CS 61A Fall 2020

Structure and Interpretation of Computer Programs

Practice Midterm 1

INSTRUCTIONS

This is your exam. Complete it either at exam.cs61a.org or, if that doesn't work, by emailing course staff with your solutions before the exam deadline.

This exam is intended for the student with email address `<EMAILADDRESS>`. If this is not your email address, notify course staff immediately, as each exam is different. Do not distribute this exam PDF even after the exam ends, as some students may be taking the exam in a different time zone.

For questions with **circular bubbles**, you should select exactly *one* choice.

- ◯ You must choose either this option
- ◯ Or this one, but not both!

For questions with **square checkboxes**, you may select *multiple* choices.

- ☐ You could select this choice.
- ☐ You could select this one too!

You may start your exam now. Your exam is due at <DEADLINE> Pacific Time. Go to the next page to begin.

Preliminaries

You can complete and submit these questions before the exam starts.

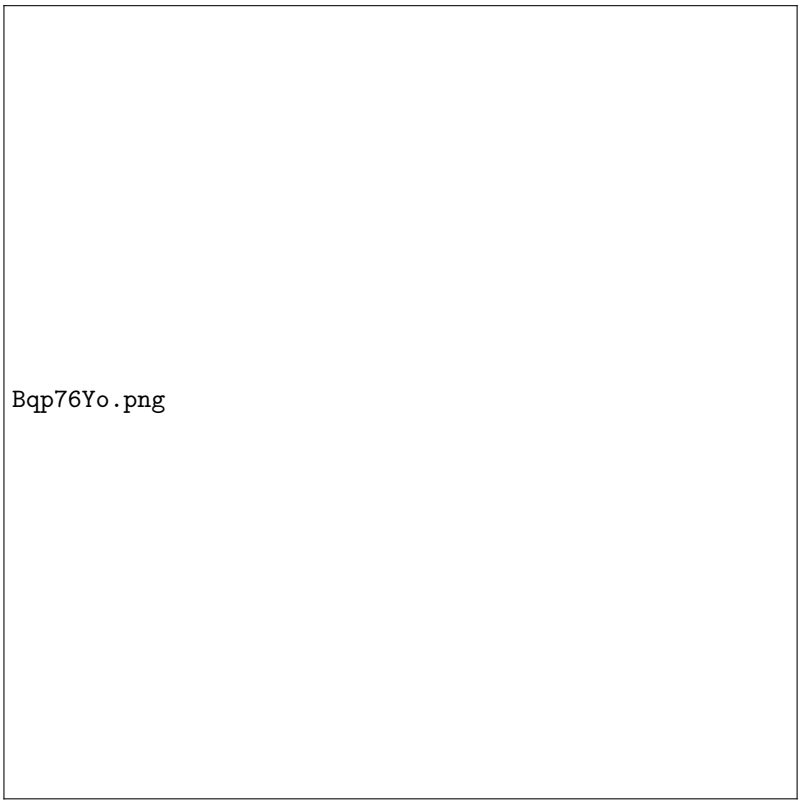
(a) What is your full name?

(b) What is your student ID number?

1. (4.0 points) Alphabet Soup

Fill in each blank in the code example below so that the following environment diagram results from fully executing it.

Click here to open the diagram in a new window.




```
def f(x):

    return lambda y: x + y

def _________(x):
       (a)

    return y(x) + z

x = 1

y = _________
       (b)


---------
   (c)

z = f(8)
```

(a) (1.0 pt) Fill in blank (a).

(b) (2.0 pt) Which of these could fill in blank (b)?

- ◯ `lambda z: z + 2`
- ◯ `lambda y: y + 2`
- ◯ `lambda z: y + 2`
- ◯ `lambda y: z + 2`

(c) (1.0 pt) Fill in blank (c). You **may not** write a `def` statement.

2. (8.0 points) Enrollment

Assume `enrolled` takes a positive integer `n` and returns the number of courses at UC Berkeley that have at least `n` students enrolled. The implementation has been omitted intentionally.

Assume that all courses have at least one student enrolled.

