

# 《程序设计课程设计》 课设指导书



软件学院

二〇二四年

# 目 录

一	前言 .....	1
1	目的与任务 .....	1
2	培养目标 .....	1
3	选用教材及主要参考书 .....	1
二	要求 .....	3
1	采用项目小组开发模式 .....	3
2	课设报告要求 .....	4
三	备选题目 .....	6
1	算法过程可视化系统 .....	6
2	飞机大战游戏单机版 .....	8
四	课设报告参考样例 .....	10
五	考核标准 .....	20
1	成绩评定标准 .....	20
2	项目组成员考核 .....	20
3	项目组成员个人成绩考核 .....	21

## 一 前言

### 1 目的与任务

《程序设计课程设计》是软件工程专业的重要实践性课程。此课程目的目的在于培养学生具备需求分析、设计、开发、测试和应用的职业技能，具备综合运用计算机软硬件理论和技术，分析和解决计算机领域相关工程问题的能力。课程将理论知识和实际应用问题进行有机结合，提高学生程序设计、程序调试及项目开发能力，为后续课程：操作系统、软件工程，编译原理等课程的学习奠定必要的实践基础。

《程序设计课程设计》是利用数据结构与算法、离散结构、程序设计语言理论、实验课中学到的编程知识和编程技巧，通过布置具有一定难度和工作量的综合课程设计题目，利用 C/Java/Python 等语言作为开发工具，使学生通过课程设计掌握高级编程语言的知识和编程技术，掌握程序设计的思想和方法，初步具备利用计算机求解实际问题的能力。给定两个题目：**【1】**算法可视化系统、**【2】**飞机大战单机版游戏。学生可根据自身情况选择一个题目完成。如有自备题目，必须经指导教师审查通过方可实施，否则不计入成绩。

通过《程序设计课程设计》课程的学习，能够帮助学生加深理解数据结构、离散数学、高级语言基本概念，达到培养学生良好程序设计的习惯和运用高级语言编写程序解决实际问题的能力，使学生学会把书本知识用于解决实际问题，起到深化理解和灵活掌握教学内容的目的。课程同时使学生在程序设计方法及上机操作等基本技能和科学作风方面受到比较系统和严格的训练。

### 2 培养目标

#### (1) 基本要求

掌握高级语言程序设计的方法、数据结构和离散数学理论知识，熟悉高级程序的开发环境及高级程序的调试过程，巩固和加深对理论课中知识的理解，提高学生对所学知识的综合运用能力。

#### (2) 具有下列基本技能

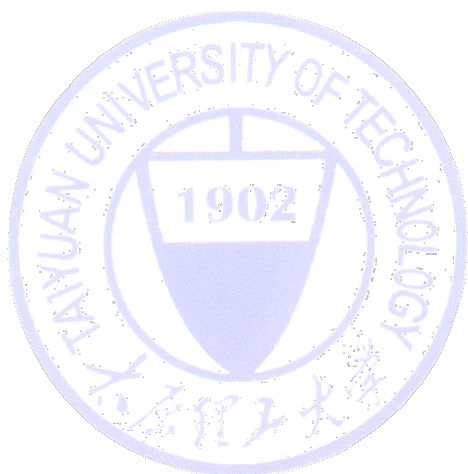
- ①具备查阅参考资料、手册的自学能力，通过独立思考深入钻研问题，学会自己分析、解决问题。
- ②通过对所选题目方案分析比较，确立方案，编制程序与调试程序。
- ③能熟练调试程序，在教师的指导下，完成课题任务。
- ④根据个人的设计调试过程，按课程设计报告的要求撰写设计报告。

### 3 选用教材及主要参考书

#### (1) 教材：《程序设计课程设计》课设指导书

(2) 参考书

- [1] 呼克佑. C 语言程序设计(第 2 版). 电子工业出版社, 2018 年 8 月 1 日
- [2] 董付国. Python 程序设计基础与应用. 机械工业出版社, 2018 年 9 月 1 日
- [3] 严蔚敏. 数据结构(C 语言版). 清华大学出版社, 2022 年 1 月 1 日
- [4] 郝晓燕. 离散数学(第 2 版). 人民邮电出版社, 2021 年 5 月 1 日



## 二 要求

### 1 采用项目小组开发模式

要求学生采用“项目小组”的形式，结合具体的开发项目进行设计。分析设计过程可以采用面向对象或面向过程分析设计。

具体要求如下：

(1) 班级按项目小组进行分组，**每组 4-5 人**。可以任意选择以下两种分组模式：

- ① 按“主程序员”组织软件开发小组，程序员小组的人数应视任务的大小和完成任务的时间机时定。为降低系统开发过程的复杂性，小组内程序员之间的任务界面必须清楚并尽量简化。“主程序员”应该是“超级程序员”。其他成员，包括程序员、后备工程师等，是主程序员的助手。主程序员负责规划、协调和审查小组的全部技术活动。程序员负责软件的分析 and 开发。后备工程师是主程序员的助手，必要时能代替主程序员领导小组的工作并保持工作的连续性。
- ② 按“无我程序设计”建立软件民主开发小组。程序员小组的人数应视任务的大小和完成任务的时间机时定。组织形式强调组内成员人人平等，组内问题均由集体讨论决定。这种组织形式有利于集思广益、互相取长补短，但工作效率比较低。

