

【注意：】

- 1、本次作业**只能**定义简单变量并用顺序+分支+循环结构实现
- 2、**不允许**使用后续课程中的知识点，**包括但不限于**函数、数组、结构体、类等相关概念!!!
- 3、除明确要求外，已学过的知识中，不允许使用 goto
- 4、cstdio 及 cmath 中的系统函数**可以**直接使用，包括课上未介绍过的，具体可自行查阅相关资料
- 5、除明确要求外，所有 cpp 源程序不允许使用 scanf/printf 进行输入/输出
- 6、多编译器下均要做到“0 errors, 0 warnings”
- 7、部分题目要求 C 和 C++两种方式实现，具体见网页要求
- 8、输出为浮点数且未指定格式的，均要求为 double 型，C++为 cout 缺省输出，C 为%f 的缺省输出
- 9、认真阅读格式要求及扣分说明!!!

【输出格式要求：】

- 1、为方便机器自动判断正确性，作业有一定的输入输出格式要求（但不同于竞赛的无任何提示）
- 2、每个题目见具体说明，必须按要求输入和输出，不允许有偏差
- 3、没有特别说明的情况下，最后一行有效输出的最后有一个 endl
- 4、本次作业的**比对要求**为 txt_compare 在 --trim right 下与 demo 做到**完全一致**

【特别声明：】

本次作业的目标是如何从计算机的视角去考虑问题，附件所带的 5 个 exe **没有任何问题**，均遵守下述的“错误处理逻辑”和具体的题目要求，如果你的 exe 和附件的输出不同，**表示你没有理解**下述的“错误处理逻辑”和题目要求，**不希望**问出“我觉得附件的 exe 处理流程不合理，我认为 xx 的输出才合理的”等类似问题!!!

补充：

- 12、解决 scanf("%d", &int 型变量); / cin>>int 型变量时，因为输入错误（包括但不限于超过上下限、非法输入字母等情况）而导致的出错问题

【要求：】① 下述程序运行三次，分别要求：

- a) 输入一个[0-100]间的数字
 - b) 先输入一个<0 或>100 的数字，要求再次输入时输入 0-100 间的数字
 - c) 输入一个/多个字母或标点符号（观察此时的程序表现）
- ② 在给出的基准源程序 3-b12.cpp/3-b12.c 的基础上，自行查阅相关资料，使①-c 能正确处理输入错误并再次提示输入（**注：**只要输入与预期输出不符合，都算错误，例如，如果下例的 C++基准程序中输入字母，得到 x=0，也是需要处理的错误）
 - ③ 本题可以使用一些未学过的系统函数/cin.xx 的成员函数形式（即使某些语句不知道准确含义，只要有效即可，后续课程中会逐步学到，但**仍不允许**使用数组、string 等后续知识）
 - ④ 给出 3-b12-基准程序.cpp/3-b12-基准程序.c 供参考

- ⑤ c 方式输入错误时, vs 和 dev 还有细节差别, 因此提供两个 c 方式的 exe, 其中 Dev 下 C 方式的 exe **不要求** txt_compare --trim right 与 demo 完全一致
- ⑥ 后续题目中输入错误的处理, 均可以使用本题的方法

【错误处理逻辑:】(编译器视角的错误处理描述, 认真阅读并思考与人的视角的差异)