For all functions, you only need to consider arguments that would not cause an error.

```
def enrolled(n):
    """Return the number of courses that have at least n students enrolled.

    >>> enrolled(1)
    6507
    >>> enrolled(10)
    825
    >>> enrolled(500)
    14
    >>> enrolled(1900)
    1
    """
    assert n > 0
    ...
```

(a) (6.0 points)

Select the correct description of each function below.

i. (2.0 pt)

```
def mystery1(n):
    assert n > 0
    return enrolled(n) != enrolled(n+1)
```

`mystery1` returns ...

◯ whether any course has at least `n` students enrolled.

◯ whether any course has at most `n` students enrolled.

◯ whether any course has more than `n` students enrolled.

◯ whether any course has less than `n` students enrolled.

◯ whether any course has exactly `n` students enrolled.

◯ whether any course has exactly `n+1` students enrolled.

◯ whether any course has exactly `n-1` students enrolled.

◯ None of these

ii. (2.0 pt) Clarifications: - "x is at most n" means `x <= n`. - "x is less than n" means `x < n`. - "x is at least n" means `x >= n`. - "x is more than n" means `x > n`.

```
def mystery2(n, m):
    assert n > m
    return enrolled(m) - enrolled(n)
```

`mystery2` returns ...

◯ how many courses have at least `m` but less than `n` students enrolled.

◯ how many courses have more than `m` but at most `n` students enrolled.

◯ how many courses have at least `n` but less than `m` students enrolled.

◯ how many courses have more than `n` but at most `m` students enrolled.

◯ None of these

iii. (2.0 pt)

```
def mystery3(f):
    n = 1
    while True:
        if f(n):
            n = n + 1
        else:
            return n - 1
```

The expression `mystery3(enrolled)` evaluates to ...

◯ a function that computes the total number of courses.

◯ the total number of courses.

◯ a function that computes the enrollment of the largest course.

◯ the enrollment of the largest course.

◯ a function that computes the total enrollment for all courses.

◯ the total enrollment for all courses.

◯ None of these

(b) (2.0 points)

Implement `less`, which takes a positive integer `n` and returns the number of courses that have less than `n` students enrolled.

You may use built-in functions such as `max`, `min`, `pow`, and `abs`.

```
def less(n):

    return _________ - _________
              (a)         (b)
```

i. (1.0 pt) Which of these could fill in blank (a).

- ◯ `enrolled(n)`
- ◯ `enrolled(n+1)`
- ◯ `enrolled(n-1)`
- ◯ `enrolled(1)`

ii. (1.0 pt) Fill in blank (b).

3. (6.0 points) General Artificial Intelligence

Implement `chatbot`, which takes a function `response` and returns a `learn` function. The `learn` function can be taught new responses by calling it or its first return value. The second return value prints responses to messages that it has learned. If the second return value receives a message it has never learned, then the `response` function originally passed to `chatbot` is called.

```
def chatbot(response):
    """A chatbot that responds using the replies it learns.

    >>> siri = chatbot(lambda m: print('What?!?'))
    >>> siri, hey = siri('Weather?', 'smoky')
    >>> siri, hey = siri('Globe?', 'warm')
    >>> siri, hey = siri('Vaccine?', 'not yet')
    >>> siri, hey = siri('Classes?', 'online')
    >>> hey('Classes?')
    online
    >>> hey('Weather?')
    smoky
    >>> hey('Play some Hog?')
    What?!?
    >>> hey('Vaccine?')
    not yet
    """
    def learn(message, reply):

        def chat(m):

            if _________:
                  (a)

                print(_________)
                         (b)

            else:

                _________
                   (c)

        return _________, _________
                  (d)        (e)

    return _________
              (f)
```

(a) (1.0 pt) Fill in blank (a):

(b) (1.0 pt) Fill in blank (b):

(c) (1.0 pt) Fill in blank (c):

(d) (1.0 pt) Which of these could fill in blank (d)?

- ◯ `learn`
- ◯ `chat`
- ◯ `chatbot(learn)`
- ◯ `chatbot(chat)`

(e) (1.0 pt) Which of these could fill in blank (e)?

- ◯ `learn`
- ◯ `chat`
- ◯ `chatbot(learn)`
- ◯ `chatbot(chat)`

(f) (1.0 pt) Fill in blank (f) **with a single name**.

No more questions.