

实验 1：计算器工具

1. 实验目的

- 1) 掌握用户界面程序的开发技巧；
- 2) 掌握函数库的使用方法；
- 3) 练习提高程序健壮性；

2. 实验环境

- 1) 台式机 1 台；
- 2) Windows 操作系统；
- 3) C++或 JAVA 编程环境；

3. 实验内容

编写一个计算器工具，能够实现 Windows 附件中科学型计算器的功能。

4. 实验步骤

- 1) 获取数据的方式：

通过输入一个字符串来表示需要进行计算的算术式，通过对字符串进行解析从而获取运算符和数字。数字由两部分组成，即小数点前的部分和小数点后的部分。以小数点位分界线进行读入。

小数点之前的数字可以直接按照 $(ans * 10 + num)$ 的公式进行计算。小数点之后的数字还需要额外计算 **base**。当一个数字读取结束之后，需要将小数点后的数字除以 10 的 **base** 次方，再加上小数点前的部分，以此实现数字的读入。

考虑到符号的有限，可以直接对符号进行特判（加减乘除等等）。

详细实现参考函数 `readnum`。

- 2) 对包含加减乘除括号的表达式进行运算：

不同的运算符具有不同的优先级，在执行一个运算符之前一些运算符必须先被执行。比如当前方具有乘运算时，加运算的出现会迫使乘运算执行再将其产生的数字压回数字栈，在此之后加运算符才会入操作符栈。后括号的出现会强迫前方的运算全部完结，在将其产生的数字压回数字栈后，后括号会和操作符栈的前括号相抵消，它不入栈。

按照以上的做法，当每一个字符都入栈后，就可以按顺序从数字栈和操作符栈中获取数据，并将其当做后缀表达式进行操作。可以每一次都从操作符栈和数字栈中获得栈顶，再去除栈顶进行运算直到栈空，最后留下的数据就是运算结果。

详细实现参考函数 `cal`。

- 3) 拓展：对 `sin`、`cos`、`tan` 函数的嵌套表达式进行运算：

按照通常的写法，所有的二元运算只要赋值于正确的优先级，全都可以根据设计思路二加以实现。

首先是 `sin(x)` 的计算方法，正确解法是用泰勒展开，函数库中已经具有三角

函数。但接下来会有例如 $\sin(\cos(x))$ 的表达式。显然，做法是先计算 $\cos(x)$ ，再将 $\cos(x)$ 当做 x 计算 $\sin(x)$ 。同理可推， $\sin(\cos(x)*3.14)$ 的计算方法就是将 $\cos(x)*3.14$ 当做 x 来计算 $\sin(x)$ 。

对一个字符串进行括号解析，递归地找到括号最里层的表达式，对其利用 `cal` 函数进行计算，再回溯地求解外层的括号。以此完成嵌套三角函数表达式的计算。

详细实现参考函数 `deal`。

4) 拓展：对包含加、减、乘、除、括号、三角函数的表达式进行运算：

根据设计思路三的方法，所有的嵌套的三角函数式都可以被运算；根据设计思路二的方法，所有的加、减、乘、除、括号的表达式都可以被运算。可以将其进行整合。

对于一个输入的字符串，对其进行分解再合并运算。例如 $\sin(x)*\cos(3*\tan(2.6))$ ，可以将其分解为 $\sin(x)$ 和 $\cos(3*\tan(2.6))$ ，根据设计思路三，可以分别计算出 $\sin(x)$ 的值为 $x1$ ， $\cos(3*\tan(2.6))$ 的值为 $x2$ 。这样所产生的表达式就是 $x1*x2$ 。对其进行合并，根据设计思路二，就可以求出其结果。