软件开发各个阶段所需要的技术人员类型、层次和数量是不同的。软件项目的计划与分析阶段只需要少数人，主要是系统分析员、从事软件系统论证和概要设计的软件高级工程师和项目高级管理人员。概要设计时要增加一部分高级程序员，详细设计时，要增加软件工程师和程序员，在编码和测试阶段还要增加程序员、软件测试员。在此过程中软件开发管理人员和各类专门人员逐渐增加，到测试阶段结束时，软件项目开发人员的数量达到顶峰。秘书负责维护和软件配置中的文档、源代码、数据及所依附的各种磁介质；规范并收集软件开发过程中的数据；规范并收集可重用软件，对它们分类并提供检索机制；协助软件开发小组准备文档，对项目中的各种参数，如代码行、成本、工作进度等，进行估算；参与小组的管理、协调和软件配置的评估。

(2) 每个项目小组选出一名项目组长，由项目组长召集项目组成员讨论、选定开发项目，同一项目小组同学采用同一个课程设计题目。项目组长的课程设计成绩有加分。项目开发的每项任务要落实到人且规定该任务的起止日期和时间。

(3) 需求分析，完成需求技术报告，全体成员参加需求讨论，修订并评审需求分析人员的需求分析报告，确定系统的需求分析模型。

(4) 项目组每位人员均需要参加系统总体设计。

(5) 每一位同学均需按照需求分析任务书和概要设计说明的要求，独立该项目 2 个以上模块的详细设计、编

码、测试工作。项目小组应共同进行各模块详细设计的评审工作。

(6) 编程语言、数据库平台可以由项目小组根据项目性质选择，同一个项目的编程语言与数据库平台应一致。项目提交时，应将所有模块集成在一起。不能集成在一起运行的，酌情扣分。

(7) 每一位同学完成自己所负责模块的测试工作，并完成测试报告。

(8) 每个项目小组必须提交一份完整的项目技术文档，文档内容包含《需求分析说明书》、《概要设计说明书》、《详细设计说明书》、《测试报告》与《项目开发日志》等部分，《项目可行性研究报告》可以选作。所有文档必须符合《计算机开发规范》。

(9) 每一位同学应提交一份开发日志，详细记录课程设计过程中出现的问题、解决方案及课程设计心得，务必真实详尽。

## 2 课设报告要求

程序设计课程设计最后以小组为单位提交一份纸质版报告，报告提交时间咨询各自指导教师为准。程序设计课程设计报告格式参考报告模板，要求有统一的报告封面、任务书、日志、目录、正文以及最后的设计心得体会等内容，正文部分排版基本要求为五号 宋体，对于每一个所选定的题目报告需要包含以下内容：

### (1) 需求分析

简要说明程序设计的任务，程序要做什么。明确规定以下内容：

- ① 输入的形式和输入值的范围；
- ② 输出的形式；
- ③ 程序所能达到的功能；
- ④ 测试数据：包括正确的输入及其输出结果和含有错误的输入及其输出结果。

### (2) 概要设计

说明本程序中用到的所有抽象数据类型的定义、主程序的流程以及各程序模块之间的层次（调用）关系。

### (3) 详细设计

实现概要设计中定义的所有数据类型，对每个操作写出伪码算法；对主程序和其他模块也写出伪码算法（伪码算法的详细程度为按照伪码算法可以在计算机键盘直接输入高级程序设计语言程序）；画出函数的调用关系图。

### (4) 测试分析

内容包括：

- ① 测试过程中遇到的问题是如何解决的以及对设计与实现的回顾讨论与分析；此部分详细给出说明
- ② 算法的时空分析和改进设想；
- ③ 经验和体会。

（5）使用说明

说明如何使用程序，列出每一步的操作步骤。

（6）测试结果

列出测试结果，包括输入和输出。测试数据应该完整和严格。

（7）附录

列出程序文件名的清单以及带注释的源程序。



### 三 备选题目

## 1 算法过程可视化系统

### (1) 问题描述

随着多媒体计算机和网络技术的迅速发展，应用多媒体 CAI 技术已经成为广大教育工作者改革教学方法，提高教学质量的首选途径。多媒体演示课件是多媒体教育发展到一定阶段的产物，也是目前多媒体教育改革的一个标志。利用多媒体演示课件，可以代替传统的黑板板书，而且还能把各种与教材有关的多媒体素材融合在演示课件中，在增加知识信息的同时，大大提高了课堂效率。

### (2) 基本要求

在数据结构和离散数学这两门课程中，最难理解的就是其中的各类算法，为了加深用户对算法的理解，多媒体 CAI 加入大量的算法演示。这样有助于学生更加方便得学好课程。以一些经典的算法问题为例，设计算法过程可视化系统，用菜单的形式选择各项功能。

