many environment frames get created during the evaluation of Program Z? (Do not count the global environment frame. We assume that the application of the primitive functions `is_null`, `head`, `tail`, `pair`, `list`, and `set_tail` do not create any frame.)

A. 5
B. 7
C. 8
D. 9
E. 10
F. 12 **(answer)**
G. 14

(18) Of the environment frames that get created during the evaluation of Program Z, how many extend the program environment *directly*? (Do not count the global environment frame. We assume that the application of the primitive functions `is_null`, `head`, `tail`, `pair`, `list`, and `set_tail` do not create any frame.)

A. 1
B. 3
C. 4
D. 6 **(answer)**
E. 8
F. 9
G. 10

(19) How many function objects get created during the evaluation of Program Z? (Do not count function objects of primitive and pre-declared functions such as `math_floor`, `pair`, `head`, `tail`, `length`, and `map`.)

A. 2
B. 3
C. 4
D. 6
E. 7 **(answer)**
F. 9
G. 10

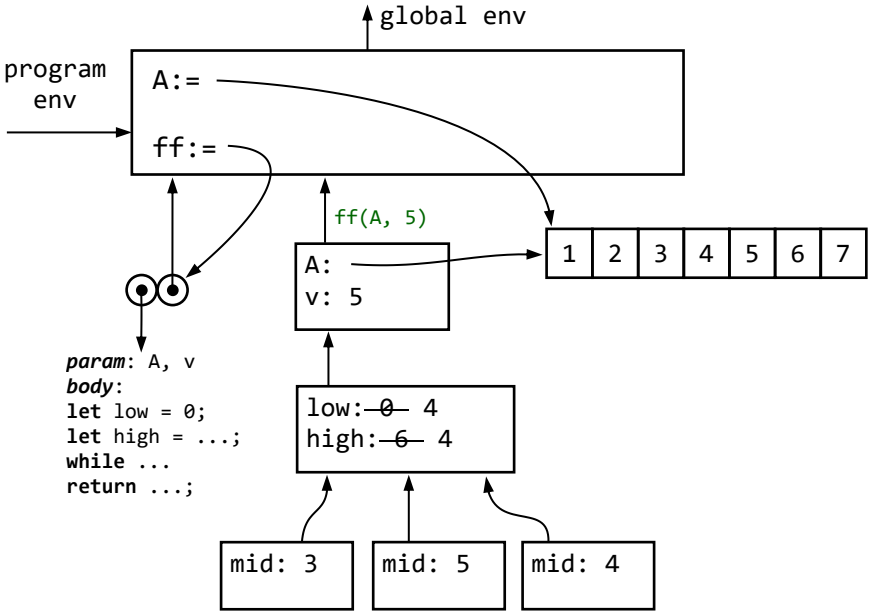
(20) Of the function objects that get created during the evaluation of Program Z, how many have the program environment as their environment (i.e. their right circles point to the program frame)? (Do not count function objects of primitive and pre-declared functions such as `math_floor`, `pair`, `head`, `tail`, `length`, and `map`.)

A. 1
B. 2
C. 3 **(answer)**
D. 6
E. 7
F. 8
G. 9

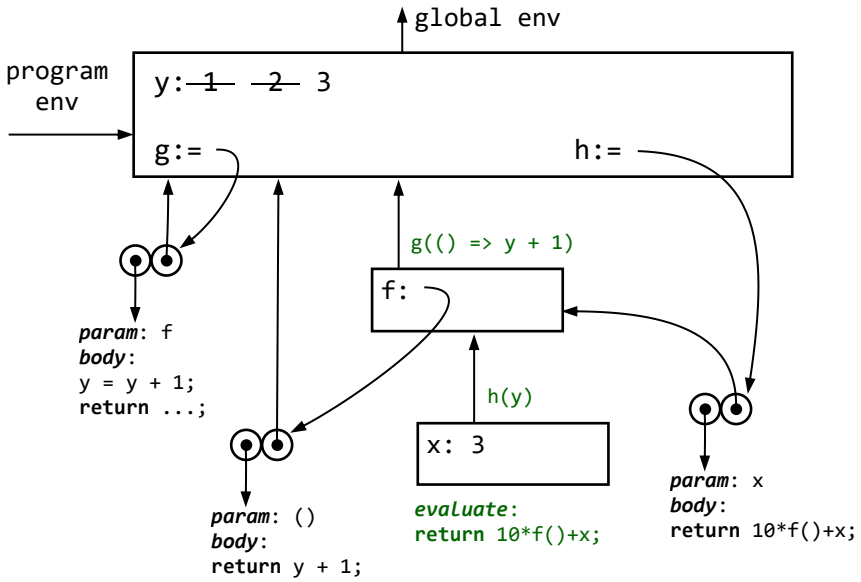
END OF QUESTIONS

(blank page)