详细实现参考函数 `solve`。

5) 注意特殊的输入

- a) 程序要求能够辨别负号和减运算。
- b) 字符串的开头可能是没有意义的前括号。
- c) 浮点数计算时会出现不可避免的精度损失。

5. 实验方式

每位同学独立上机编程实验，实验指导教师现场指导。

6. 参考内容

C++标准库中的数学函数

7. 相关软件下载

无

8. 实验报告要求

- 1) 需求分析
- 2) 概要设计
 - a) 抽象数据类型
 - b) 算法
- 3) 详细设计
 - 程序代码（注释）
- 4) 调试分析
 - 调试过程中所做的工作，时间复杂度等
- 5) 测试结果
 - 输入数据和输出数据示例
- 6) 说明（如果有）

- 7) 编程语言：C 语言或 C++语言
- 8) 实习报告提交方式：下次上机前，将实验报告和源程序压缩成一个 rar 文件，文件名称为学号+班级+姓名+第几次实验, 最后一次上机下课前交实验报告。实验报告作为本课程的平时成绩一部分。最后一次实验报告下课前交。

实验 2：模拟旅馆管理系统中的床位分配和回收

1. 实验目的

- 1) 熟悉一种模拟类型算法编写；
- 2) 掌握自定义数据结构满足算法需求；
- 3) 练习界面程序编写；

2. 实验环境

- 1) 台式机 1 台；
- 2) Windows 操作系统；
- 3) C++或 JAVA 编程环境；

3. 实验内容

某旅馆有若干等级的房间，每个等级有若干房间和床位，请模拟旅馆管理系统的床位分配和回收功能，设计能为单个和合住旅客分配床位并且在旅客离店时回收床位的系统并模拟一个。

4. 实验步骤

- 1) 数据输入：
房间类别、数量、各房间床位数量、计费标准、入住旅客日期、所需房间等级、无房间是否愿意更换等级。

- 2) 结果输出：

分配成功的客人信息、不成功的信息（客人不愿意更换等级和无房间），一个月內旅馆的收入。

5. 实验方式

每位同学独立上机编程实验，实验指导教师现场指导。

6. 参考内容

最终结果：

- 1). 记录房间种类、数量、床位数量、房间花费、房间等级
- 2). 记录客人的入住日期、要求的房间等级、是否愿意更换房间等级
- 3). 使用恰当的策略分配给顾客最合适的房间
- 4). 采用线段树的数据结构高效解决此问题
- 5). 输出分配成功的客人信息和不成功的信息

6). 查询任意时间区间内的总收入

7. 相关软件下载

无

8. 实验报告要求

- 1) 需求分析
- 2) 概要设计
 - a) 抽象数据类型
 - b) 算法
- 3) 详细设计
程序代码（注释）
- 4) 调试分析
调试过程中所做的工作，时间复杂度等
- 5) 测试结果
输入数据和输出数据示例
- 6) 说明（如果有）
- 7) 编程语言：C 语言或 C++语言
- 8) 实习报告提交方式：下次上机前，将实验报告和源程序压缩成一个 rar 文件，文件名称为学号+班级+姓名+第几次实验, 最后一次上机下课前交实验报告。实验报告作为本课程的平时成绩一部分。最后一次实验报告下课前交。

实验 3：导航软件

1. 实验目的

- 1) 掌握图的存储结构;
- 2) 掌握图的基本算法;
- 3) 练习界面程序编写;

2. 实验环境

- 1) 台式机 1 台;
- 2) Windows 操作系统;
- 3) C++或 JAVA 编程环境;

3. 实验内容

编写一个天津大学新校区的导航软件,要求给出校内所有建筑和道路并具有道路是否可通行的标志,软件可以给出两点间机动车和自行车的最短路线。

4. 实验步骤

1) 对问题建模

本题是最短路径题。最短路径的算法包括但不限于 Dijkstra、Bellman-ford、Floyd、SPFA。考虑到数据范围,即天津大学的标志性建筑物不超过 100,无论用以上那种方法都可以解决此问题。可以使用以邻接矩阵为基础的 Dijkstra 算法,并利用数组标记出行走路线。

