Solved.

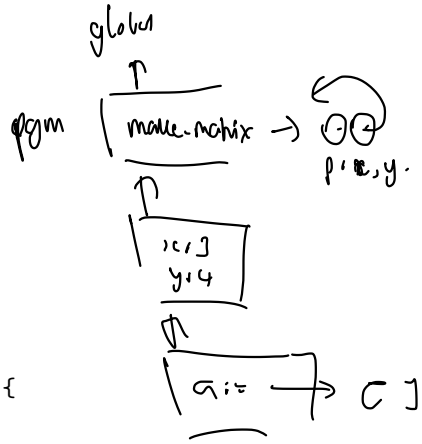
CS1101S Programming Methodology

Reading Assessment 2, redacted version of 2020/21

Use **only** the given answer sheet to indicate your answer to each of the following 13 questions. Use a pencil, and mark only one choice for each question. Do not write your name, but only your student number, on the answer sheet.

The paper ends on page 7. Pages 8 and 9 are empty. Feel free to tear off these sheets and use them as scratch paper.

Note that in this redacted version, several questions are missing, compared to the original, because they are obsolete in the context of the 2019/20 material.



Part A

Consider the following Source program:

```
function make_matrix(x, y) {
    let a = [];
    for (let i = 0; i < x; i = i + 1) {
        a[i] = [];
        for (let j = 0; j < y; j = j + 1) {
            a[i][j] = 0;
        }
    }
    return a;
}

make_matrix(3,4);
```

Question 1: How many arrays get created, when this program runs?

1 [A] 1

1 [B] 4

1 [C] 12

1 [D] 81

1 [E] none of the above

Question 2: How many array assignments (assigning a value at a particular index in an array) are carried out during the evaluation of the program above?

2 [A] 3

2 [B] 4

2 [C] 12

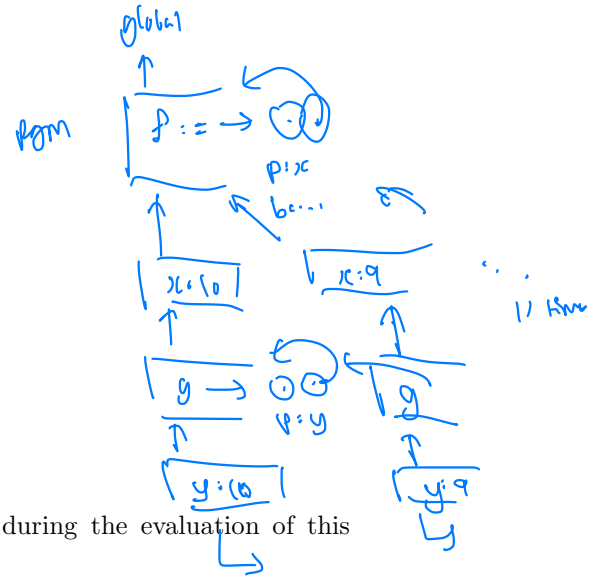
2 [D] 64

2 [E] none of the above

Part B

Consider the following Source program:

```
function f(x) {
    function g(y) {
        return (y === 0) ? x : f(y - 1);
    }
    return g(x);
}
f(10);
```



Question 3: How many environment frames get created during the evaluation of this program, excluding the global frame?

3 [A] 4

3 [B] 12

3 [C] 23

3 [D] 34

3 [E] none of the above

Question 4: How many function objects get created during the evaluation of this program?

4 [A] 2

4 [B] 12

4 [C] 23

4 [D] 34

4 [E] none of the above

Question 5: Some of the environment frames will contain the variable `g`. Which variable do these frame also contain?

