

补充:**3、TString 类的定义与实现****3.1. 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。

3.2. 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。

3.3. 要求实现的基本操作**3.3.1. 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; //s1/s2 均为"Hello"
- ⑤ char *s = "Hello";
TString s1 = s; //s1 为"Hello"
- ⑥ char s[] = "Hello";
TString s1 = s; //s1 为"Hello"

3.3.2. 输入操作: (以空格/回车做为输入结束)

- 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)

3.3.3. 输出操作:

- ① TString s1("hello");
cout << s1; //输出"hello"
- ② TString s1;
cout << s1; //输出"<NULL>", 说明: 这是对 NULL 进行的特殊处理

3.3.4. 取字符串操作: (将 TString 中的内容以 char *方式返回, 只读, 不可更改内容)

- ① TString s1("hello");
printf("%s", s1.c_str()); //输出为"hello"

3.3.5. 赋值操作:

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

3.3.6. 连接操作: (运算符+表示字符串连接后赋值给另一个串)

- 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"
```

⑤ TString s1("Hello"), s3
char c = '!';
s3 = s1 + c; //s3 为"Hello!"

⑥ TString s1("ello"), s3
s3 = 'H' + s1; //s3 为"Hello"

3.3.7. 自连接操作：（运算符+=表示字符串连接后赋值给自己）

① TString s1("tong"), s2("ji");
s1 += s2; //s1 为"tongji"

② TString s1("tong");
s1 += "ji"; //s1 为"tongji"

③ TString s1("tong");
char *s="ji";
s1 += s; //s1 为"tongji"

④ TString s1("tong");
char s[]="ji";
s1 += s; //s1 为"tongji"

⑤ TString s1("Hello");
char c = '!';
s1 += c; //s1 为"Hello!"

3.3.8. 删除操作：（运算符-表示从字符串中删除另一个字符串/一个字符后赋值给另一个串）

① TString s1("tongji"), s2("ji"), s3;
s3 = s1 - s2; //s3 为"tong"

② TString s1("tongji"), s3;
s3 = s1 - "ji"; //s3 为"tong"

③ TString s1("tongji"), s3;
char *s="ji";
s3 = s1 - s; //s3 为"tong"

④ TString s1("tongji"), s3;
char s[]="ji";
s3 = s1 - s; //s3 为"tong"

⑤ TString s1("tongji"), s3;
char c1 = 'j', c2 = 'i';
s3 = s1 - c1 - c2; //s3 为"tong"

3.3.9. 自删除操作：（运算符-=表示从字符串中删除另一个字符串/一个字符后赋值给自己）

① TString s1("tongji"), s2("ji");
s1 -= s2; //s1 为"tong"

② TString s1("tongji");
s1 -= "ji"; //s1 为"tong"

③ TString s1("tongji");
char *s="ji";
s1 -= s; //s1 为"tong"

④ TString s1("tongji");
char s[]="ji";
s1 -= s; //s1 为"tong"

⑤ TString s1("tongji");
char c1 = 'j', c2 = 'i';

- 3.3.10. 复制操作：（运算符*表示将字符串自身复制若干倍后赋值给另一个串）
- ① TString s1("tong"), s2;
 s2 = s1*2; //s2 为"tongtong"
 - ② TString s1, s2;
 s2 = s1*5; //s2 为<NULL> "
- 3.3.11. 自复制操作：（运算符*表示将字符串自身复制若干倍后赋值给自己）
- ① TString s1("tong"), s2;
 s1 *= 2; //s1 为"tongtong"
 - ② TString s1;
 s1 *= 5; //s1 为<NULL> "
- 3.3.12. 反转操作：（运算符!表示将字符串反转后赋值给另一个串）
- ① TString s1("tong"), s2;
 s2 = !s1; //s2 为"gnot", s1 仍为"tong"
 - ② TString s1;
 s2 = !s1; //s2 为<NULL> "
- 3.3.13. 比较操作：（按 strcmp 的规则返回即可）
- 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
- 3.3.14. 求串长度：（按 strlen 的规则返回即可）
- ① TString s1("Hello");
 cout << s1.length(); //输出为 5
 - ② 定义全局函数 TStringLen(const 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
- 3.3.15. 取串中某个字符的值/给串中的某个字符赋值：（按字符数组的规则即可）
- ① TString s1("hello");
 cout << s1[1]; //输出为 e
 - ② TString s1("hello");