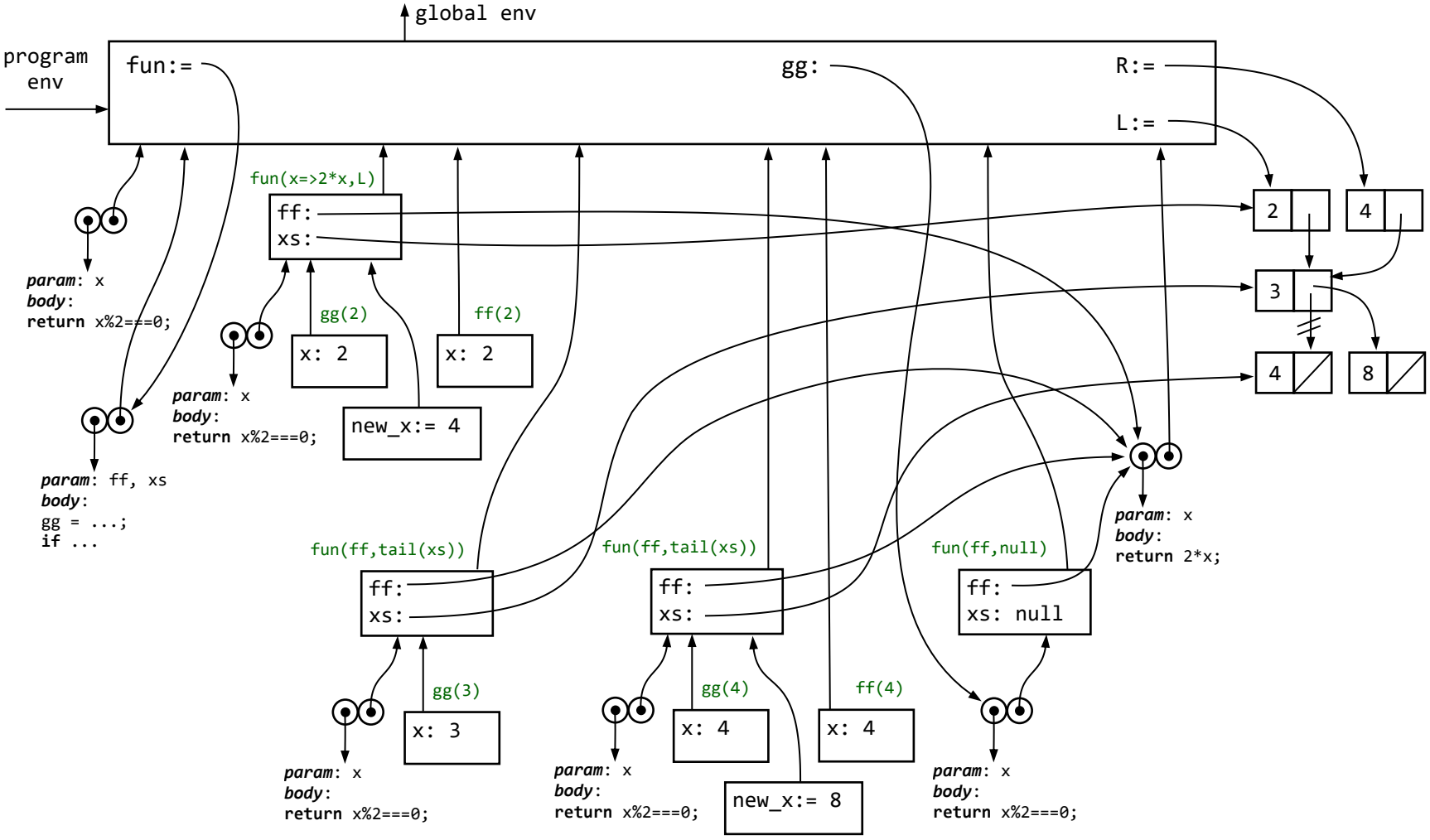
Program X:


global env
program
env
A:=
ff:=
ff(A, 5)
A:
v: 5
1
2
3
4
5
6
7
param: A, v
body:
let low = 0;
let high = ...;
while ...
return ...;
low: 0 4
high: 6 4
mid: 3
mid: 5
mid: 4

Program Y:


global env
program
env
y: ~~1~~ ~~2~~ 3
g:=
h:=
g(() => y + 1)
f:
param: f
body:
y = y + 1;
return ...;
h(y)
x: 3
param: ()
body:
return y + 1;
evaluate:
return 10*f()+x;
param: x
body:
return 10*f()+x;

Program Z:


global env
program env
fun:=
gg:
R:=
L:=
fun(x=>2*x,L)
ff:
xs:
param: x
body:
return x%2===0;
gg(2)
ff(2)
x: 2
x: 2
new_x:= 4
param: ff, xs
body:
gg = ...;
if ...
fun(ff,tail(xs))
fun(ff,tail(xs))
fun(ff,null)
xs: null
gg(3)
gg(4)
ff(4)
x: 3
x: 4
x: 4
new_x:= 8
2
4
3
4
8