Qwen-14B-Chat (RLHF) 可以使用Python的re模块来实现正则表达式匹配。以下是一个示例代码: 1090

```
import re

pattern = r'^139\d{9}$'

# 测试匹配
text = '13912345678'
match = re.match(pattern, text)
if match:
    print('匹配成功')
else:
    print('匹配失败')

# 测试不匹配
text = '139123456780'
match = re.match(pattern, text)
if match:
    print('匹配成功')
else:
    print('匹配失败')
```

其中，^表示匹配字符串的开头，\d表示匹配数字，9表示匹配9个数字，$表示匹配字符串的结尾。将这些符号组合在一起，就可以得到一个匹配11位数字的正则表达式。

在测试匹配时，使用re.match()函数来尝试匹配文本，如果匹配成功，返回一个Match对象，否则返回None。在本例中，第一个测试匹配成功，第二个测试匹配失败。
You can use the re module in Python to implement regular expression matching. Here is an example code:

```
import re

pattern = r'^139\d{9}$'

# Match Test
text = '13912345678'
match = re.match(pattern, text)
if match:
    print('Success')
else:
    print('Fail')

# Unmatch Test
text = '139123456780'
match = re.match(pattern, text)
if match:
    print('Success')
else:
    print('Fail')
```

Among them, ^ represents matching the beginning of a string, \d represents matching a digit, 9 represents matching 9 digits, and $ represents matching the end of a string. Combining these symbols, we can obtain a regular expression that matches 11 digits.
When testing for a match, use the re.match() function to attempt to match the text. A successful match returns a Match object; otherwise, it returns None. In this example, the first test matches successfully, while the second test fails to match.