可视化技术不限，建议选择如下方案：

- Java 多线程
- C 语言 + Qt
- 网页技术 + CSS / HTML5
- Python + Tkinter / Qt



每个分组必须从 7 个算法中选择至少选择 4 个进行组内分工，最终评分决定于题目选择的难度和完成情况。最后小组内必须将这四个题目进行整合成为一个应用程序，并统一接口。


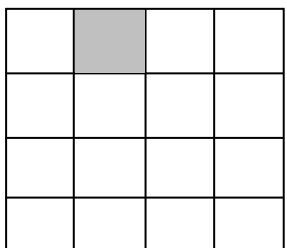
程序要求书写规范，注意代码风格和注释。UI 要对用户足够友好。

### (3) 高级功能

设计图形化界面（GUI）的形式来动态显示算法的执行过程；预留接口可扩充更多教学内容以及算法

	难度	题目	问题描述
1	★★★	汉诺塔问题	<p>现有三根柱子 A、B 和 C，其中 A 柱上从下往上按照大小顺序摞着 <math>n</math> 片圆盘，现需按照以下规则把圆盘按大小顺序重新摆放在 C 柱上：</p> <ul style="list-style-type: none"> <li>● 在三根柱子之间一次只能移动一个圆盘。</li> <li>● 移动的时候始终只能小圆盘压着大圆盘。</li> <li>● 盘子只能在三个柱子上存放</li> </ul> <p>请给出一种可能的移动方案，并将移动过程进行可视化。（注意：不只是结果可视化）</p>



2	★	最小生成树问题	设 $G = (V, E)$ 是一个无向连通图, $G$ 的生成树上各边权值之和称为该生成树的代价。请使用 Prim 或 Kruskal 算法确定 $G$ 的最下代价生成树, 并将最小生成树产生过程进行可视化。(注意: 不只是结果可视化)
3	★★★★	哈夫曼树构造问题	哈夫曼编码是广泛地用于数据文件压缩的十分有效的编码方法。对于一个包含 $n$ 种字符的报文, 每 $i$ 字符在报文中出现的频数为 $f_i$ , 为了提高通信效率, 试确定针对这些字符的最优二元前缀码, 使得该报文的传输长度最短。并给出确定该最优二元前缀码的哈夫曼树的构造过程。(注意: 不只是结果可视化)
4	★★★★★	旅行商 (TSP) 问题	现有 $n$ 个城市, 两两之间均有路径, 但长度可能不同。一个旅行商要将一个城市作为始发城市, 接着旅行剩下的 $n-1$ 个城市, 最优回到始发城市。要求: 除了始发城市, 其他经过且仅经过一次。试确定一个的巡回路线 (哈密尔顿回路) 使其路径总长度最短, 并将该回路确定的过程进行可视化。(注意: 不只是结果可视化)
5	★★★★	图的 $m$ 着色问题	<p>给定无向连通图 <math>G = (V, E)</math>, 求图 <math>G</math> 的最小数 <math>m</math>, 使得用 <math>m</math> 种颜色对 <math>G</math> 中的顶点着色, 可使任意两个相邻顶点着色不同。并将着色过程进行可视化。(注意: 不只是结果可视化)</p> 
6	★★	快速排序/冒泡排序 (二选一)	试对 $n$ 个元素的数组进行快速排序或冒泡排序, 并对排序过程进行可视化。(注意: 不只是结果可视化)
7	★★★★★	棋盘覆盖问题	<p>在一个 <math>2^k \times 2^k</math> (<math>k \geq 0</math>) 个方格组成的棋盘中, 恰有一个方格与其他方格不同, 称该方格为特殊方格。试用以下 4 种不同形状的 L 型骨牌覆盖给定棋盘上除特殊方格以外的所有方格。要求任何 2 个 L 型骨牌不得重叠覆盖。并给出覆盖过程的可视化。(注意: 不只是结果可视化)</p> <div style="display: flex; justify-content: space-around; align-items: flex-end;"> <div style="text-align: center;">  <p><math>k = 2</math> 时的一种棋盘</p> </div> <div style="text-align: center;">  <p>4 种不同形状的 L 型骨牌</p> </div> </div>

## 2 飞机大战游戏单机版

问题描述：设计一款单机版的射击类游戏。

### (1) 基本要求

- 玩家注册
- 玩家登录
- 游戏数据持久化到文件或者数据库中
- 游戏具有成绩排行榜
- 游戏具有开始、暂停、结束功能键

### (2) 核心功能需求分析参考样例

- 参与的角色
  - 英雄机、子弹、大敌机、小敌机、子弹补给、炸弹补给、天空
- 英雄机发射子弹
  - 英雄机、子弹、小敌机、大敌机、子弹补给、炸弹补给都在天空上飞行
  - 子弹可以射击小敌机、大敌机、子弹补给、炸弹补给
  - 小敌机、大敌机、子弹补给、炸弹补给可以和英雄机撞击