5 [A] `x`

5 [B] `y`

5 [C] `f`

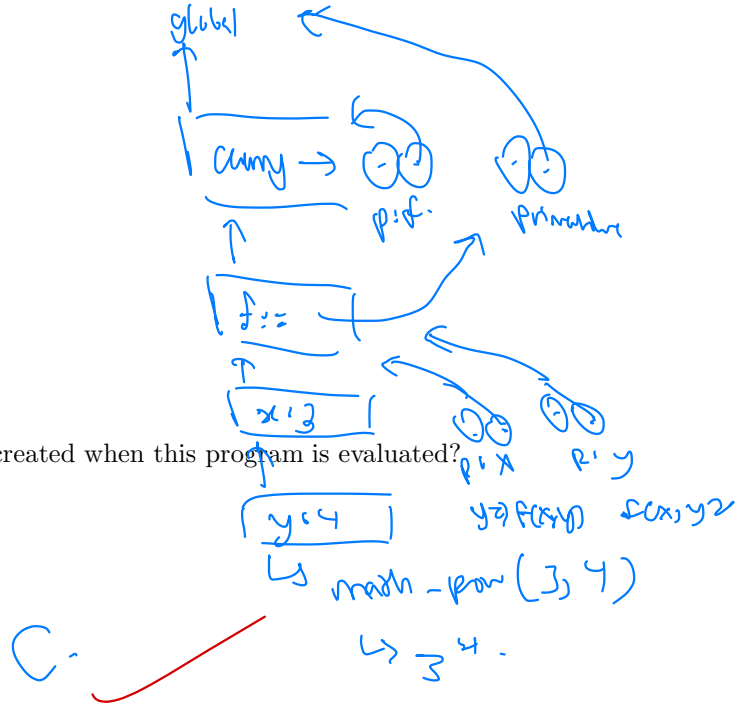
5 [D] `display`

5 [E] none of the above

Part C

Consider the following program.

```
function curry(f) {
    return x => y => f(x, y);
}
curry(math_pow)(3)(4);
```



Question 6: How many function objects get created when this program is evaluated?

6 [A] 0

6 [B] 1

6 [C] 3

6 [D] 5

6 [E] none of the above

Question 7: One of the environment frames that get created will contain the variable `f`. Which variable does this frame also contain?

7 [A] `x`

7 [B] `y`

7 [C] `curry`

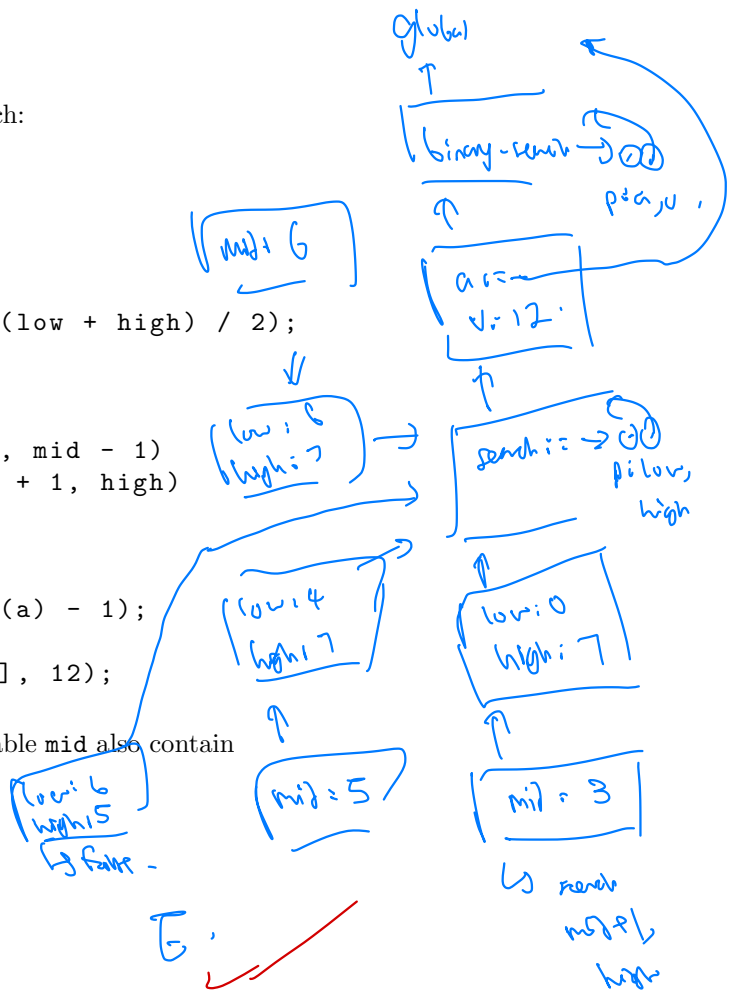
7 [D] `math_pow`

7 [E] none of the above

Part D

Consider the following program for binary search:

```
function binary_search(a, v) {
    function search(low, high) {
        if (low > high) {
            return false;
        } else {
            let mid = math_floor((low + high) / 2);
            return (v === a[mid])
                   ||
                   ( (v < a[mid])
                     ? search(low, mid - 1)
                     : search(mid + 1, high)
                   );
        }
    }
    return search(0, array_length(a) - 1);
}
binary_search([1,3,5,7,9,11,13,15], 12);
```



Question 8: The frames that contain the variable `mid` also contain

8 [A] the variables `low, high`

8 [B] the variables `a, v`

8 [C] the variable `binary_search`

8 [D] the variable `search`

8 [E] none of the above

Question 9: How many times will the function `search` be called during the evaluation of this program?

