

实践3小结

- **silly mistakes:**

- 提交有语法错误的程序；鉴于VS2010容错性太强，语法使用不熟练的同学，在机房实验时建议使用VC6.0。
- 从键盘输入信息前，无任何提示；不读题。

- **logic problems:**

- 键盘输入信息与程序逻辑之间的关系；
- if语句的逻辑；
- switch语句通常情况下default分支也要以break结束；

- **赋值表达式**

- 有方向性，左值有可写内存，右值有确定值。
- 运算符：=，复合赋值(+=等)，自增自减(++等)。

实践3小结

- **if**语句

- 作为条件的表达式外的()**不能少**，返回**bool**型。
- 语句块外的**{ }**，在单个语句时**可省略大括号**——**多个语句组成的块一定要用大括号括起来！**

Tips: 无论是不是一句话，都{}起来！是个很保险的做法！
但是，请注意！下列三组语句等价。

```
if (a>=0)
{
    if (b>=0) cout<<b;
    else cout<<-b;
}
```

```
if (a>=0)
{
    if (b>=0) cout<<b;
    else cout<<-b;
}
```

```
if (a>=0)
    cout<<(b>=0?b:-b);
```

实践3小结

//错误1

```
if(x<1)
```

```
else if(1<=x<=10)
```

```
else if(x>=10)
```

//错误2、3

```
if(1<=x & x<=10)
```

```
if(1<=x , x<=10)
```

//瑕疵

```
if(x<1)
```

```
else if(1<=x && x<=10)
```

```
else if(x>=10)
```

//正确设计

```
if(x<1) {...}
```

```
else if(x<=10) {...}
```

```
else{...}
```



实践3小结

- switch语句

- 作为条件的表达式外的()不能少

- 取值不仅是bool型，与if语句的差别
 - 取值只能是离散量(int, char, enum, bool)，不能是连续量

- 语句块无需{}

- 通常，每个分支(包括default)的语句序列均以break结束。此时，各分支(包括default)的前后顺序不影响程序运行结果。
 - 若出于特别的逻辑设计，省略了break，则分支顺序的设计将影响程序运行结果。

实践4 循环结构程序设计

- 实验内容：
 - 自定义数据类型：数组
 - 三种程序结构的熟练掌握
 - 算法初步

实践4：

- 掌握C++程序循环结构
 - while、do while、for语句，循环语句
 - break, continue语句，转向语句。
- 调试程序实践
 - 出现逻辑错误时，debug功能观察变量，分析原因。
- 实践内容：课本习题2.5~2.7, 乘法表，实验四、3, 实验四、5.【课堂上会另提供几个趣味小题目】
- 本次实践共给出 4 题，请根据自己的学习情况酌情完成，底限是完成其中 2 题，且 for, while 两种循环结构都要练习使用！

实践4：提交程序清单1

1、课本习题2.5~2.7，循环结构实现简单功能。

要求1：不是用字符串直接输出，要求使用循环语句，使用循环语句，使用循环语句输出，预习时，计算好每轮循环次数与输出信息的位置之间的关系。

要求2：在程序顶部，使用注释给出程序运行结果。

并为自己的程序输出设计具有你自己的个人风格的表头，程序首先输出表头，计算自己做的题号：

$(\text{学号后二位} - 1) \% 3 + 5$

表头简例：

```
C:\WINDOWS\system32\cmd.exe
*****
* 学号: D0000035 *
* 习题: 2.6 *
*****
```

实践4：提交程序清单2

2、利用for循环语句，输出九九乘法表。

提示：两层循环的嵌套，外层循环 $i=1..9$ ，内层循环 $j=1..i$ 。（预习时分析每式与 i,j 的关系）

要求：程序运行后输出如下，每个算式之间用“水平制表符(Tab键)”间隔，但要求每行第一个算式前、最后一个算式后，不输出制表符。

$$1 \times 1 = 1$$

$$1 \times 2 = 2 \quad 2 \times 2 = 4$$

.....

$$1 \times 9 = 9 \quad 2 \times 9 = 18 \quad \dots \quad 9 \times 9 = 81 \quad \text{【未完待续】}$$

九九乘法表(续)

- 传统的下三角九九乘法表很容易就实现了，现在挑战一下，设计你的循环参数，输出上三角的九九乘法表
- 要求：程序运行后输出如下，每个算式之间用“水平制表符(Tab键)”间隔，但要求每行第一个算式前、最后一个算式后，不输出制表符。

$1 \times 1 = 1$ $1 \times 2 = 2$ $1 \times 3 = 3$ $1 \times 9 = 9$

$2 \times 2 = 4$ $2 \times 3 = 6$ $2 \times 9 = 18$

$3 \times 3 = 9$ $3 \times 9 = 27$

.....

$9 \times 9 = 81$

实践4：提交程序清单3

3、实验四-3，**Josephus**问题的**数组**实现。

要求：输入**N K M**(合理简洁的提示)，输出**淘汰顺序**。其中：
N为总人数，初始从第K人开始由1到M循环报数，报到M的人被淘汰。（若实现有困难，可设定K=1，并参考流程图实现）

