班级：------　　姓名：------　　学号：------　　实验分数：------

实验一　谓词逻辑表示法

```
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"
```

一、实验目的

1、熟悉一阶谓词逻辑表示法；

2、用程序设计语言将习题2.1中的语句表示出来，并能实现简单的查询（查询内容可以自己设计）；

3、用程序设计语言将习题2.4中的语句表示出来，并能实现推理。

注：公式的latex转换链接https://www.latexlive.com/

样例

```
def likes(people, hobby, mode = 1):
    DATA = [['John', 'Sports'], ['John', 'Games'], ['Julia', 'Music'], ['Sam',
'Games']]
    if (people[0] != '$') and (hobby[0] != '$'):
        return [people, hobby] in DATA
    else:
        T = DATA[:]
        if (people[0] != '$'):
            T = [x for x in DATA if x[0] == people]

        if (hobby[0] != '$'):
            T = [x for x in T if x[1] == hobby]

        return T
```

问题一：John喜欢运动吗？

```
likes('John','Sports')
```

```
True
```

问题二：John喜欢什么？

```
X = likes('John','$x')
X = [x[1] for x in X]
X
```

```
['Sports', 'Games']
```

问题三：John喜欢Sports和Games？

```
likes('John','Sports') and likes('John','Games')
```

```
True
```

```
问题五：谁喜欢Sports和Games？
```

```
X1 = likes('$name','Sports')
X1 = [x[0] for x in X1]
X1
X2 = likes('$name','Games')
X2 = [x[0] for x in X2]
X2
```

```
['John']
```

```
['John', 'Sam']
```

```
print([x for x  in X1 if x in X2])
```

```
['John']
```

```

```

二、实验内容

（一）用程序设计语言实现习题2.1：

1、有的人喜欢梅花，有的人喜欢菊花，有的人既喜欢梅花又喜欢菊花。

1）定义谓词：

```

```

2）谓词的应用：

问题1：钱一喜欢梅花：

```

```

问题2：有的人喜欢梅花，有的人喜欢菊花，有的人既喜欢梅花又喜欢菊花。

```

```

问题三：谁既喜欢梅花又喜欢菊花？

```

```

2、他每天下午都去玩足球。

3、所有人都有饭吃。

4、喜欢玩篮球的人必喜欢玩排球。

5、要想出国留学，必须通过外语考试。

（二）用程序设计语言实现习题**2.4**：

1、如果张三比李四大，那么李四比张三小。

（1）用谓词公式表示：

old(x,y)：x比y大。则语句*如果张三比李四大，那么李四比张三小*可以表示为：

$$old(Zhang, Li) \rightarrow \neg old(Li, Zhang)$$

（2）用程序实现：

```
def old(x,y):
    if x>y:
        return True
    else:
        return False
```

```
Zhang = 21
Li = 20
print(old(Zhang,Li),not(old(Li,Zhang)))
print("old(Zhang,Li)=>not(old(Li,Zhang))")
```

```
True True
old(Zhang,Li)=>not(old(Li,Zhang))
```

2、甲和乙结婚了，则或者甲为男，乙为女；或者甲为女，乙为男。

（1）用谓词公式表示：

（2）用程序实现：

```
def Man(x):
    if x == '男':
        return True
    else:
        return False
```

```
def Marry(x,y):
    if Man(x) and not(Man(y)):
        return True
    elif Man(y) and not(Man(x)):
        return True
    else:
        return False
```

```
甲 = '男'
乙 = '女'
print(Marry(甲,乙))
print((Man(甲) and not(Man(乙))) or (Man(乙) and not(Man(甲))))
print('Marry(甲,乙)=>(Man(甲) and not(Man(乙))) or (Man(乙) and not(Man(甲)))')

甲 = '女'
乙 = '男'
print(Marry(甲,乙))
print((Man(甲) and not(Man(乙))) or (Man(乙) and not(Man(甲))))
print('Marry(甲,乙)=>(Man(甲) and not(Man(乙))) or (Man(乙) and not(Man(甲)))')
```

```
True
True
Marry(甲,乙)=>(Man(甲) and not(Man(乙))) or (Man(乙) and not(Man(甲)))
True
True
Marry(甲,乙)=>(Man(甲) and not(Man(乙))) or (Man(乙) and not(Man(甲)))
```

3、如果一个人是老实人，他就不会说谎。张三说谎了，所以张三不是一个老实人。

三、结论

1、

2、

3、