

离散数学
项目说明文档

命题逻辑联结词

作者姓名:	<u>高逸轩</u>
学 号:	<u>2053385</u>
指导教师:	<u>唐剑锋</u>
学院专业:	<u>软件学院 软件工程</u>



同济大学
Tongji University

1 题目简介

1.1 题目要求

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。

1.2 项目需求分析

本项目在实现的过程中，考虑并且满足了以下的需求：

✓ 健壮性

当用户输入的数据不合理时，系统应当给予相应的提示而非直接报错。

1.3 项目要求

1.3.1 功能要求

输入两个命题变量 p 、 q 的值（0/1），计算并输出两个变量的合取、析取、蕴涵、等值运算结果。若用户输入错误，如输入数字不是 0/1、输入字母等，则提示用户重新输入 p/q 的值，直到用户输入正确为止。在本次逻辑运算结束之后，询问用户是否继续进行下一次运算（y/n），输入 y 则清屏，重复上述步骤；输入 n 则退出程序。若用户本次输入错误，则提示用户重新输入，直到输入正确。

1.3.2 输入格式

两个命题变量 p 、 q 的值（0/1）

1.3.3 输出格式

打印输入两个变量的合取、析取、蕴涵、等值运算结果，并询问用户是否继续进行下一次运算。

1.3.4 项目简单示例

```
*****
**                                     **
**           欢迎进入逻辑运算程序           **
**                                     **
*****

    请输入P的值（0或1），以回车结束:1

    请输入Q的值（0或1），以回车结束:0

    合取：
         $P \wedge Q = 0$ 
    析取：
         $P \vee Q = 1$ 
    条件：
         $P \rightarrow Q = 0$ 
    双条件：
         $P \leftrightarrow Q = 0$ 

    是否继续运算？(y/n) n
    欢迎下次再次使用！
```

2 项目实施

本项目核心共两部分：

- ✓ 健壮性处理
- ✓ 逻辑运算

下面将依次对本项目核心的这两个部分进行介绍。

2.1 健壮性处理

为实现输入 p/q 的值的健壮性，使用了自定义头文件 `input_tools.h` 和文件 `input_tools.cpp`，其详细介绍请见其介绍报告。

为实现输入是否进入下一次运算的提示 y/n 的健壮性，部分代码如下：

```
// 判断是否进入下一次运算
char c = '\0';
judge: printf("\n是否继续运算? (y/n) ");
cin >> c;
cin.clear();
cin.ignore(1024, '\n'); // 清除缓存区，避免缓存区中的剩余字符对后续运行产生影响
if (c == 'y') system("cls"); // 选择了y，则继续运行，清屏
else if (c == 'n')
{
    printf("欢迎下次再次使用!\n"); // 选择了n，则退出，跳出循环
    break;
}
else
{
    printf("输入错误. 请重新输入!\n"); // 选择了其他选项，则重新输入用户选择
    goto judge;
}
```

其思想是每次从输入缓存区中读入一个字符，同时清除缓存区，避免一次错误输入多个字符导致后续多次输入错误提示。若其为 y ，则清屏，重新进入下次循环，进入下次运算；若其为 n ，则给出提示“欢迎下次再次使用”，并跳出循环，结束本次程序；若其输入错误，则给出提示“输入错误，请重新输入！”，并重新输入 y/n 。

2.2 逻辑运算

2.2.1 运算原理介绍

(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 $P \wedge Q$ ，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 $P = 1$ ， $Q = 1$ 时方可 $P \wedge Q = 1$ ，而 P 、 Q 只要有一方为 0 则 $P \wedge Q = 0$ 。 $P \wedge Q$ 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 $P \vee Q$ ，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 $P = 0$ ， $Q = 0$ 时方可 $P \vee Q = 0$ ，而 P 、 Q 只要有一为 1 则 $P \vee Q = 1$ 。 $P \vee Q$ 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 $P \rightarrow Q$ ，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 $P = 1$ ， $Q = 0$ 时方可 $P \rightarrow Q = 0$ ，其余均为 1。

(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 $P \leftrightarrow Q$ ，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 $P = 1$ ， $Q = 1$ 时方可 $P \leftrightarrow Q = 1$ ，其余均为 0。

2.2.2 代码实现介绍

```
// 对p q进行二元运算，并输出结果
printf("\n\n");
printf(" 合取:\n      P/\Q = %d\n", p & q);           // 合取式p/\q, 在变量运算中为 p & q
printf(" 析取:\n      P\Q = %d\n", p | q);           // 析取式p\q, 在变量运算中为 p | q
printf(" 条件:\n      P->Q = %d\n", (!p) | q);         // 条件运算式p->q, 即(!p)\q, 在变量运算中为(!p) | q
printf(" 双条件:\n      P<->Q = %d\n", ((!p) | q) & ((!q) | p)); // 双向条件运算p<->q, 即(p->q) /\ (q->p), 在变量运算中表示为((!p) | q) & ((!q) | p)
```

