# 解题思路

## 1.1、需求分析：

从键盘输入两个命题变元P和Q的真值，求它们的合取、析取、条件和双向条件的真值。

该需求比较简单，只需要计算四个真值即可。

## 1.2、实验原理：

（1）合取：二元命题联结词。将两个命题P、Q联结起来，构成一个新的命题P∧Q, 读作P、Q的合取, 也可读作P与Q。这个新命题的真值与构成它的命题P、Q的真值间的关系为只有当两个命题变项P = T, Q = T时方可P∧Q =T, 而P、Q只要有一方为F则P∧Q = F。P∧Q可用来表示日常用语P与Q, 或P并且Q。

（2）析取：二元命题联结词。将两个命题P、Q联结起来，构成一个新的命题P∨Q, 读作P、Q的析取, 也可读作P或Q。这个新命题的真值与构成它的命题P、Q的真值间的关系为只有当两个命题变项P = F, Q = F时方可P∨Q =F, 而P、Q只要有一为T则P∨Q = T。P∨Q可用来表示日常用语P或者Q。

（3）条件：二元命题联结词。将两个命题P、Q联结起来，构成一个新的命题P→Q, 读作P条件Q, 也可读作如果P，那么Q。这个新命题的真值与构成它的命题P、Q的真值间的关系为只有当两个命题变项P = T, Q = F时方可P→Q =F, 其余均为T。

（4）双向条件：二元命题联结词。将两个命题P、Q联结起来，构成一个新的命题P←→Q, 读作P双条件于Q。这个新命题的真值与构成它的命题P、Q的真值间的关系为当两个命题变项P = T, Q =T时方可P←→Q =T, 其余均为F。

## 1.3、程序框架

首先读入P和Q的数据，并且检查合法性，如果不合法，需要重新读入。之后根据实验原理进行计算，输出计算结果。最后根据用户的输入，决定继续进行计算还是退出程序。

# 2、数据结构

逻辑结构：线性结构

存储结构：顺序表，静态存储。

代码：int a[4];

# 3、核心算法

## 3.1、算法逻辑：

与运算：使用&&

或运算：使用||

蕴含运算：转换成

双向蕴含运算：转换成

## 3.2、代码：

void calcu(int\* a, int i, int j)

{

a[0] = i && j; //与运算

a[1] = i || j; //或运算

a[2] = (!i) || j; //蕴含运算，将其转化为与或非形式

a[3] = ((!i) || j) && ((!j) || i); //等值运算，将其转化为与或非形式

}//进行运算