- 英雄机发射单倍、双倍子弹，由火力值控制

英雄机发射单倍火力，火力值不减

英雄机发射一次双倍火力，则火力值减 2

- 子弹射击小敌机，子弹与小敌机消失，玩家得分 1

子弹射击大敌机，子弹与大敌机消失，玩家得分 2

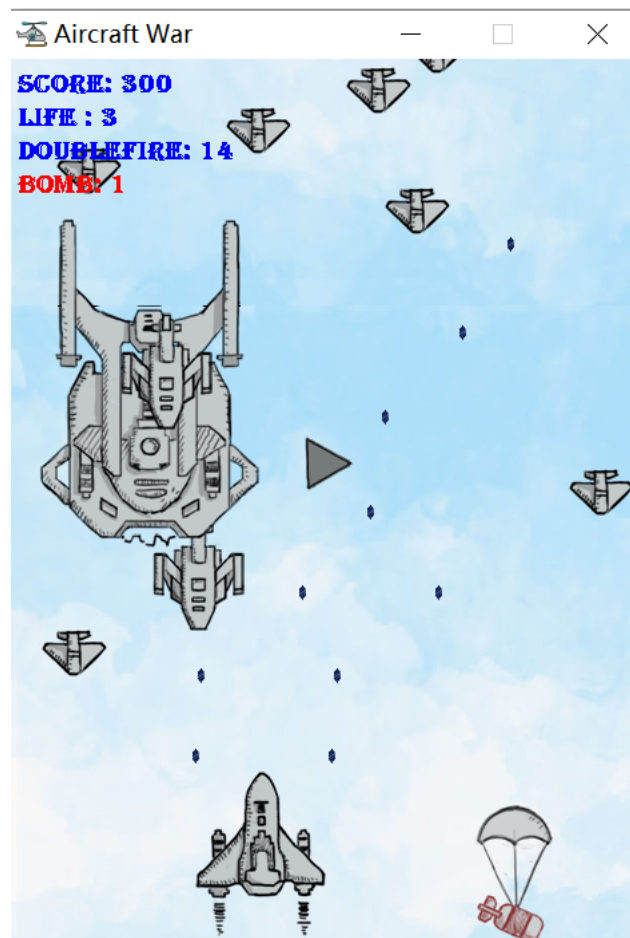
子弹射击子弹补给，子弹与子弹补给消失，英雄机得奖励（20 火力值）

子弹射击炸弹补给，子弹与炸弹补给消失，英雄机得奖励（1 炸弹）

- 小敌机、大敌机、子弹补给、炸弹补给撞击英雄机，命减命 1，火力值清空，小敌机、大敌机、子弹补给、炸弹补给消失

- 英雄机发射炸弹，屏幕内的所有敌机消失

(3) 核心功能参考效果



(4) 实现技术不限，也可以使用 Unity、Unreal 等图形引擎

游戏制作素材下载:

百度网盘链接: [https://pan.baidu.com/s/10FploHBtU2Td8n0MYJqs\\_g](https://pan.baidu.com/s/10FploHBtU2Td8n0MYJqs_g) 提取码: 4216

## 四 课设报告参考样例

【迷宫问题】编制一个求解迷宫通路的程序。

### 1 需求分析

(1) 以二维数组迷宫[M+2][N+2]表示迷宫，其中：迷宫[0][J]和迷宫[M+1][J] ( $0 \leq J \leq N+1$ ) 及迷宫[我][0]和迷宫[我][N+1] (第一  $0 \leq M+1$ ) 为添加的一圈障碍。数组中以元素值为 0 表示通路，1 表示障碍。限定迷宫的大小 M,  $N \leq 10$ 。

(2) 用户以文件的形式输入迷宫的数据：文件中第一行的数据为迷宫的行数 M 和列数 N；从第 2 行至第 M+1 行（每行 N 个数）为迷宫值，同一行中的两个数字之间用空白字符相隔。

(3) 迷宫的入口位置和出口位置可由用户随时设定。

(4) 若设定的迷宫存在通路，则以长方阵形式将迷宫及其通路输出到标准输出文件（即终端）上，其中，字符“#”表示障碍，字符“\*”表示路径上的位置，字符“@”表示“死胡同”，即曾途经然而不能到达出口的位置，余者用空格符印出。若设定的迷宫不存在通路，则报告相应信息。

(5) 本程序只求出一条成功的通路。然而，只需要对迷宫求解的函数作小量修改，便可求得全部路径。

(6) 测试数据见原题，当入口位置为 (1, 1)，出口位置为 (9, 8) 时，输出数据应为：

表 3-1 输出数据表

*	*	#	@	@	@	#	
	*	#	@	@	@	#	
*	*	@	@	#	#		#
*	#	#	#			#	@
*	*	*	#	*	*	*	@
	#	*	*	*	#	*	#
	#	#	#	#		*	#
#	#				#	*	#
#	#					*	*