```
s1[0] -= 32;
cout << s1;          //输出为 Hello
```

【要求:】1、程序由三个文件组成, 各文件的说明如下:

10-b3.h: 给出 TString 类的定义及其它需要的定义

10-b3.cpp: 给出 TString 类的所有成员函数的实现及其它需要的全局函数的实现

10-b3-main.cpp: 在 main 函数中给出了 TString 类的测试用例, 不准修改, 不需要提交, 检查作业时会替换本文件

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. 307-311 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; 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**

例 3: TString s1("Hello"); 若 s1 = s1-"He"; 则 s1 要减为 4 字节

5、实现过程**不允许**使用系统提供的 string 类, 但可以使用<cstring>中字符串函数

6、给出 Windows 和 Linux 的可执行文件供参考

7、最后的 100MB 累加测试完成后, 用“任务管理器”查看的内存占用**不能超过 115MB**

8、最后的 100MB 累加测试, +=方式的完成时间, **不允许**超过+方式的 75%

```
D:\Workspace\高级语言程序设计\部分作业(2017级-VS2017)\Debug\第10章-TString类(2017级).exe
s1已有长度: 99.2827 MB字节, 本次增加 50245 字节
s1已有长度: 99.343 MB字节, 本次增加 63276 字节
s1已有长度: 99.3832 MB字节, 本次增加 42148 字节
s1已有长度: 99.4332 MB字节, 本次增加 52408 字节
s1已有长度: 99.4763 MB字节, 本次增加 45222 字节
s1已有长度: 99.5221 MB字节, 本次增加 47968 字节
s1已有长度: 99.5579 MB字节, 本次增加 37541 字节
s1已有长度: 99.5934 MB字节, 本次增加 37224 字节
s1已有长度: 99.6253 MB字节, 本次增加 33444 字节
s1已有长度: 99.6763 MB字节, 本次增加 53507 字节
s1已有长度: 99.7123 MB字节, 本次增加 37803 字节
s1已有长度: 99.7546 MB字节, 本次增加 44309 字节
s1已有长度: 99.805 MB字节, 本次增加 52832 字节
s1已有长度: 99.8606 MB字节, 本次增加 58310 字节
s1已有长度: 99.9167 MB字节, 本次增加 58821 字节
s1已有长度: 99.9617 MB字节, 本次增加 47157 字节
s1已有长度: 100.023 MB字节, 本次增加 64736 字节, 总用时 471.688秒, 本次1MB用时 9.282秒
time=471.719
内存分配到达满100MB, 测试结束
本次测试耗时 471.719秒
内存性能测试(s1=s1+str方式)
老师的机器运行VS2017编译的程序, 大约耗时450-500秒
老师的机器运行Linux编译的程序, 大约耗时160-180秒
【说明】: 只有相同编译器下的运行时间才有可比性
老师的机器为Intel(R) Core(TM) i5-4210U CPU @1.70GHz 2.40GHz (虚拟机为1CPU, 2G内存)
如果时间相差太大, 除CPU的性能差异外, 还有可能是算法问题
打开任务管理器查看程序内存占用情况<不允许超过115MB>...
按回车键继续。
```

```
s1已有长度: 99.867 MB字节, 本次增加 37307 字节
s1已有长度: 99.9062 MB字节, 本次增加 41077 字节
s1已有长度: 99.9552 MB字节, 本次增加 51383 字节
s1已有长度: 100.003 MB字节, 本次增加 50589 字节, 总用时 164.767秒, 本次1MB用时 2.668秒
time=164.767
内存分配到达满100MB, 测试结束
本次测试耗时 164.767秒
内存性能测试(s1=s1+str方式)
老师的机器运行VS2017编译的程序, 大约耗时450-500秒
老师的机器运行Linux编译的程序, 大约耗时160-180秒
【说明】: 只有相同编译器下的运行时间才有可比性
老师的机器为Intel(R) Core(TM) i5-4210U CPU @1.70GHz 2.40GHz (虚拟机为1CPU, 2G内存)
如果时间相差太大, 除CPU的性能差异外, 还有可能是算法问题
按回车键继续。
```

