

书: P. 339 4 (矩阵的定义及要求如下, 要求两种方法实现, 部分程序均已给出)

方法一: 矩阵的+重载用成员函数方式实现

方法二: 矩阵的+重载用友元函数方式实现

5 (要求同上, 矩阵的+重载用成员函数方式实现, 部分程序已给出)

补充:

1. 定义并实现一个 Date 类, 能实现以下要求:

① 有年、月、日三个成员

② 允许用以下方式进行对象初始化

Date d1; //2000. 1. 1

Date d2(2013, 5, 29); //2013. 5. 29

③ 定义 set 成员函数, 能设置对象的日期

同时设定年月日:

set(2013, 5, 29); //设置当前日期为 2013. 5. 29

set(2013, 5); //设置当前日期为 2013. 5. 1

set(2013); //设置当前日期为 2013. 1. 1

只设定年、月、日的部分(为 0 的参数不变)

(假设对象当前日期是 2013. 5. 29)

set(2007, 0, 0); //设日期为 2007. 5. 29(月日不变)

set(0, 3, 18); //设日期为 2013. 3. 18(年不变)

④ 定义 get 成员函数, 能取对象当前的日期

(假设对象当前日期是 2013. 5. 29)

get(y, m, d); //y, m, d 为 int 型变量, 调用后 y, m, d 的值分别是 2013, 5, 29

⑤ 定义 show 成员函数, 能显示对象当前的日期

(假设对象当前日期是 2013. 5. 29)

输出为: 2013 年 5 月 29 日

⑥ 定义转换构造函数, 将一个整数当做从 1900. 1. 1 开始的天数转换为 Date 类对象

假设 Date d1; d1=1; 则 d1 为 1900. 1. 1 (任何<=1 的数字都是 1900. 1. 1)

Date d2; d2=73049; 则 d2 为 2099. 12. 31 (任何>=73049 的数字都是 2099. 12. 31)

Date d3; d3=41422; 则 d3 为 2013. 5. 29

定义类型转换函数, 将一个 Date 类对象转换为从 1900. 1. 1 开始的天数

假设 Date d1(1900, 1, 1); 则 int(d1)的结果是 1

假设 Date d2(2099. 12. 31); 则 int(d2)的结果是 73049

假设 Date d3(2013, 5, 29); 则 int(d3)的结果是 41422

⑦ 重载+/-运算符, 实现以下功能

(设 d1 的日期 2013. 5. 29, d2 是 2013. 5. 25)

d1+17 / 17+d1 : 2013. 6. 15

d1-10: 2013. 5. 19

d1-d2: 4

⑧ 重载++/--运算符(前后缀都要), 实现以下功能

若 d1 当前为 2013. 5. 29, 则 d1++: 2013. 5. 30

若 d1 当前为 2013. 5. 31, 则 d1++: 2013. 6. 1

若 d1 当前为 2013. 1. 1, 则 d1--: 2012. 12. 31

⑨ 重载 cout 运算符, 与 show 函数一样的方式输出

重载 cin 运算符, 键盘输入三个整数依次为年月日

⑩ 重载 6 个比较运算符, 能比较两个日期型, 返回值为 bool 型

【要求:】1、部分程序已给出, 按要求实现即可

2、可参考例 10.5, 10.6 的 Time 类 (P.323-325)

3、以上各成员函数均允许重载

4、运算符重载成员函数、友元函数方式不限

5、限定日期的范围为 1900.1.1-2099.12.31, 闰年、大小月的合法性要进行考虑

6、在构造函数及 set 函数中若出现非法年份, 均用缺省值 2000 代替

例: Date d1(1870, 3, 15), 则 d1 为 2000.3.15

d1.set(2112, 3, 4), 则 d1 为 2000.3.4

在构造函数及 set 函数中若出现月、日非法, 均按缺省值 1 处理

例: Date d1(2005, 13, 15), 则 d1 为 2005.1.15

d1.set(2009, 5, 32), 则 d1 为 2009.5.1

7、在++/--/+/-/转换构造中若出现日期超过限定范围的情况, 则按上下限处理, 即若小于 1900.1.1 则设为 1900.1.1, 若大于 2099.12.31, 则设为 2099.12.31

2. TString 类的定义与实现

2.1. 引入

字符串的基本操作都是基于一维字符数组的, 因此其赋值、比较、连接、求长度等方法均要用函数去实现, 且实现过程中必须注意空间是否足够、最后是不是有 '\0' 等情况。

