CMSC 330 Fall 2020 Midterm I

Q1 Introduction

0 Points

[omitted]

Q2 Programming Language Concepts

11 Points

Q2.1 Ruby Objects

4 Points

In Ruby, which of the following is an object? Check all that apply.

- A string
- An integer
- `nil`
- A class

Q2.2 Ruby Arrays vs Hashes

3 Points

Briefly describe one difference between Arrays and Hashes in Ruby. Which would be more suitable for storing a sorted collection of values?

Q2.3 OCaml Currying

4 Points

Write an OCaml function called `foo` which is equivalent to the following, but simplified into a function that takes one argument.

```
let foo = (fun a b c -> a * c + b) 7 3
```

Here is an outline for the simplified function:

```
let foo c = _____________
```

Q3 Ruby Regular Expressions

14 Points

Q3.1 Fix RegEx 1

2 Points

Insert or remove **just one character** from the following regular expression so that it matches the behavior explained below. If you change the regex by more than one character, your answer will not receive any credit.

```
^\w+\d+\w*
```

The regex should match these strings:

`CMSC389N`

`CMSC434`

`CHEM135`

But not these strings:

`AMST101`

`DENT305`

`BSCI207`

Q3.2 Fix RegEx 2

3 Points

I've been trying to write a regular expression that can accept nonempty strings containing only lowercase letters, but I'm having some issues. What's wrong with my regular expression below? Explain the problem, and give a correct regular expression.

My regular expression is `^[a-z]$`.

What is wrong with the current regular expression?

What would the correct regular expression be?

Q3.3 Fix RegEx 3

4 Points

I'm trying to write another regular expression that matches a comma-delimited list of at least 3 nonnegative integers, something like the following:

```
1,2,3
```

```
142,2,412,42,34
```

My regular expression is `^\d+(,?\d*)+$`.

Give an example of a test string that shows that my regular expression does not behave as expected. This can be either a string that should be accepted but is instead rejected, or it could be a string that should be rejected but is instead accepted.

Your test string:

- This string should be rejected, but it is instead accepted
- This string should be accepted, but it is instead rejected

Fix the regular expression so that it works as expected. You may change as many characters in the regular expression as you would like.

Q3.4 Write RegEx 1

5 Points

Write a regular expression that accepts strings of the form

```
id: XXX-XX-XXXX codename: <codename>
```

where each `X` represents a number (can be different numbers) and `<codename>` is a string beginning with an uppercase letter that may have additional uppercase and/or lowercase letters after it.

For example, the following strings should be accepted:

```
id: 669-98-3600 codename: Watch
```

```
id: 123-45-6789 codename: McGregor
```

And the following strings should be rejected:

```
id: 123456789 codename: Wrong
```

```
id: 987-65-4321 codename: nope
```

Write your regular expression below:

Q4 Ruby Fill-in-the-Blank

12 Points

Q4.1 Complete the Function 1

4 Points

Complete the code snippet to produce the desired output.

```
def my_function(input)
    input =~ /(________) (________)/
    puts ($1 + $2)
end
```

Example usage:

```
my_function("Jin Ramen 123 xyz")       # Should output "Ramen123"
my_function("Pokemon 15")              # Should output "Pokemon15"
my_function("Pavan 4321 hello world")  # Should output "Pavan4321"
```

First blank:

Second blank:

Q4.2 Complete the Function 2

2 Points

Complete the implementation of `for_each`, which takes a list and a code block and passes each element into the code block, one at a time.

```
def for_each(lst)
    for x in lst do
        _________
    end
end
```

You can assume a code block is given (no need to check). Your answer should only include the part that goes on the blank.

Q4.3 Code Blocks

6 Points

You are given the following `Fruit` class:

```
class Fruit
    attr_accessor :name, :price
    def initialize(name, price)
        @name = name
        @price = price
    end
end
```

as well as an array of `Fruit` objects, stored in a global variable called `fruit`:

```
fruit = [
    Fruit.new("Apple", 100),
    Fruit.new("Grape", 50),
    Fruit.new("Apple", 300),
    Fruit.new("Banana", 40),
    Fruit.new("Grape", 200),
    Fruit.new("Apple", 50)
]
```

Fill in the blank in the `filter` method so that it uses the passed-in code block to determine whether or not to print an element.

```
def filter(a)
    a.each { |x|
        if ______A_____ then
            puts "#{x.name},#{x.price}"
        end
    }
end
```

Blank A

Now fill in blank B so that it produces the desired output.

```
filter(fruit) { __________B__________ }
```

Desired output:

```
Apple,100
Apple,300
Grape,200
```

Blank B

Now fill in blank C so that it produces the desired output.

```
filter(fruit) { __________C__________ }
```

Desired output:

```
Apple,100
Apple,300
Apple,50
```

Blank C

Q5 Ruby Programming

19 Points

There is an election and you have been given the very important task of calculating who will be the future president. DVH of the Racket party is up against Anwar Mamat of the C++ party. You will receive a file with votes on each line and you must tally them up. Each line in the file will correspond to a voting district. Lines will have the following form:

```
Dist[<id>]: <votes>
```

The `<id>` is a 6-digit number, and `<votes>` is a comma-separated list of votes, each of which is either AM or DVH. For example, a line might look like the following:

```
Dist[012345]: DVH,AM,DVH,DVH,AM,AM
```

This means in district 012345, DVH got 3 votes and AM got 3 votes.

