OCaml Intro + Language Learning Skills

For this next activity, we're going to work on a strategy that can be helpful both when we're debugging our own compilers and when we're trying to learn a new language. Here's the process we'll use.

1. Read the program carefully.
2. Predict what will happen.
3. **Write down what you expect will happen before running it**. I know this seems silly, but actually grab a piece of paper and jot it down! It's important to commit to your prediction and think about the mental model of the language that has led you to make that prediction. It's ok to be wrong! We're just looking to get some information that will either confirm our mental model or help us adjust the mental model to something more accurate.
4. Run the program. If your prediction wasn't right, what part of the mental model might have failed you? What's an alternative mental model that would produce the right answer? Do you need other resources to help you find out? Or is there another program you could run that would help you test a new hypothesis?

What will be the printed output for each of the following OCaml programs? For lines where there are multiple alternatives in boxes, please predict the output for each possible program variation in turn. (First predict the output if the line is variant (a), then if it's variant (b), etc.)

Remember, make your predictions *before* you run the programs! But feel free to run the programs once you've jotted down your predictions.

If you don't yet have OCaml set up on your machine, feel free to use an online resource like https://www.tutorialspoint.com/compile_ocaml_online.php to run your programs.

```
let f x =                                                    1
        if x = "61C"
        then "A prereq!"
        else if x = "164"
        then "I'm taking this now!"
        else if x = "264"
        then "I may take this later!"
        else "Default: " ^ x
;;
```

(a)	(b)	(c)
`print_endline (f "test");;`	`print_endline (f "164");;`	`print_endline (f 164);;`

(a) _____

(b) _____

(c) _____

```
let g x =                                                    2
        match x with
          "61C" -> "A prereq!"
        | "164" -> "I'm taking this now!"
        | "264" -> "I may take this later!"
        | _     -> "Default: " ^ x
;;
```

(a)	(b)	(c)
`print_endline (g "test");;`	`print_endline (g "164");;`	`print_endline (g 164);;`

```
let ready prior_course current_course =
       match prior_course with
         "61C" -> match current_course with
                    "164" -> "ready!"
                    | _ -> "hm, not sure"
       | _ -> "hm, not sure about that either"
;;
```

3

(a)	(b)	(c)
`print_endline (ready "61C" "164");;`	`print_endline (ready "61C" "160");;`	`print_endline (ready "160" "164");;`

8

(a) _____

(b) _____

(c) _____

```
let ready_2 prior_course current_course =
       match prior_course with
         "61C" -> (match current_course with
                    "164" -> "ready!"
                    | _ -> "hm, not sure")
       | _ -> "hm, not sure about that either"
;;
```

4

(a)	(b)	(c)
`print_endline (ready_2 "61C" "164");;`	`print_endline (ready_2 "61C" "160");;`	`print_endline (ready_2 "160" "164");;`

8

(a) _____

(b) _____

(c) _____

```
let ready_3 prior_course current_course =
       match prior_course, current_course with
         "61C", "164" -> "ready!"
       | _ -> "not sure!"
;;
```

5

(a)	(b)	(c)
`print_endline (ready_3 "61C" "164");;`	`print_endline (ready_3 "61C" "160");;`	`print_endline (ready_3 "160" "164");;`

6

(a) _____

(b) _____

(c) _____