(7) 程序执行的命令为：

1) 创建迷宫； 2) 求解迷宫； 3) 输出迷宫的解。

### 2 概要设计

(1) 设定栈的抽象数据类型定义为：

```
ADT stack{
    数据对象：D = {ai|ai∈charset, i = 1, 2, …, n, n≥0}
```

数据关系:  $R1 = \{ \langle ai-1, ai \rangle | ai-1, ai \in D, i = 2, \dots, n \}$

基本操作:

InitStack(&S)

操作结果: 构造一个空栈 S。

DestroyStack(&S)

初始条件: 栈 S 已存在。

操作结果: 销毁栈 S。

ClearStack(&S)

初始条件: 栈 S 已存在。

操作结果: 将 S 清为空栈。

StackLength(&S)

初始条件: 栈 S 已存在。

操作结果: 返回栈 S 的长度。

StackEmpty(&S)

初始条件: 栈 S 已存在。

操作结果: 若 S 为空栈, 则返回 TRUE, 否则返回 FALSE。

GetTop(S, &e)

初始条件: 栈 S 已存在。

操作结果: 若栈 S 不空, 则以 e 返回栈顶元素。

Push(&S, e)

初始条件: 栈 S 已存在。

操作结果: 在栈 S 的栈顶插入新的栈顶元素 e。

Pop(&S, &e)

初始条件: 栈 S 已存在。

操作结果: 删除 S 的栈顶元素, 并以 e 返回其值。

StackTraverse(S, visit())

初始条件: 栈 S 已存在。

操作结果: 从栈底到栈顶依次对 S 中的每个元素调用函数 visit()。

}ADT stack

(2) 设定迷宫的抽象数据类型为:

ADT maze{

数据对象:  $D = \{ ai, j | ai, j \in \{ ' ', '#', '@', '*' \}, 0 \leq i \leq m+1, 0 \leq j \leq n+1, m, n \leq 10 \}$

数据关系:  $R = \{ ROW, COL \}$

$ROW = \{ \langle ai-1, j, ai, j \rangle | ai-1, j, ai, j \in D, i = 1, \dots, m+1, j = 0, \dots, n+1 \}$

$COL = \{ \langle ai, j-1, ai, j \rangle | ai, j-1, ai, j \in D, i = 0, \dots, m+1, j = 1, \dots, n+1 \}$

基本操作:

InitMaze(&M, a, row, col)

初始条件: 二维数组  $a[row+2][col+2]$  已存在, 其中自第 1 行至第 row+1 行、每行中自第 1 列至第 col+1 列的元素已有值, 并且以值 0 表示通路, 以值 1 表示障碍。

操作结果: 构成迷宫的字符型数组, 以空白字符表示通路, 以字符 '#' 表示障碍, 并在迷宫四周加上一圈障碍。

MazePath(&M)

初始条件: 迷宫 M 已被赋值。

操作结果: 若迷宫 M 中存在一条通路, 则按如下规定改变迷宫 M 的状态: 以字符 "\*" 表示路径上的位置, 字符 "@" 表示 "死胡同"; 否则迷宫的状态不变。

PrintMaze(M)

初始条件: 迷宫 M 已存在。

操作结果: 以字符形式输出迷宫。

}ADT maze;

(3) 本程序包含三个模块

### 1) 主程序模块

```
void main( ){
    初始化
    do{
        接受命令;
        处理命令;
    }while(命令! = “退出”);
}
```

### 2) 栈模块----实现栈抽象数据类型

### 3) 迷宫模块----实现迷宫抽象数据类型

各模块之间的调用关系如下：

主程序模块



迷宫模块



栈模块



### (4) 求解迷宫中一条通路的伪码算法：

设定当前位置的初值为入口位置；

```
do{
    若当前位置可通，则{
        将当前位置插入栈顶；           //纳入路径
        若该位置是出口位置，则结束；    //求得路径存放在栈中
        否则切换当前位置的东邻方块为新的当前位置；
    }
    否则{
        若栈不空且栈位置尚有其他方向未被探索，则
            设定新的当前位置为沿顺时针方向旋转找到的栈顶位置的下一相邻块；
        若栈不空但栈顶位置的四周均不可通，则{
            删去栈顶位置；           //后退一步，从路径中删去该通道块，
            若栈不空，则重新测试新的栈顶位置，
            直到找到一个可通的相邻块或出栈至栈空；
        }
    }
}while(栈不空)，
{栈空说明没有路径存在}
```