为了区别开机动车和非机动车的路径,可以直接建立两份不同的图模型,并通过对每一个图模型添加道路来实现图的建立。由于输入的是名称,需要将名称字符串用数字表示,因此利用下标递增的哈希方法来建立联系。同时,也利用字符串数组来实现下标到名称的映射。

2) 处理问题中的小细节

- a). 考虑输入的起点就是终点的情况
- b). 考虑构图时未提到的名称在查询时被输入的情况
- c). 考虑构图输入边时提到的点数少于总点数的情况

5. 实验方式

每位同学独立上机编程实验,实验指导教师现场指导。

6. 参考内容

最终结果:

- 1). 使用 Dijkstra 计算出任意两点间的最短路径
- 2). 打印出为到达目的地需要经过的地点和每一段路程的长度
- 3). 将机动车道和非机动车道分开运算

7. 相关软件下载

无

8. 实验报告要求

- 1) 需求分析
- 2) 概要设计
 - a) 抽象数据类型
 - b) 算法
- 3) 详细设计
程序代码（注释）
- 4) 调试分析
调试过程中所做的工作，时间复杂度等
- 5) 测试结果
输入数据和输出数据示例
- 6) 说明（如果有）
- 7) 编程语言：C 语言或 C++ 语言
- 8) 实习报告提交方式：下次上机前，将实验报告和源程序压缩成一个 rar 文件，文件名称为学号+班级+姓名+第几次实验, 最后一次上机下课前交实验报告。实验报告作为本课程的平时成绩一部分。最后一次实验报告下课前交。

实验 4：压缩软件

1. 实验目的

- 1) 掌握文件处理的方法；
- 2) 掌握两种压缩算法；
- 3) 比较不同方法的时间复杂度；

2. 实验环境

- 1) 台式机 1 台；
- 2) Windows 操作系统；
- 3) C++或 JAVA 编程环境；

3. 实验内容

编写一个压缩软件，选择两种压缩算法（自选），将用户提交的文件实现压缩并提示用户压缩率及提示用户按压缩率高的算法压缩；该软件还可通过文件格式识别文件是否是本软件压缩并按压缩时的算法解压。

4. 实验步骤

- 1). 实现两种压缩算法
- 2). 实现压缩时的进度条显示
- 3). 实现对自己编码的压缩文件进行解压
- 4). 对比压缩率并提示用户按高压缩率的方式压缩
- 5). 输入路径即可压缩任意格式文件
- 6). 压缩或解压后的文件与之前的文件在同一目录下

5. 实验方式

每位同学独立上机编程实验，实验指导教师现场指导。

6. 参考内容

1)资料搜集

可供选择的算法包括：RLE、LZW、LZ77、LZ78、Huffman、Arithmetic 这几种。对 ZIP 进行解析之后可知一个压缩软件利用的不只是一种压缩算法，而是一种反复的压缩，并且理论下限是这块数据的信息熵。

在本题中可以使用 LZ77 和 HuffmanTree 的压缩方法。

2)LZ77 的实现思路

LZ77 的基本想法是，设立两个指针，第一个指针成为文件指针，指向需要匹配的文件位置，第二个指针成为滑动窗口指针，指向滑动窗口开始的地方。当文件指针指向的数据能够和滑动窗口内的数据相匹配时，就开始累积长度，一旦长度超过 3，就认为这些匹配的长度被 (distance, length) 对替换具有压缩的效果，否则，我们认为其是一个 literal。为了区别它们，需要利用一比特来标记

接下来的数据是 (distance, length) 对还是 literal。

在解压的时候，依次读入数据，如果读到的是 (distance, length) 对，就从之前的 distance 复制 length 数据再去除标志位，否则直接去除标志位。

