第9周上机作业

实验指导书

p57 3 改错题

p63 2调试示例

p65 3 改错题

p74 2调试示例

p76 编程题（2）

课后作业

1. 课后作业请在规定时间之前提交到ftp相应文件夹，地址为ftp://public.sjtu.edu.cn，用户名:chuxing，密码:chuxing7，每次作业都会建文件夹，命名为数字，代表第几次。**千万**不要放在之前的作业文件夹，或者放在别的同学的作业目录下。本次作业提交**截止**时间为11月13日20：00
2. 提交方式为将所要求文件放在一个文件夹内，**不要**压缩，文件夹名为学号，每道题目也只要用数字1，2，3命名就行，避免出现汉字。例如：张三的学号为5140000001，则张三每次作业完成后将所有相关文件放在5140000001文件夹内，然后将该文件夹上传至ftp相应的文件夹内（请注意是第几次作业的目录，以及是哪个班级）
3. 本次作业2-10题均需提供cpp文件，因此本次作业每个同学提供9个文件
4. 做作业有问题可以与其他同学讨论，但请勿复制他人程序，也可以发送邮件至助教邮箱chuxing@sjtu.edu.cn，wfwzy2012@sjtu.edu.cn，insanelung@sjtu.edu.cn或我的邮箱进行讨论。

课后作业

1阅读理解以下程序，上机运行，思考一下为什么是这样的结果（不用提交）

#include <iostream>

#include <typeinfo>

using namespace std;

int main()

{

char\* proverb="An apple a day keeps the doctor away";

cout << proverb << '\t' << sizeof (proverb) << '\t'<< typeid(proverb).name() << '\n' ;

cout << proverb+1 << '\t' << sizeof (proverb+1) << '\t'<< typeid(proverb+1).name() << '\n' ;

cout << proverb[0] << '\t' << sizeof (proverb[0]) << '\t'<< typeid(proverb[0]).name() << '\n' ;

cout << \*proverb << '\t' << sizeof( \*proverb) << '\t'<< typeid(\*proverb).name() << '\n' ;

cout << proverb[1] << '\t' << sizeof( proverb[1]) << '\t'<< typeid(proverb[1]).name() << '\n' ;

cout << proverb[0]+1 << '\t' << sizeof( proverb[0]+1) << '\t'<< typeid(proverb[0]+1).name() << '\n' ;

return 1;

}

int main()

{

char\* proverb[]={"An apple a day keeps doctor away","Two apples a day are better !"};

cout << proverb << '\t' << sizeof( proverb) << '\t'<< typeid(proverb).name() << '\n' ;

cout << proverb[0] << '\t' << sizeof (proverb[0]) << '\t'<< typeid(proverb[0]).name() << '\n' ;

cout << \*proverb << '\t' << sizeof( \*proverb) << '\t'<< typeid(\*proverb).name() << '\n' ;

cout << \*proverb[1] << '\t' << sizeof( \*proverb[1]) << '\t'<< typeid(\*proverb[1]).name() << '\n' ;

cout << (\*proverb)[0] << '\t' << sizeof ((\*proverb)[0]) << '\t'<< typeid((\*proverb)[0]).name() << '\n' ;

cout << (\*proverb)[1] << '\t' << sizeof ((\*proverb)[1]) << '\t'<< typeid((\*proverb)[1]).name() << '\n' ;

cout << \*\*proverb << '\t' << sizeof( \*\*proverb) << '\t'<< typeid(\*\*proverb).name() << '\n' ;

cout << proverb[1] << '\t' << sizeof (proverb[1]) << '\t' << typeid(proverb[1]).name() << '\n' ;

cout << \*proverb[1] << '\t' << sizeof (\*proverb[1]) << '\t' << typeid(\*proverb[1]).name() << '\n' ;

cout << proverb << '\t' << sizeof (proverb) << '\t'<< typeid(proverb).name() << endl;

cout << proverb+1 << '\t' << sizeof (proverb+1) << '\t'<< typeid(proverb+1).name() <<endl;

cout << \*(proverb+1) << '\t' << sizeof (\*(proverb+1)) << '\t'<< typeid(\*(proverb+1)).name() <<endl;

cout << proverb[0]+1 << '\t' << sizeof( proverb[0]+1) << '\t'<< typeid(proverb[0]+1).name() << '\n' ;

cout << \*(proverb[0]+1) << '\t' << sizeof (\*(proverb[0]+1) ) << '\t' << typeid(\*(proverb[0]+1)).name() << '\n' ;

cout << proverb[0][1] << '\t' << sizeof( proverb[0][1]) << '\t' << typeid(proverb[0][1]).name() << '\n' ;

cout << (\*proverb)[1] << '\t' << sizeof ((\*proverb)[1] ) << '\t' << typeid((\*proverb)[1] ).name() << '\n' ;

return 1;

}

2 p58 编程题（2）

3 设计程序，将第5章的选择排序程序改写为函数模板，并编写一个main函数演示其既能够对整数数组，也能够对浮点数数组进行排序

4 p59 编程题（3）

5 利用递归实现二分搜索

6在数学中，集合不允许出现重复元素，而C++中的数组允许元素相同，设计程序利用数组实现数学中的集合概念，要求提供以下函数1）新集合的生成（将一个数组中的重复元素删除）；2）增加集合中的元素；3）查找集合中是否有某个元素；4）求两个集合的交集；5）求两个集合的并集。在以上函数的基础上编写一个main函数，通过调用你所设计的函数来展现这些集合函数的功能。（元素都为整数，并假设任何运算数组规模都不会超过MAXSIZE）

7 实现两个正浮点数的精确加法。参与运算的浮点数及计算结果的有效位数最多为128位。

8 p67 编程题（2）

9 p67 编程题（3）

10 p76 编程题（1）