## 3 详细设计

### (1) 坐标位置类型

```
typedef struct{
    int r, c;      //迷宫中行、列的范围
}postype;
```

### (2) 迷宫类型

```
typedef struct{
    int    m, n;
    char   arr[RANGE][RANGE]; //各位置取值 ‘ ’ , ‘#’ , ‘@’ 或 ‘*’
}mazetype;
```

```
void initmaze(mazetype &maze, int a[][], int row, int col)
//按照用户输入的 row 行和 col 列的二维数组（元素值为 0 或 1）
//设置迷宫的初值，包括加上边缘一圈的值
```

```
bool mazepath(mazetype &maze, postype start, postype end)
//求解迷宫 maze 中，从入口 start 到出口 end 的一条路径
//若存在，则返回 TRUE；否则返回 FALSE
```

```
void printmaze(mazetype maze)
//将迷宫以字符型方阵的形式输出到标准输出文件上
```

### (3) 栈类型

```
typedef struct{
    int          step;      //当前位置在路径上的“序号”
    postype      seat;      //当前的坐标位置
    directivetype di;       //往下一坐标位置的方向
}Elemtype;                //栈的元素类型
```

```
typedef struct nodetype{
    elemtype    data;
    nodetype    *next;
}nodetype, *linktype; //结点类型，指针类型
```

```
typedef struct{
    linktype top;
    int      size;
}stack;     //栈类型
```

栈的基本操作设置如下：

```
void initstack(stack &S)
//初始化，设 S 为空栈（S.top = NULL）
```

```
void destroystack(stack &S)
//销毁栈 S，并释放所占空间
```

```
void clearstack(stack &S)
//将 S 清为空栈
```

```
int stacklength(stack S)
//返回栈 S 的长度 S.size
```





```

status stackempty(stack S)
//若 S 为空栈 (S.top == NULL), 则返回 TRUE; 否则返回 FALSE

status gettop(stack s, Elemtype e)
//若栈 S 不空, 则以 e 带回栈顶元素并返回 TRUE, 否则返回 FALSE;

status push(stack &S, Elemtype e)
//若分配空间成功, 则在 S 的栈顶插入新的栈顶元素 e, 并返回 TRUE,
//否则栈不变, 并返回 FALSE

status pop(stack &S, Elemtype &e)
//若栈不空, 则删除 S 的栈顶元素并以 e 带回其值, 且返回 TRUE
//否则返回 FALSE

void stacktraverse(stack s, status(*visit)(Elemtype e))
//从栈底到栈顶依次对 S 中的每个结点调用函数 visit

```

其中部分操作的算法:

```

status push(stack &S, Elemtype e){
    //若分配空间成功, 则在 S 的栈顶插入新的栈顶元素 e, 并返回 TRUE;
    //否则栈不变, 并返回 FALSE
    if (Makenode(p, e)){
        p^.next = s.top;
        s.top = p;
        s.size++;
        return TRUE;
    }
    else return FALSE;
}

status pop(stack &S, Elemtype &e){
    //若栈不空, 则删除 S 的栈顶元素并以 e 带回其值, 且返回 TRUE,
    //否则返回 FALSE, 且 e 无意义
    if(stackempty(s)
        return FALSE);
    else{
        p = S.top;
        S.top->next;
        e = p->date;
        S.size--;
        return TRUE;
    }
}

```

#### (4) 求迷宫路径的伪码算法

```

status mazepath(mazetype maze, postype start, postype end){
    //若迷宫中存在从入口 start 到出口 end 的通道, 则求得一条存入在栈中
    // (从栈底到栈顶为从入口到出口的路径), 并返回 TRUE; 否则返回 FALSE
    initstack(S);
    curpos = start; //设定“当前位置”为“入口位置”
    curstep = 1;
    found = FALSE;           //探索第一步
    do{
        if(pass(maze, curpos)){
            //当前位置可以通过, 即是未曾走到过的通道块

```



```

    footprint(maze, curpos); //留下足迹
    e = (curstep, curpos, 1);
    push(S, e); //加入路径
    if(same(curpos, end))
        found = TRUE; //到达终点（出口）
    else{
        curpos = nextpos(curpos, 1); //下一位置是当前位置的东邻
        curstep++; //探索下一步
    }//else
} //if
else //当前位置不能通过
    if(!stackempty(S)){
        pop(S, e);
        while(e.di == 4 && !stackempty(S)){
            markprint(maze, e, seat);
            pop(S, e);
            curstep--; //留下不能通过的标记，并退回一步
        }//while
        if(e.di < 4){
            e.di++;
            push(S, e) //换下一个方向探索
            curpos = nextpos(e.seat, e.di); //设定当前位置是该新方向上的相邻块
        }//if
    } //if
} while(!stackempty(S) && !found);
return found;
} //mazepath

```

#### (5) 主函数和其他函数的伪码算法

```

void main() { //主程序
    initialization(); //初始化
    do{
        readcommand(cmd); //读入一个操作命令符
        interpret(cmd); //解释执行操作命令符
    } while(cmd != 'q' && cmd != 'Q' );
} //main

void initialization() { //系统初始化
    clrscr(); //清屏
    在屏幕上方显示操作命令清单:
    creatmaze—c    mazepath—m    printmaze—p    Quit—q;
    在屏幕下方显示操作命令提示框:
} //initialization

void readcommand(char &cmd) { //读入操作命令符
    显示键入操作命令符的提示信息;
    do{
        cmd = getch();
    } while(cmd != 'c' , 'C' , 'm' , 'M' , 'p' , 'P' , 'q' , 'Q' );
} //readcommand