值得一提的是，LZ77 压缩时对位的读取需要从高往低读，称之为高低读取，这将区别于下文中将提到的低高读取。

3) HuffmanTree 的实现思路

HuffmanTree 的基本思想是，如果能够知道整个文件每个字符出现的频率，并按重要性进行霍夫曼编码，并将其直接写入文件，就可以实现压缩。而进行解码的时候，由于霍夫曼编码的前缀编码特性，可以直接进行翻译。

HuffmanTree 的建树是把已经统计完每个字符出现的频率的字典中的每一个字符看做单节点的树，以权重为标准构造最小堆，再不断地对子树进行合并构造树的。而树的销毁，则是按照先序遍历类似的方式进行递归地销毁的。在一棵树构造完毕后，需要对每一个叶子节点赋予一个二进制编码，可以使用字符串来表示，利用先序遍历的方式递归整棵树构造字典。

需要强调的是因为解码时 HuffmanTree 需要得到所有编码，因而不能再简单地将字符补零写入，而是需要特别记录最后有用的字符数（相比较而言，LZ77 记录的是总字节数）。

4) 对文件类的操作

本题的难点是考察对于文件流的操作，包括二进制读入、写出和文件指针的移动。头文件 <fstream> 包含了所有文件类。打开文件的函数 fopen(文件名, “属性”), 属性可选 rb 或 wb, 分别代表二进制读入和二进制写入。函数 fread(内存地址, 大小, 数量, 文件指针) 可以向内存地址读入数量个大小的文件内容。函数 fwrite(内存地址, 大小, 数量, 文件指针) 可以向内存中读出数量个大小的文件内容。函数 fgetc(文件指针) 可以从文件指针处读入一字节数据。函数 feof(文件指针) 判定文件是否读入结束。函数 fseek(文件指针, 偏移量, 定位) 可以对文件指针进行 O(1) 复杂度的移动，其中定位包括 SEEK_SET, SEEK_CUR, SEEK_END, 分别表示指向文件开头, 指向文件当前和指向文件末尾。函数 ftell(文件指针) 可以得到当前文件指针所指向位置相对于文件头的偏移量。fclose(文件指针) 可以关闭已经打开的文件。

5) 对比特位的操作

本题的第二个难点就在于对比特位的操作，因为 c++ 中最小的单位是 char, 或 bool。如果用 JAVA 自然可以解决这个问题，但鉴于其他题目都是 c++ 编写，为保证代码一致性，可以自行手动解决这个问题。

LZ77 的高低读取：利用一个 flag 为标记，通过 & 或 | 的位运算从高往低读取一个二进制数字。参考函数 LZ77::ReadBit。

HuffmanTree 的低高读取：利用一个 bit 为标记，读出的每一位都排在二进制的末尾，将其左移一位后可以读取下一个 bit。参考函数 Tree::WriteCode。

事实上，低高写入和高低写入本质是一样的，都是通过位运算对比特位进行编码，在大多数情况下写入方式是可以互换的。在本例中使用两种不同方式的原因仅仅是因为容易理解和思考。

7. 相关软件下载

无

8. 实验报告要求

- 1) 需求分析
- 2) 概要设计
 - a) 抽象数据类型
 - b) 算法
- 3) 详细设计
程序代码（注释）
- 4) 调试分析
调试过程中所做的工作，时间复杂度等
- 5) 测试结果
输入数据和输出数据示例
- 6) 说明（如果有）
- 7) 编程语言：C 语言或 C++语言
- 8) 实习报告提交方式：下次上机前，将实验报告和源程序压缩成一个 rar 文件，文件名称为学号+班级+姓名+第几次实验, 最后一次上机下课前交实验报告。实验报告作为本课程的平时成绩一部分。最后一次实验报告下课前交。

实验 5：排课系统

1. 实验目的

- 1) 掌握多种存储结构;
- 2) 练习界面程序编写;
- 3) 练习复杂问题的解决方法;

2. 实验环境

- 1) 台式机 1 台;
- 2) Windows 操作系统;
- 3) C++或 JAVA 编程环境;