```
D:\WorkSpace\高级语言程序设计\部分作业(2017级-VS2017)\Debug\第10章-TString类(2017级).exe
s1已有长度: 99.2337 MB字节, 本次增加 44633 字节
s1已有长度: 99.2944 MB字节, 本次增加 63555 字节
s1已有长度: 99.3272 MB字节, 本次增加 34385 字节
s1已有长度: 99.3788 MB字节, 本次增加 54107 字节
s1已有长度: 99.4308 MB字节, 本次增加 54524 字节
s1已有长度: 99.4685 MB字节, 本次增加 39597 字节
s1已有长度: 99.5042 MB字节, 本次增加 37436 字节
s1已有长度: 99.543 MB字节, 本次增加 40699 字节
s1已有长度: 99.5902 MB字节, 本次增加 49452 字节
s1已有长度: 99.6462 MB字节, 本次增加 58742 字节
s1已有长度: 99.7085 MB字节, 本次增加 65331 字节
s1已有长度: 99.7599 MB字节, 本次增加 53910 字节
s1已有长度: 99.8037 MB字节, 本次增加 45946 字节
s1已有长度: 99.8606 MB字节, 本次增加 59626 字节
s1已有长度: 99.9221 MB字节, 本次增加 64524 字节
s1已有长度: 99.9784 MB字节, 本次增加 58994 字节
s1已有长度: 100.032 MB字节, 本次增加 56687 字节, 总用时 219.297秒, 本次1MB用时 3.953秒
time=219.328
内存分配到达满100MB, 测试结束
本次测试耗时 219.328秒
内存性能测试(s1+=str方式)
老师的机器运行VS2017编译的程序, 大约耗时200-240秒
老师的机器运行Linux编译的程序, 大约耗时100-120秒
【说明】: 只有相同编译器下的运行时间才有可比性
老师的机器为Intel(R) Core(TM) i5-4210U CPU @1.70GHz 2.40GHz (虚拟机为1CPU, 2G内存)
如果时间相差太大, 除CPU的性能差异外, 还有可能是算法问题
打开任务管理器查看程序内存占用情况<不允许超过115MB>...
按回车键继续
```

```
s1已有长度: 99.8837 MB字节, 本次增加 55783 字节
s1已有长度: 99.9216 MB字节, 本次增加 39806 字节
s1已有长度: 99.9788 MB字节, 本次增加 59886 字节
s1已有长度: 100.013 MB字节, 本次增加 35482 字节, 总用时 100.611秒, 本次1MB用时 1.874秒
time=100.611
内存分配到达满100MB, 测试结束
本次测试耗时 100.611秒
内存性能测试(s1+=str方式)
老师的机器运行VS2017编译的程序, 大约耗时200-240秒
老师的机器运行Linux编译的程序, 大约耗时100-120秒
【说明】: 只有相同编译器下的运行时间才有可比性
老师的机器为Intel(R) Core(TM) i5-4210U CPU @1.70GHz 2.40GHz (虚拟机为1CPU, 2G内存)
如果时间相差太大, 除CPU的性能差异外, 还有可能是算法问题
按回车键继续
```

任务管理器					
文件(F) 选项(O) 查看(V)					
进程 性能 应用历史记录 启动 用户 详细信息 服务					
名称	状态	7% CPU	51% 内存	1% 磁盘	0% 网络
应用 (6)					
> Microsoft Visual Studio 2017...		0%	240.4 MB	0 MB/秒	0 Mbps
> Microsoft Word (32 位) (2)		0.4%	99.2 MB	0 MB/秒	0 Mbps
> TIM (32 位) (2)		0.1%	140.6 MB	0 MB/秒	0.1 Mbps
> Windows 资源管理器 (2)		0.2%	39.7 MB	0 MB/秒	0 Mbps
> 第10章-TString类(2017级).exe...		0%	110.6 MB	0 MB/秒	0 Mbps
> 任务管理器		1.3%	26.8 MB	0 MB/秒	0 Mbps
后台进程 (109)					
> [设置图标]		0.3%	1.8 MB	0 MB/秒	0 Mbps
> 64-bit Synaptics Pointing Enh...		0%	0.1 MB	0 MB/秒	0 Mbps
> AcroTray (32 位)		0%	0.1 MB	0 MB/秒	0 Mbps
> Antimalware Service Execu...		0%	70.3 MB	0 MB/秒	0 Mbps
> Application Frame Host		0%	0.2 MB	0 MB/秒	0 Mbps
< 简略信息(I) 结束任务(E)					

【编译器要求:】

		编译器VS	编译器CB	编译器DEV	编译器Linux
10-b3.h	TString类-头文件	Y	/	/	Y
10-b3.cpp	TString类-实现	Y	/	/	Y

【作业要求:】

- 1、**6月5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