**输入域测试练习**

**背景：**

2020年12月31日，某品牌拖地机器人由于时间戳被错误地置为13月1日，导致当日无法正常使用（https://www.zhihu.com/question/437791481/answer/1659932774）。

**任务1:**

给定以下一个用C++语言编写的NextDate函数的骨架，其功能为：在输入一个合法的日期后，计算该日期下一天的年、月、日并直接通过参数返回。请补全该程序（不得改变函数接口）。

|  |
| --- |
| void NextDate(int &year, int &month, int &day)  {  if(!ValidDate(year, month, day)){  return false;  }  else{  if(month == 1 || month == 3 || month == 5 || month == 7 || month == 8 || month == 10){  if(day == 31){  ...  }  else{  ...  }  }  else if(month == 4 || month == 6 || month == 9 || month == 11){  ...  }  else if(month == 12){  ...  }  else{  if((LeapYear(year) && day == 29) || (!LeapYear(year) && day == 28)){  ...  }  else{  ...  }  }  }  } |

注意，用于判断输入合法性的ValidDate函数以及用于判断平年闰年的LeapYear函数请自行实现。

**任务2:**

使用GTest单元测试框架对补全后的NextDate函数进行测试。根据PIE模型的原理需要确保NextDate函数中所有的代码行均被执行过至少一次（整个if语句看作一行不考虑短路计算）。

注意，GTest单元测试框架的使用请通过中国大学MOOC平台线上自学。相关课程内容已发布。

**任务3:**

明确给出NextDate函数的输入域（input domain），描述形式不限。

**任务4:**

编写一个程序，随机生成NextDate函数的100条测试用例，测试用例存储至一个文本文件RT\_test\_data.csv，每行存放一条测试用例，测试用例内各个字段之间的分割方式按照csv规范。编写程序读取测试用例文件，使用GTest单元测试框架执行这100条测试用例对NextDate函数进行测试。

**任务5:**

使用PICT工具生成一组强度为2的组合测试用例，测试用例存储至一个文本文件CT\_test\_data.txt，每行存放一条测试用例，测试用例内各个字段之间的分割方式按照csv规范。编写程序读取测试用例文件，使用GTest单元测试框架执行这些测试用例对NextDate函数进行测试。

注意，组合测试的概念和PICT工具的使用请通过中国大学MOOC平台线上自学。相关课程内容已发布。

**提交要求：**

任务1、2以代码的形式提交；任务3描述形式不限；任务4提交生成随机测试数据的代码，以及读取测试数据文件并使用GTest进行测试用例执行的程序代码；任务5提交运行PICT的命令和PICT生成的数据，以及读取测试数据文件并使用GTest进行测试用例执行的程序代码。注意，任务4和任务5中读取测试数据文件并使用GTest进行测试用例执行的程序可以只实现一次。所有任务提交的内容拷贝至一个word文档中，文件以学号+姓名的方式命名。上机实验之后两周内以自然班为单位统一提交。

自动化批量生产测试用例时，如何自动获得测试用例的预期结果是一个问题。按照互联网公司常用的灰度发布AB测试的方式，可以使用一个不同版本NextDate函数（比如其他同学编写的）来产生预期结果。