2.2. 实现目标

参照第 5 章 5.6 中 string 类的方法, 用比较简洁易懂的方法来实现字符串的基本操作。

2.3. 要求实现的基本操作

2.3.1. 定义对象并初始化:

- ① TString s1; //s1 为 NULL
- ② TString s1("hello"); //s1 为 "Hello"
- ③ TString s1="hello"; //s1 为 "Hello"
- ④ TString s1("Hello"), s2=s1; //s2 也为 "Hello"
- ⑤ char *s = "Hello";
TString s1 = s; //s1 为 "Hello"
- ⑥ char s[] = "Hello";
TString s1 = s; //s1 为 "Hello"

2.3.2. 输入操作(输入以空格/回车结束)

- ① TString s1;
cin >> s1; //若键盘输入 Hello, 则 s1 得到 "Hello"
- ② TString s1;
cin >> s1; //若键盘输入 Hello 123, 则 s1 得到 "Hello" (空格为分隔符)

2.3.3. 输出操作:

- ① TString s1("hello");
cout << s1; //输出 "hello"
- ② TString s1;
cout << s1; //输出 "<NULL>"

2.3.4. 赋值操作:

- ① TString s1("hello"), s2;
s2=s1; //s2 也为 "Hello"
- ② TString s1("Hello");
s1="Hi"; //s1 为 "Hi", 原 "Hello" 不再保留

2.3.5. 连接操作:

- ① TString s1("tong"), s2("ji"), s3;
s3 = s1+s2; //s3 为"tongji"
s3 = s2+s1; //s3 为"jitong"
- ② TString s1("tong"), s3;
s3 = s1+"ji"; //s3 为"tongji"
s3 = "ji"+s1; //s3 为"jitong"
- ③ TString s1("tong"), s3;
char *s="ji";
s3 = s1+s; //s3 为"tongji"
s3 = s+s1; //s3 为"jitong"
- ④ TString s1("tong"), s3;
char s[]="ji";
s3 = s1+s; //s3 为"tongji"
s3 = s+s1; //s3 为"jitong"

2.3.6. 比较操作:

- ① TString s1="house", s2="horse";
s1 > s2; (包括其它 5 种比较运算) //返回 0/1
- ② TString s1="house";
s1 > "horse"; (包括其它 5 种比较运算) //返回 0/1
"horse" > s2; (包括其它 5 种比较运算) //返回 0/1
- ③ TString s1="house";
char *s="horse"
s1 > s; (包括其它 5 种比较运算) //返回 0/1
s > s2; (包括其它 5 种比较运算) //返回 0/1
- ④ TString s1="house";
char s[]="horse"
s1 > s; (包括其它 5 种比较运算) //返回 0/1
s > s2; (包括其它 5 种比较运算) //返回 0/1

2.3.7. 求串长度:

- ① TString s1("Hello");
cout << s1.length(); //输出为 5
- ② 定义全局函数 TStringLen(TString &);
TString s1("Hello"), s2("123");
char *s3="abcde";
char s4[]="wxyz";
TStringLen(s1+s2); //返回值为 8
TStringLen(s2+s1); //返回值为 8
TStringLen(s1+"pq"); //返回值为 7
TStringLen("pq"+s1); //返回值为 7
TStringLen(s1+s3); //返回值为 10
TStringLen(s3+s1); //返回值为 10
TStringLen(s1+s4); //返回值为 9
TStringLen(s4+s1); //返回值为 9

【要求:】1、部分程序已给出, 按要求实现即可

2、可部分参考作业 9-b3, 但 9-b3 中未考虑申请空间不够的情况, 此处必须考虑

3、P. 318-322 的例 10.4 中, 仅比较运算符的重载有参考价值, 定义及构造等完全不同

4、在操作系统的内存允许的情况下, **均不再考虑空间是否够用, 但也不能浪费空间**

例 1: TString s1("Hello"); 则最多允许申请 6 个字节的空间, **不能多申请**

例 2: TString s1; 则若 s1 = ***** 或 s1 = s1 + ***** 等语句反复出现时, 不能

简单限定空间不超过多少字节，**要无尽利用空间，直到内存空间被耗尽为止**
5、不允许使用系统提供的 string 类

【本次作业占平时成绩分数：13】

【作业要求：】

- 1、**6月2日前**网上提交本次作业(10-b2 除外)，网址：<http://210.75.207.54:7200>
- 2、**6月16日前**网上提交 10-b2
- 3、每题所占平时成绩的具体分值见网页
- 4、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