```

```
void interpret(char cmd){//解释执行操作命令
    switch(cmd){
        case 'c' , 'C' :
            提示用户输入“迷宫数据的文件名 filename”;
            从文件读入数据分别存储在 rnum, cnum 和二维数组 a2 中;
            initmaze(ma, a2, rnum, cnum); // 创建迷宫
            输出迷宫建立完毕的信息
            break;
        case 'm' , 'M' :
            提示用户输入迷宫的入口 from 和出口 term 的坐标位置;
            if(mazepath(ma, from, term)) //存在路径
                提示用户察看迷宫;
            else
                输出该迷宫没有从给定的入口到出口的路径的信息;
            break;
        case 'p' , 'P' :
            printmaze(ma): //将标记路径信息的迷宫输出到终端
    }//switch
} //interpret
```

(6) 函数的调用关系图反映了演示程序的层次结构:



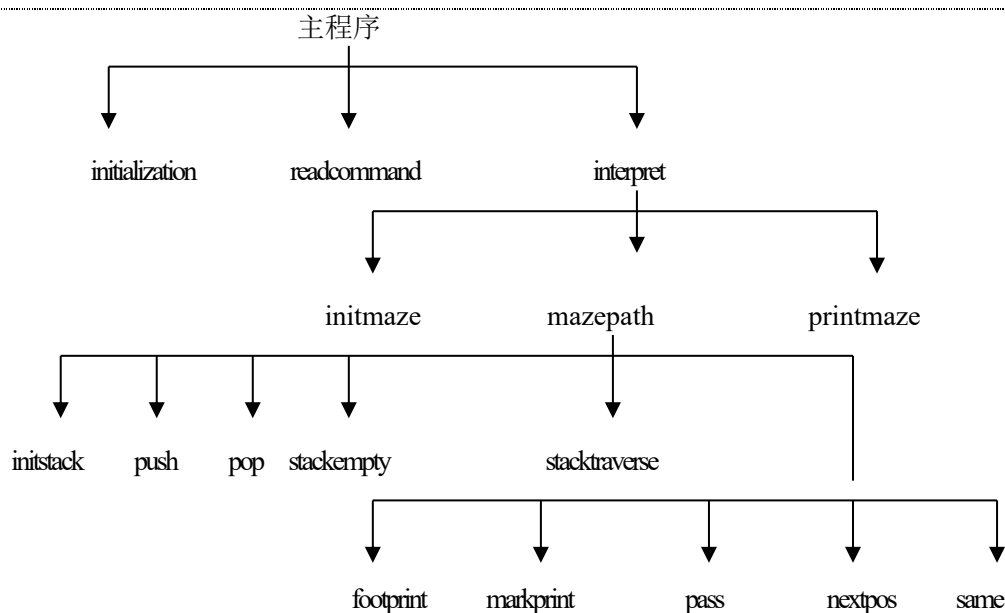


图3-1 函数的调用关系图

#### 4 调试分析

(1) 本次作业比较简单，只有一个核心算法，即求迷宫的路径，所以总的调试比较顺利，只在调试 `mazepath` 算法时，遇到两个问题：其一是，起初输出的迷宫中没有加上“@”的记号，后发现是因为在 `markprint` 函数中的迷宫参数丢失“变参”的原因；其二是，由于回退时没有将 `curpos` 随之减一，致使栈中路径上的序号有错。

(2) 栈的元素中的 `step` 域没有太多用处，可以省略。

(3) `Stacktraverse` 在调试过程中很有用，它可以插入在 `mazepath` 算法中多处，以察看解迷宫过程中走的路径是否正确，但对最后的执行版本没有用。

(4) 本题中三个主要算法：`initmaze`, `mazepath` 和 `printmaze` 的时间复杂度均为  $O(m*n)$ ，本题的空间复杂度亦为  $O(m*n)$ （栈所占最大空间）

(5) 经验体会：借助 `DEBUG` 调试器和数据观察窗口，可以加快找到程序中疵点。

#### 5 用户手册

(1) 本程序的运行环境为 `DOS` 操作系统，执行文件为：`testmaze.exe`

(2) 进入演示程序后，即显示文本方式的用户界面：

(3) 进入“产生迷宫 (`createmaze`)”的命令后，即提示键入迷宫数据的文件名，结束符为“回车符”，该命令执行之后输出“迷宫已建成”。

(4) 进入“求迷宫路径 (`mazepath`)”的命令后，即提示键入入口位置（行号和列号，中间用空格分开，结束

符为“回车符”)和出口位置(行号和列号,中间用空格分开,结束符为“回车符”),该命令执行之后输出相应信息。  
若迷宫中存在路径,则执行此命令后,迷宫状态已改变,若要重复执行此命令,无论是否改变出口和入口的位置,均需重新输入迷宫数据。