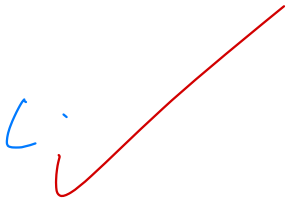
9 [A] twice

9 [B] 3 times

9 [C] 4 times

9 [D] 5 times

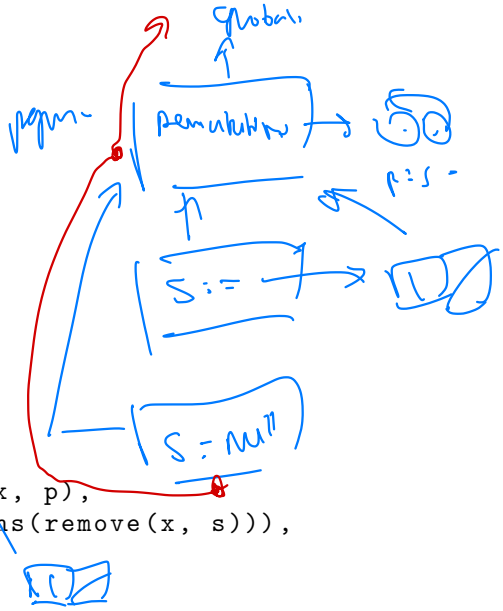
9 [E] none of the above



Part E

Consider the following program from the lectures.

```
function permutations(s) {
    return is_null(s)
        ? list(null)
        : accumulate(
                  append, null,
                  map(x =>
                        map(p => pair(x, p),
                            permutations(remove(x, s))),
                      s));
}
permutations(list(1));
```



Question 10: According to the environment model, the evaluation of a variable requires a lookup of the variable in the innermost frame of the current environment. If the lookup fails, the innermost frame of the *enclosing* environment is used, and so on until the global environment is reached. When evaluation reaches the expression `pair(x, p)` during the evaluation of the program, the variable `pair` is looked up in frame(s). How many frame lookups will fail, before the value of `pair` is found in the global frame?

10 [A] 1

10 [B] 2

10 [C] 3

10 [D] 4

10 [E] none of the above

Question 11: How many function objects get created during the evaluation of the program?

11 [A] 1

11 [B] 3

11 [C] 5

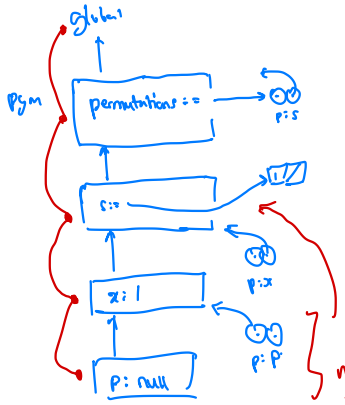
11 [D] 6

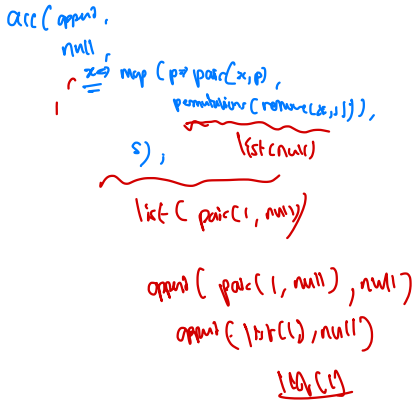
11 [E] none of the above

— END OF QUESTIONS —

Scratch Paper

```
function permutations(s) {
    return is_null(s)
        ? list(null)
        : accumulate(
                  append, null,
                  map(x =>
                        map(p => pair(x, p),
                            permutations(remove(x, s))),
                      s));
}
permutations(list(1));
```





∴ 4 failed frames.

not sure, shouldn't this be attached to the prev frame?

Scratch Paper

CS1101S Programming Methodology

Reading Assessment 2 Solution, redacted version of 2020/21

Use **only** the given answer sheet to indicate your answer to each of the following 13 questions. Use a pencil, and mark only one choice for each question. Do not write your name, but only your student number, on the answer sheet.

The paper ends on page 7. Pages 8 and 9 are empty. Feel free to tear off these sheets and use them as scratch paper.

Note that in this redacted version, several questions are missing, compared to the original, because they are obsolete in the context of the 2019/20 material.