以 C++ 方式的基准源程序中 cin>>x 为例, 说明错误的不同类型及基本处理准则

- ① 假设键盘输入是 78, 则 x 读到 78, 此时 cin 状态正确, int 型变量读入正确, x 符合程序预期 [0..100], 整个程序正确结束
- ② 假设键盘输入是 123, 则 x 读到 123, 此时 cin 状态正确, int 型变量读入正确, 但 x 不符合用户程序的预期 [0..100], 会重新要求读入; 因为输入缓冲区已空, 因此光标闪动, 再次等待键盘输入
- ③ 假设键盘输入是 abcd: 则 cin 直接进入错误状态, 应该清除当前输入缓冲区并且恢复 cin 的正确状态, 重新读入
- ④ 假设键盘输入是 78abcd, 则 x 读到 78, 此时 cin 状态正确, int 型变量读入正确, x 符合程序预期 [0..100], 因此退出循环并打印 x, 随后整个程序正确结束 (此时 cin 状态正确, 不清除缓冲区, 因此缓冲区仍有 abcd 存在, 如果程序运行结束, 则缓冲区会随着程序运行结束而消失; 如果还有后续语句, 且后续语句中有输入语句, 则下次输入时会从缓冲区中读取, 如果下次是读取字符串/字符, 则可正确读入; 如果下次是十进制方式读取 int 型, 则会出错 (见③), **但无论如何, 不影响本次的正确性**)
- ⑤ 假设键盘输入是 123abcd, 则 x 读到 123, 此时 cin 状态正确, int 型变量读入正确, 但 x 不符合程序预期 [0..100], 会重新要求读入; 再次读入时碰到 abcd (详见④的描述, 同③处理即可)
- ⑥ 假设键盘输入是 9876543210abcd, 则 x 读到 9876543210, 超出了 int 范围, 因此 x 的值为 int 的上限/下限/0 等特殊值 (不同编译器表现不同), 且 cin 状态错误, 此时即使 x 的值为 0 (本课程所用的 VS+Dev 均为 int 上限), 符合程序预期的 [0..100], 但仍然应该同③处理
- ⑦ 如果是 scanf ("%d", &a) 方式入到 9876543210abcd, 则 x 有值 (非 9876543210, 具体自行研究) 且 scanf 返回 1, 无法判断错误
- ⑧ **任何情况下, 约定 cin 为错误状态才清空当前输入缓冲区中的所有内容, 否则不清除**

13、 从键盘输入年份、月份以及该月第 1 天的星期值, 打印某月份的月历

【要求:】① 年份限定在 2000-2030 之间, 月份 1-12, 星期值 0-6 (代表星期日-星期六)

- ② 有关闰年、大小月的判断必须正确
- ③ 不考虑输入星期值与实际值的正确性差别 (比如 2022.10.1, 星期输入 0-6 均可)
- ④ 给出示例程序 3-b13-demo.exe 供参考 (月历部分输出格式要一致)

注: demo 有 C/C++ 两种, 输入超过 int 上限时, 两者表现不同, 具体参考 12 题

⑤ **考虑各种输入错误且错误处理方法必须与 demo 一致**, demo 的输入处理逻辑如下:

- a) 首先输入年和月, 要求必须两者同时正确, 任何一个有错则需要重新输入年和月 (例: 输入 2020 abc 2020 2, 清空并重输, 输入 2020 13 2022 3, 则年/月为 2022/3)

【注意:】可以把⑤-a) **理解为**实际工作中甲方提出的特殊要求, 你现在要做的是使错误逻辑处理符合这个特殊要求, 而不是质疑甲方)

- b) 年月正确后, 进入星期的输入, 星期错误则重新输入星期, 不再输入年和月

【编译器要求:】

		编译器VS	编译器Dev
3-b12-1.cpp	解决cin输入int错误	Y	Y
3-b12-2.c	解决scanf输入int错误	Y	Y
3-b13-1.cpp	打印月历	Y	Y
3-b13-2.c	打印月历(C方式)	Y	Y

【作业要求:】

- 1、**4月2日前**网上提交本次作业
- 2、每题所占平时成绩的具体分值见网页
- 3、超过截止时间提交作业会自动扣除相应的分数，具体见网页上的说明

【特别说明:】

用输入重定向/管道方式处理含有错误输入的数据时，**最后一个数据必须是正确的**，和控制台输入的差别如下：

控制台方式下，前面若干次输入错误，系统都会清空缓冲区后要求重新输入，如果不输入，就会停在那里等待；

输入重定向/管道方式下，前面若干次输入错误，系统都会清空缓冲区后继续从输入文件/管道中读数据，如果读不到数据（文件结束或管道中无后续数据提供），系统会返回 EOF (End Of File)，此时 `cin.good()` 为 0，会进入**无限死循环**，从而导致输出重定向文件不停增长**（这就是上周很多同学碰到的问题）**。

注：用 `get_input_data.exe` 方式读 `test-data.txt` 形式的，要理解每个[]组是一个题目的测试数据，因此，每组的最后一个数据必须是正确的）