



Section-Wide Items

Video Lectures

Homework 2



编程作业: Homework 2
3h



阅读材料: Homework 2
Detailed Guidelines for Peer
Assessment



互评作业: Homework 2
正在评分



审阅同学的作业: Homework
2

Community-Contributed Resources



Problem 1a

```

1 fun all_except_option (s,xs) =
2   case xs of
3     [] => NONE
4   | x::xs' => if same_string(s,x)
5                 then SOME xs'
6                 else case all_except_option(s,xs')
7                       NONE => NONE
8                       | SOME y => SOME(x::y)
9

```

- It is fine for a solution to be more complicated if the reas to be in **xs** multiple times although the problem does no this situation.
- It is fine to use the = operator directly instead of **same_s** the specification asked for **same_string**.
- A helper function is really not needed here, but many per use one. Generally consider a good such solution to be w okay if it is really nice.

Remember that you are grading on general style, not how c solution a student solution is. It is perfectly fine for a solutio different from the sample, as long as it has good style.

Problem 1b

Here is a sample solution:

```

1 fun get_substitutions1 (substitutions,str) =
2   case substitutions of
3     [] => []
4   | x::xs => case all_except_option(str,x) of
5                 NONE => get_substitutions1(xs,s
6                 | SOME y => y @ get_substitutions
7

```

- Give at most a 3 if they use helper functions other than **all_except_option** and ML's append operator **@**. This defined in `get_substitutions1` as they are not helpful here
- If a solution uses a local `val` binding for no useful purpose everything else is great, but this really is inferior style. Fo

```

1 fun get_substitutions1 (substitutions,str) =
2   case substitutions of
3     [] => []
4   | x::xs => let val foo = all_except_option(str

```