要求：在程序顶部用注释给出你的测试结果，下列几个输入输出范例，帮助你**由简到繁**检验程序：

输入 9 1 1 输出 1 2 3 4 5 6 7 8 9

输入 8 5 2 输出 6 8 2 4 7 3 1 5

输入 41 1 3 输出 3 6 9 12 15 18 21 24 27 30 33 36 39

1 5 10 14 19 23 28 32 37 41 7 13 20 26 34 40 8 17 29 38

11 25 2 22 4 35 16 31 【测试case 3即原始约瑟夫问题】

Tips，提示信息及流程图见下页。

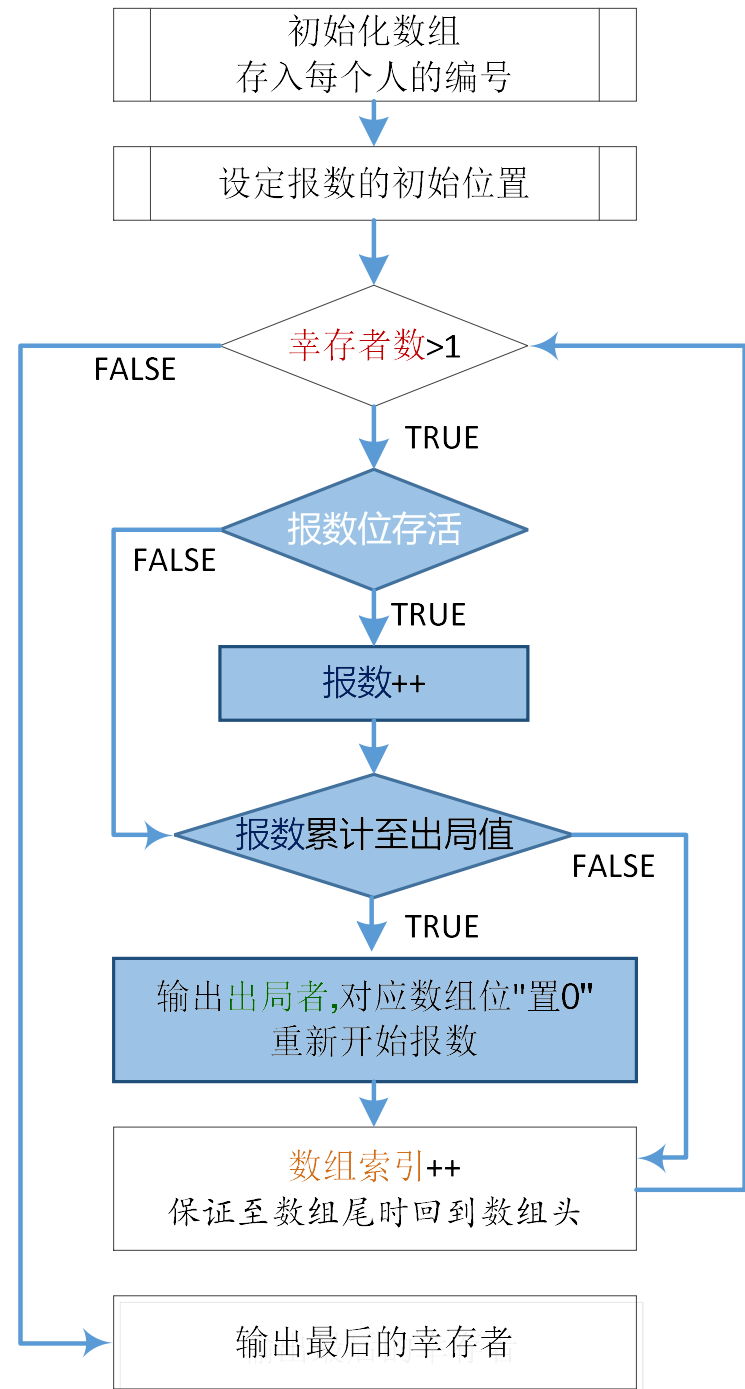
【未完待续】 

提示：数组大小可遵循“大开小用”原则，若定义size=100长度的数组，输入N<100时均可执行。

提示：数组初始化时可依次存入1,2...N，淘汰后相应位置的数组值设置为0。

注意：数组有效下标为：
 $0 \dots (\text{size}-1)$ ，当报数到最后一人时，回到第一人的位置，使用求余运算($\% N$)实现。

【未完待续】



约瑟夫问题（续）

- 作为学编程的经典题例，网络上Josephus问题解法大把大把的。请注意本题要求使用数组实现，其他任何“高级”方法都不允许使用。你可以选择不做，但若借鉴了，请理解后转换成本题要求的实现方式。
- 在初学阶段，数组采用“大开小用”方式，如定义size为100的数组，测试人数通常在50以内，此时存储空间足够，你可以选择从下标1开始使用数组，用以存储Josephus问题中每个人的编号。但是，既然学习成为C/C++程序员，请学习从下标0开始使用数组。
- 思考一下，若要求在程序最后一次性输出淘汰顺序，该如何实现。

实践4：提交程序清单4

4、实验四-5，**DayInYear**不限实现方式，**尝试多种方案实现**（每个人的脑洞不同，程序可有万千呈现）。

要求1. 键盘输入年月日，合理简洁的输入提示。

Tips: 本题与第3题一样，略有难度，若实现有困难的同学，也可实现本题部分功能，即**DateValidity**：**键盘输入一个日期**如：1900年2月29日，**判断该日期是否为合法日期并输出判断结果。**

要求：在程序顶部用注释给出你的测试结果，下列几个输入输出范例：

输入：1900 2 29 输出：日期非法

输入：2020 1 2 输出：2020年1月2日是2020年的第2天

【未完待续】 

实践4：提交程序清单4

提示：要计算某日期**年-月-日**是一年中的第几天，首先，前提是这个日期是一个**合法**的日期。然后，要根据**月**份，计算前面月份的**累计天数**。最后，加上**日**数。

难点：由于2月的存在，每月的天数不是单一规律的。

Tips:因闰年算法局限性，限定测试年份为1000~3000.

重要！！！！

提交的文件名格式：

Exp04_学号_实验名.cpp

- 注意:
- 1) 下划线
 - 2) 学号别漏掉
 - 3) 各种名字(变量名, 文件名)的可读性!
 - 4) 每个实验只要提交一个程序源文件, 即.cpp文件。