3. 实验内容

设计一个排课系统，要求从文本文件导入教室情况(大小，数量)、培养方案(课程、学生年级、人数、教师)、约束条件(有些约束从培养方案中可知，如一个教师可以教授两门课不可以安排在同一时间上，其他包括课程先修关系，教师的特殊情况如不能上上午一二节课等)，系统最终生成课表或给出无法排课的错误提示。

4. 实验步骤

1) 对问题建模

排课系统是一个 NP 问题，这意味着它不能在多项式的时间复杂度之内被计算出来。这是一个很好的提示，因为对 NP 问题的解决方案只有几种，比如状态压缩的动态规划，投掷随机数的概率算法，以及这道题可以使用的 DFS 搜索。

为了满足上述最终结果中提到的一系列条件，需要在不同的数据之间建立联系。需要用到面向对象的思想。

a). 抽象出教室类，它具有容量、可用和编号这三个属性。

b). 抽象出课程类，它具有开课老师的编号、选课学生的数目、选课学生的年级、课时、课程编号、使用的教室编号、是否必修课的选项、课程名称、最近一次分配的时间这多个属性。

c). 抽象出老师类，它具有老师名称、编号、教授课程的编号、不能上课的时间这几个属性。

d). 抽象出时间表类，存储不同时间的课程，用于结果的输出。

2) 算法核心

重点算法是 DFS，它的本质是在内存内部遍历一棵二叉树。在这个问题中，这棵树的左孩子表示在这个时间分配这门课程，右孩子表示在这个时间不分配这门课程。当满足条件时，会往左子树的方向走，如果失败就回退父节点，再往右子树走，否则只往右子树的方向走，这是第一处剪枝。第二处剪枝是当接下来所

有时间的可利用的房间的和都没有还需要分配的课程数目多的时候直接进行回溯。最后一个剪枝是在根节点处进行的，对数据进行预处理，特别判断绝对不能排序的情况是否存在。

本题的依赖关系中需要考虑到课程的拓扑结构，没有必要真的进行拓扑排序，只需要统计入度即可，需要特别注重对度的处理。

如果所有课程都已经满足课时，就认为排课已经结束。

5. 实验方式

每位同学独立上机编程实验，实验指导教师现场指导。

6. 参考内容

几个操作技巧

1). 对左子树展开时需要计算出下一个状态的表示方法，可以把所有状态当做一个数，这个数的各个位满足不同的进制，计算出下一个状态即是对这个数进行加一操作。记录下这个数的各个位置，在回溯时可以重新得到当前状态。

2). 输出课表的时候格式有特定函数 `setw` 和 `setfill` 可以进行控制，需要注意对没有课的部分打满空格，否则课表会出现偏移。

3). 前面的约束是尽可能错开课程，可以给课程赋予一个属性表示课程在最近的时间被使用。如果课程一直向左子树展开，那么这个时间会一直被标记为上一天。一旦左子树展开失败，这个时间会被标记为绝对不可能存在的时间，即-2。这意味着在下一步操作的时候他不具有时间约束，因为下一步的时间不可能与-2匹配的上。按照这个方法，就可以实现在保证能够排课的前提下进行时间的尽可能分散。

7. 相关软件下载

无

8. 实验报告要求

- 1) 需求分析
- 2) 概要设计
 - a) 抽象数据类型
 - b) 算法
- 3) 详细设计
程序代码（注释）
- 4) 调试分析
调试过程中所做的工作，时间复杂度等
- 5) 测试结果
输入数据和输出数据示例
- 6) 说明（如果有）
- 7) 编程语言：C 语言或 C++语言
- 8) 实习报告提交方式：下次上机前，将实验报告和源程序压缩成一个 rar 文件，文件名称为学号+班级+姓名+第几次实验,最后一次上机下课前交实验报告。实验报告作为本课程的平时成绩一部分。最后一

次实验报告下课前交。