

H2: Medium	20 pt	
H3: Hard	55 pt	
Types		
O I1: Easy	20 pt	
I2: Medium	50 pt	
I3: Hard	80 pt	
Wanikani		
J1: Easy	10 pt	
J2: Medium	35 pt	
J3: Hard	45 pt	
Re-enact		
K1: Easy	15 pt	
K2: Medium	40 pt	
K3: Hard	50 pt	
Promotions		
L1: Easy	5 pt	
L2: Medium	20 pt	
L3: Hard	60 pt	

(i) FAQ

My Clarifications

Facebook © 2020 Contact Us

## **Problem I1: Types - Easy**

20 points

**Problem** My Submissions

#### Summary

You are developing a type system for a simple programming language, called *Lang*. Here is an example program:

```
(type even (int -> bool))
(type odd (int -> bool))

(func (even n) (if (= n 0) true (odd
(- n 1))))
(func (odd n) (if (= n 0) false (even
(- n 1))))

(type fib (int -> int))
(type helper (int int int -> int))

(func (fib n) (helper 0 1 (+ n 1)))
(func (helper n a b) (if (= n 0) a
(helper (- n 1) a (+ a b))))
```

Lang has the following types:

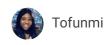
- int, representing the integers (..., -1, 0, 1, 2, etc),
- bool for booleans (true and false).
- (P1 P2 ... PN -> R) for any types P1, P2, ..., PN and R. Representing a function taking N arguments, and returning a value of type R.

A *Lang* program consists of a sequence of statements, each on their own line (Empty lines are ignored). A statement can be one of the following:

- A type declaration, of the form (type NAME TYPE) which introduces a new variable, with the given name and type.
- A function definition, of the form (func (NAME V1 V2 ... VN) BODY) which supplies the implementation of a function

### FACEBOOK Coding Competitions

# FB Hack > 2021 > EMEA Coding Challenge 2021



neen ueciaieu wiin n parameters, whose
names are given by V1, V2,, VN and the type
of its body will match the return type in its
declaration too.

Expressions in Lang are one of the following:

- An integer literal, like -1, 0, 1, 2 etc.
- A variable name, which can refer to a parameter of the function the expression is defined in, or any function declared before the expression in the program.
- A conditional of the form (if c a b) which evaluates to a if c evaluates to true and evaluates to b if c evaluates to false.
- A function call of the form (f e1 e2 ... en) where f is a variable name and ei is an expression for 1 <= i <= n.</li>

#### **Problem**

Initially, we want to know **how many types** are mentioned in the program. When counting mentions of compound types, we count the compound type but also its constituents subtypes, so the type (int bool -> (-> int)) mentions **5 types**: (int bool -> (-> int)), int, bool, (-> int), int. Note that we **count duplicates** as int is mentioned twice and counted both times.

Your input will be a *Lang* program with N lines. Count the total number of types mentioned according to this definition.

Note that *Lang* has the following built-in identifiers:

(type	true	bool)	)		
(type	false	bool)	)		
(type	+	(int	int	->	int))
(type	-	(int	int	->	int))
(type	=	(int	int	->	bool))
(type	<	(int	int	->	bool))

#### **Constraints**

1 <= N <= 100

#### **Example 1**

	H2: Medium	20 pt				
	H3: Hard	55 pt				
Тур	Types					
0	I1: Easy	20 pt				
	I2: Medium	50 pt				
	I3: Hard	80 pt				
War	nikani					
	J1: Easy	10 pt				
	J2: Medium	35 pt				
	J3: Hard	45 pt				
Re-e	enact					
	K1: Easy	15 pt				
	K2: Medium	40 pt				
	K3: Hard	50 pt				
Pror	notions					
	L1: Easy	5 pt				
	L2: Medium	20 pt				
	L3: Hard	60 pt				

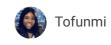
(i) FAQ

My Clarifications

Facebook © 2020 Contact Us

## FACEBOOK Coding Competitions

## FB Hack > 2021 > EMEA Coding Challenge 2021



	H2: Medium	20 pt			
	H3: Hard	55 pt			
Тур	Types				
0	I1: Easy	20 pt			
	I2: Medium	50 pt			
	I3: Hard	80 pt			
Wanikani					
	J1: Easy	10 pt			
	J2: Medium	35 pt			
	J3: Hard	45 pt			
Re-e	enact				
	K1: Easy	15 pt			
	K2: Medium	40 pt			
	K3: Hard	50 pt			
Promotions					
	L1: Easy	5 pt			
	L2: Medium	20 pt			
	L3: Hard	60 pt			

(i) FAQ

My Clarifications

Facebook © 2020 Contact Us

and (int int int -> int) **once each**. So given the example program as input, the expected output would be:

#### Output

14

### **Sample Input**

#### **Sample Output**

(type even (int -> bool)
(type odd (int -> bool)

(func (even n) (if (= n)
(func (odd n) (if (= n)

(type fib (int -> int))
(type helper (int int in)
(func (fib n) (helper 0)
(func (helper n a b) (if)

14