Part A

Consider the following Source program:

```
function make_matrix(x, y) {
    let a = [];
    for (let i = 0; i < x; i = i + 1) {
        a[i] = [];
        for (let j = 0; j < y; j = j + 1) {
            a[i][j] = 0;
        }
    }
    return a;
}

make_matrix(3,4);
```

Question 1: How many arrays get created, when this program runs?

1 [A] 1

1 [B] 4

1 [C] 12

1 [D] 81

1 [E] none of the above

Answer: B: The matrix, and its three rows.

Question 2: How many array assignments (assigning a value at a particular index in an array) are carried out during the evaluation of the program above?

2 [A] 3

2 [B] 4

2 [C] 12

2 [D] 64

2 [E] none of the above

Answer: E: 15 (12 array assignments in the inner loop and 3 assignments in the outer loop)

Part B

Consider the following Source program:

```
function f(x) {
    function g(y) {
        return (y === 0) ? x : f(y - 1);
    }
    return g(x);
}
f(10);
```

Question 3: How many environment frames get created during the evaluation of this program, excluding the global frame?

3 [A] 4

3 [B] 12

3 [C] 23

3 [D] 34

3 [E] none of the above

Answer: D: The correct number of environment frames for this program is 34.

Question 4: How many function objects get created during the evaluation of this program?

4 [A] 2

4 [B] 12

4 [C] 23

4 [D] 34

4 [E] none of the above

Answer: B: 12 (11 function objects for function `g` and and one for function `f`)

Question 5: Some of the environment frames will contain the variable `g`. Which variable do these frame also contain?

5 [A] `x`

5 [B] `y`

5 [C] `f`

5 [D] `display`

5 [E] none of the above

Answer: E: none of the above

Part C

Consider the following program.

```
function curry(f) {
    return x => y => f(x, y);
}
curry(math_pow)(3)(4);
```

Question 6: How many function objects get created when this program is evaluated?

6 [A] 0

6 [B] 1

6 [C] 3

6 [D] 5

6 [E] none of the above

Answer: C: 3

Question 7: One of the environment frames that get created will contain the variable `f`. Which variable does this frame also contain?

7 [A] `x`

7 [B] `y`

7 [C] `curry`

7 [D] `math_pow`

7 [E] none of the above

Answer: E: That frame only has variable `f`

Part D

Consider the following program for binary search:

```
function binary_search(a, v) {
    function search(low, high) {
        if (low > high) {
            return false;
        } else {
            let mid = math_floor((low + high) / 2);
            return (v === a[mid])
                   ||
                   ( (v < a[mid])
                     ? search(low, mid - 1)
                     : search(mid + 1, high)
                   );
        }
    }
    return search(0, array_length(a) - 1);
}
binary_search([1,3,5,7,9,11,13,15], 12);
```

Question 8: The frames that contain the variable `mid` also contain

8 [A] the variables `low, high`

8 [B] the variables `a, v`

8 [C] the variable `binary_search`

8 [D] the variable `search`

8 [E] none of the above

Answer: E: There is a frame for just the alternative statement of the conditional.

Question 9: How many times will the function `search` be called during the evaluation of this program?

9 [A] twice

9 [B] 3 times

9 [C] 4 times

9 [D] 5 times

9 [E] none of the above

Answer: C: 4 times (arguments: 0,7; 4,7; 6,7; 6,5)

Part E

Consider the following program from the lectures.

```
function permutations(s) {
    return is_null(s)
        ? list(null)
        : accumulate(
                    append, null,
                    map(x =>
                          map(p => pair(x, p),
                              permutations(remove(x, s))),
                        s));
}
permutations(list(1));
```

Question 10: According to the environment model, the evaluation of a variable requires a lookup of the variable in the innermost frame of the current environment. If the lookup fails, the innermost frame of the *enclosing* environment is used, and so on until the global environment is reached. When evaluation reaches the expression `pair(x, p)` during the evaluation of the program, the variable `pair` is looked up in frame(s). How many frame lookups will fail, before the value of `pair` is found in the global frame?

10 [A] 1

10 [B] 2

10 [C] 3

10 [D] 4

10 [E] none of the above

Answer: D: 4 (one frame with `p`, one with `x`, one with `s`, and one with `permutations`)

Question 11: How many function objects get created during the evaluation of the program?

11 [A] 1

11 [B] 3

11 [C] 5

11 [D] 6

11 [E] none of the above

Answer: B: 3 (each function expression/statement gives rise to one function object, here)

— END OF QUESTIONS —

Scratch Paper

Scratch Paper