在 C++ 程序中，与运算符为 $\&$ ，或运算符为 $|$ ，非运算符为 $!$ 。其运算优先顺序为： $!$ $\&$ $|$

条件运算 $p \rightarrow q$ 可转化为 $!p | q$ ；双向条件运算可转化为 $(p \rightarrow q) \& (q \rightarrow p)$ 进而转化为 $(!p | q) \& (!q | p)$

故运算并输出的部分如上所示。

3 项目测试

3.1 四种输入对应运算结果

3.1.1 $p=0, q=0$

```
*****
**                               **
**      欢迎进入逻辑运算程序      **
**                               **
*****

请输入P的值（0或1），以回车结束:0

请输入Q的值（0或1），以回车结束:0

合取:
     $P \wedge Q = 0$ 
析取:
     $P \vee Q = 0$ 
条件:
     $P \rightarrow Q = 1$ 
双条件:
     $P \leftrightarrow Q = 1$ 

是否继续运算? (y/n) n
欢迎下次再次使用!
```

3.1.2 $p=0, q=1$

```
*****
**                               **
**      欢迎进入逻辑运算程序      **
**                               **
*****

请输入P的值（0或1），以回车结束:0

请输入Q的值（0或1），以回车结束:1

合取:
     $P \wedge Q = 0$ 
析取:
     $P \vee Q = 1$ 
条件:
     $P \rightarrow Q = 1$ 
双条件:
     $P \leftrightarrow Q = 0$ 

是否继续运算? (y/n) n
欢迎下次再次使用!
```

3.1.3 p=1,q=0

```
*****
**                                     **
**      欢迎进入逻辑运算程序      **
**                                     **
*****

请输入P的值（0或1），以回车结束:1

请输入Q的值（0或1），以回车结束:0

合取:
     $P \wedge Q = 0$ 
析取:
     $P \vee Q = 1$ 
条件:
     $P \rightarrow Q = 0$ 
双条件:
     $P \leftrightarrow Q = 0$ 

是否继续运算? (y/n) n
欢迎下次再次使用!
```

3.1.4 p=1,q=1

```
*****
**                                     **
**      欢迎进入逻辑运算程序      **
**                                     **
*****

请输入P的值（0或1），以回车结束:1

请输入Q的值（0或1），以回车结束:1

合取:
     $P \wedge Q = 1$ 
析取:
     $P \vee Q = 1$ 
条件:
     $P \rightarrow Q = 1$ 
双条件:
     $P \leftrightarrow Q = 1$ 

是否继续运算? (y/n)
n
欢迎下次再次使用!
```

3.2 健壮性测试

3.2.1 p、q 值输入测试

```
*****
**                                     **
**      欢迎进入逻辑运算程序      **
**                                     **
*****

请输入P的值（0或1），以回车结束:qjf
P的值输入有误, 请重新输入!
请输入P的值（0或1），以回车结束:2
P的值输入有误, 请重新输入!
请输入P的值（0或1），以回车结束:0

请输入Q的值（0或1），以回车结束:oihugu8
Q的值输入有误, 请重新输入!
请输入Q的值（0或1），以回车结束:861
Q的值输入有误, 请重新输入!
请输入Q的值（0或1），以回车结束:1

合取:
     $P \wedge Q = 0$ 
析取:
     $P \vee Q = 1$ 
条件:
     $P \rightarrow Q = 1$ 
双条件:
     $P \leftrightarrow Q = 0$ 
```


3.2.3 是否继续下一轮运算测试

不进入下次运算

```
是否继续运算? (y/n) wqfewgrehtn
输入错误, 请重新输入!

是否继续运算? (y/n) 12343
输入错误, 请重新输入!

是否继续运算? (y/n) 2leqwrefws
输入错误, 请重新输入!

是否继续运算? (y/n) Y
输入错误, 请重新输入!

是否继续运算? (y/n) n
欢迎下次再次使用!
```

进入下次运算

```
是否继续运算? (y/n) efsgrdgf
输入错误, 请重新输入!

是否继续运算? (y/n) 21
输入错误, 请重新输入!

是否继续运算? (y/n) y

*****
**                                     **
**          欢迎进入逻辑运算程序          **
**                                     **
*****

请输入P的值 (0或1) , 以回车结束: _
```

4 心得与总结

本次项目是离散数学课程的第一个作业，在编程中体现了离散数学课堂的知识。首先为实现项目所要求的四种逻辑运算，需要掌握析取、合取、蕴涵、等价四种运算的定义，且掌握与运算、或运算、非运算等基本位运算符在 C++ 程序中的书写方式，最终将逻辑运算问题转换为基础的位运算。同时，为了实现程序的健壮性，并保证此部分代码的可复用性，我编写了 `getint()` 函数并储存在 `input_tools.h` 与 `input_tools.cpp` 中，以便于后续项目使用。并对上述文件写了详细报告，麻烦老师额外花费时间阅读。