There will be a colon and a single space after the district id, and no other whitespace anywhere on the line. You will need to complete the implementation of the class `Election`, which will process this voting data. The three functions you will write are described in the sections below.

The class outline is given to you. Please copy-paste each function into the corresponding answer section and fill in the indicated areas (you should not modify anything outside the indicated areas).

```ruby
class Election
    def initialize(filename)
        @votes = {}
        File.readlines(filename).each do |line|
            # TODO: implement the body of this loop
        end
    end

    def district_tally(district_id)
        # TODO: implement this function
    end

    def find_winner()
        # TODO: implement this function
    end
end
```

Q5.1 initialize(filename)

5 Points

This is the class constructor, it takes the name of the file containing the voting information (as described above). You should read it in and store it in the `@votes` Hash. If any line of the file is invalid, simply ignore it and keep going. The exact method you use to store the data in the Hash is up to you, but you must store all of the necessary data in this Hash. You may assume that each district ID only appears once in the file: we will not be checking your code for duplicate IDs. (HINT: We recommend creating a key for each district ID and storing the number of votes for each candidate as the value.)

Q5.2 district_tally(district_id)

7 Points

This method takes a district ID (a 6-digit number) as an integer argument and returns the winner in that district, which should be a string containing either "DVH" or "AM". Assume there is never a tie, and assume a district with the given district_id exists.

Q5.3 find_winner()

7 Points

The winner of the election is the candidate who won in the most districts. Return a string containing the overall winner, either "AM" or "DVH". Again, assume there is no tie.

Q6 OCaml Typing

18 Points

Q6.1 Type of a Function

4 Points

What is the type of the function `f` based on the following function call and its indicated output?

The call `f 3 9.4` outputs `(6, true)`.

What is the type of `f`?

Implement the function `f`.

Q6.2 Write Expression with Type 1

3 Points

Write an expression of the following type without using type annotations, and without any non-exhaustive pattern matching: `int list -> int -> int`.

Q6.3 Write Expression with Type 2

3 Points

Write an expression of the following type without using type annotations, and without any non-exhaustive pattern matching: `'a list -> 'b -> 'b list`.

Q6.4 Find the Type Error 1

4 Points

Identify what specific portion is causing the type error in the snippet below and what you can change to have it output the correct value.

```
let f a b =
    match b with
    | [] -> failwith "error"
    | x::y -> if x > 3 then y::a else b;;
```

Expected usage:

```
f [9;8;7] [4;1]           (* Returns: [1;9;8;7] *)
f [1;2;3] [4]             (* Returns: [1;2;3] *)
f [5;6;7] [10;11;12;13]   (* Returns: [11; 12; 13; 5; 6; 7] *)
```

Q6.5 Find the Type Error 2

4 Points

Briefly explain the error in the following code. You can plug it into utop, but you must explain what the error means and why it is occurring. The function should return the number of occurrences of target in `lst`.

```
let rec count_occ lst target =
    match lst with
    | [] -> []
    | h :: t -> (if h = target then 1 else 0) + count_occ t target
```

Q7 OCaml Fill-in-the-Blanks

10 Points

Q7.1 sec_to_last

5 Points

Complete the function `sec_to_last` below which returns the second to last element in a list.

```
let rec sec_to_last lst =
    match lst with
    | ____A____ -> ____B____
    | h::t -> ____C_____
    | _ -> failwith "Not enough elements!"
```

Your answer should only include the parts that go on the blanks.
Blank A

Blank B

Blank C

Q7.2 count

5 Points

Complete the function `count` which takes an element and a list, returns the number of times the element appears in the list.

```
let count elem lst =
    fold (fun a x -> if x = elem then ___A____ else ____B___) ___C____ lst
```

Your answer should only include the parts that go on the blanks.
Blank A

Blank B

Blank C

Q8 OCaml Programming

16 Points

Q8.1 is_mod_fibonacci

10 Points

Define a function `is_mod_fibonacci` of type `int list -> boolean` which given a list of integers, determines if that list fits the requirements of a modified Fibonacci sequence. This modified Fibonacci sequence refers to any list such that every element of the list is the sum of the previous two elements. If the list has fewer than 3 elements, simply return `true`.

Some examples:

```
is_mod_fibonacci [0; 1; 1; 2; 3; 5; 8] = true
is_mod_fibonacci [15; 16; 31; 47] = true
is_mod_fibonacci [10; 20; 30] = true
is_mod_fibonacci [7;8;13;45] = false
```

Implement the function `is_mod_fibonacci` below:

Q8.2 count_threes

6 Points

Fill in the blanks to complete the function `count_threes` of type `int list list -> int` that counts the number of 3's in an `int list list`.

Since `map` and `fold_left` are used, their definitions are given below for your reference:

```
let rec map f l =
    match l with
    | [] -> []
    | h :: t -> (f h) :: (map f t)

let rec fold_left f acc l =
    match l with
    | [] -> acc
    | h :: t -> fold_left f (f acc h) t
```

Some examples:

```
count_threes [[1;2;3] ; [3;3]] = 3
count_threes [[2;2;23;4;3]] = 1
count_threes [] = 0
count_threes [[] ; []] = 0
```

```
let count_threes lst =
    fold_left (fun ____A____-> ____B____ ) 0
        (____C____ (fun ____D____->____E____) [] lst)
```

Blank A

Blank B

Blank C

Blank D

Blank E