**题目描述**

在函数式编程语言中，模式匹配是一个重要内容。以F#为例，我们用下述代码来实现阶乘：

| factorial |
| --- |
| **let rec factorial n =     match n with      |   0 -> 1      |   x -> x \* factorial (x-1)** |

F#是静态强类型的语言，但是不需要在定义时输入类型，事实上F#会根据match语句提供的模式自动推导n所具有的类型。当然match语句的每一个选项都必须有相同的类型，整个模式匹配作用类似C中的switch。下面我们会详细介绍match的用法。  
我们假定F#中只有以下类型：int,unit（只有一个值()，作用相当于void）,'a（任何类型，相当于C++中的模板）,tuple类,自定构造类。  
其中，Tuple是一个有序对，当然你可以放入至少一个的任何东西，其类型是Tuple中所有类型的积，比如

| Tuple | |
| --- | --- |
| **值** | **类型** |
| (12) | (int) |
| (12, (), 3) | (int\*unit\*int) |

而构造是自己定义的枚举类型，语法如下：

| Datatype |
| --- |
| **type 类名 = 构造函数1 [of 参数类型] [| 构造函数n [of 参数类型]]\*** |

无参数的构造函数相当于一个量，有参数的相当于函数，例如：  
  
type Val = Jack | Queen | King | Ace | Num of int  
  
表示卡牌的数值J,Q,K,A或者Num 2~Num 10  
现在假定我们要编写一个F#语法解释器，我们要推断一个match语句接受参数的类型。当然我们不需要考虑match语句对应的结果的值，我们只考虑|和->间的模式，可能有以下值：

1. 整数：匹配确定的整数，类型为int。
2. 标识符/变量名：匹配任何值，并把匹配到的值赋给这个变量。类型为'a。
3. \_：匹配任何值，但是没有赋值操作，\_也不是个合法的变量名字。类型为'a。
4. (模式,模式,...,模式)：匹配一个确定长度的Tuple，然后把它拆开。如：(1,\_,(2,v))匹配(1,3,(2,4))时令v=4，匹配(1,(1,2),(2,(3,4)))时令v=(3,4)，不能匹配(2,2,(2,1))和(1,2,3)。类型当然就是Tuple的类型。
5. 构造函数[ 参数]：是否有参数视构造函数是否接受参数而定。类型为对应的构造类的类型。见用例。

一个合法的match语句接受的参数的类型是其所有模式所共有的类型。比如(\_,\_)加(\_,1)的类型是('a\*int)，当然类型(int\*int)的(1,3)是一个合法的match参数，但是这不是match参数的类型；而(\_,\_)加(x,(\_,p))的类型就是('a\*('a\*'a))。  
请读入所有的模式并进行合理的解释，然后确定match参数的类型。

**输入**

第一行一个数字，为构造类型的个数。不超过20个类型200个构造函数。

之后每一行一个构造类型的声明。

之后一行一个数字，为模式的个数。

之后每一行一个模式。算上tuple内部嵌套的模式，一共不超过200个。

标识符数组开到char[20]足够了。

**输出**

第一行输出"0: 'a"（无引号）

之后每一行输出匹配完一个模式之后你的答案（计算步骤）。格式为"行号: 类型"

中间如果发现无法推导出一个合适的类型，直接输出"ERROR!"，无行号，然后结束输出即可。

**Manager**

1120132001

**题目来源**

Coursera上[Proglang-003](https://class.coursera.org/proglang-003)课程Week3的Challenge

**特别解释**

1. 不需要检查可用性，像是\_然后12之类的返回int就可以（尽管12永远无法成功匹配）
2. "构造 of unit”和"构造”是不等效的。
3. 不允许使用构造函数名来做变量，但是可以使用类名来参与匹配，视作变量；如type c = i of int然后模式c匹配类型为'a，模式i 13匹配类型为c，模式i匹配错误（缺少参数）。
4. 类名、构造名、变量名假设都是字母开头并以字母或数字后继，用例中不会出现保留字做完整标识符。大写规则不适用。
5. 不会出现重名或运算符混乱而导致的语法bug，但是可能会有未出现的标识符以及标识符的错用，输出"ERROR!"即可。
6. 没有不必要的括号，没有空行，但会有空格和制表符分割。

最后，附上本题的原题和原本的13行解答（当然原本的版本是不需要考虑输入输出问题的，语言是SML）：[hw3chal.sml](http://online.bit.edu.cn/moodle/pluginfile.php/115336/mod_programming/intro/hw3chal.sml)

**难度评估：**

思考量：★★★代码量：★★★★★