(5) 输入“显示迷宫”的命令后,随即输出当前的迷宫,即迷宫的初始状态或求出路径之后的状态。

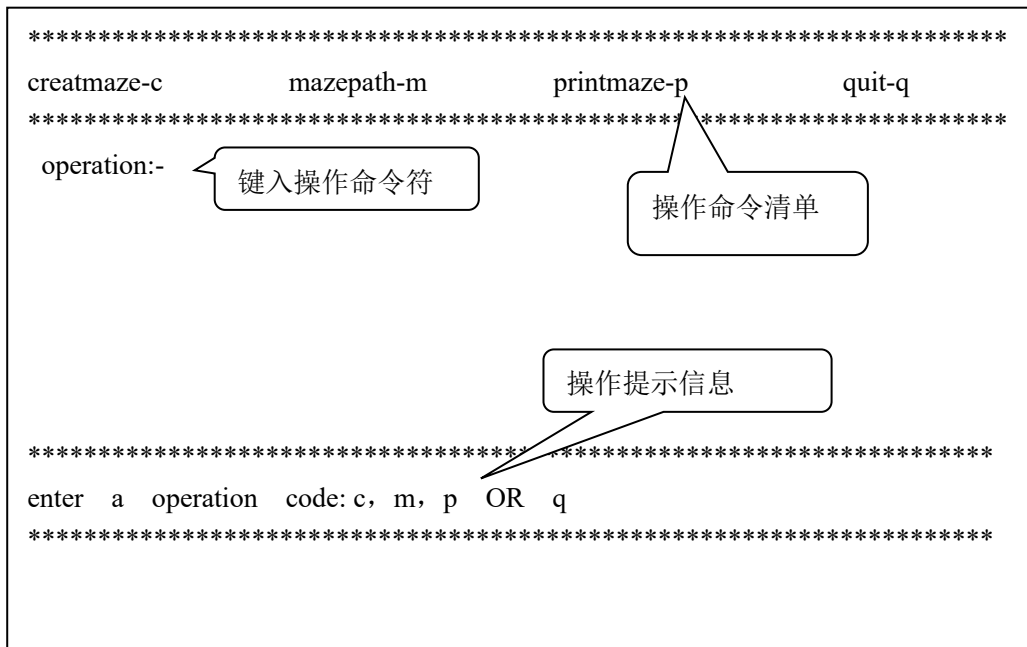


图 3-2 操作界面

## 6 测试结果

三组测试数据和输出结果分别如下:

(1) 输入文件名为: m1.dat, 其中迷宫数据为:

```
3 2
0 0
0 0
0 0
```

入口位置: 1 1

出口位置: 3 2

求解路径后输出的迷宫:

表 3-2 输出数据

*	*
	*
	*

(2) 输入文件名: m2.dat, 其中迷宫数据为:

```
3 4
0 0 0 0
0 0 1 1
0 0 0 0
```

入口位置: 1 1

出口位置: 3 4

求解路径后输出的迷宫:

表 3-3 输出数据

*	*	@	@
	*	#	#
	*	*	*

(3) 输入文件名: m3.dat, 其中迷宫数据同题目中的测试数据。

入口位置: 1 1

出口位置: 9 8

求解路径后输出的迷宫正确, 并和需求分析中所列相同。

(4) 输入文件名: m4.dat, 其中迷宫数据为:

```
4 9
0 0 0 0 0 0 1 0 0
0 1 0 0 0 1 0 0 0
0 0 1 1 1 0 0 1 1
0 0 1 1 1 0 1 0 0
```

入口位置: 1 1

出口位置: 4 9

输出信息为: 此迷宫从入口到出口没有路径。

## 7 附录

源程序文件名清单:

```
base.h          //公用的常量和类型
stkpas.h        //栈类型
maze.h          //迷宫类型
testmaze.c      //主程序
```

## 五 考核标准

课程设计成绩依据学生的动手能力、独立分析解决问题的能力、创新能力、课程设计报告、现场验收设计回答问题情况以及学习态度综合评定。

### 1 成绩评定标准

#### (1) 优秀

设计态度端正，认真遵守各项规章制度，能正确理解设计的目的要求，能独立、正确地完成设计任务，会分析和处理设计中遇到的问题，能掌握设计中所用到的各项技能，认真撰写课程设计报告，有一定创新精神和能力。

#### (2) 良好

设计态度端正，认真遵守各项规章制度，能理解设计的目的和要求，能认真而正确地完成设计任务，能分析和处理设计中遇到的一些问题。能掌握设计中所用到的设计技能，对难点较大的操作完成有困难。撰写设计报告比较认真。

#### (3) 中等

设计态度较端正，能遵守各项规章制度，能基本理解实验目的要求，能认真努力进行课程设计，但技能一般。能分析和处理设计中一些较容易的问题，掌握设计技能的大部分。能完成设计报告。能认真遵守各项规章制度，学习努力。

#### (4) 及格

设计态度基本端正，基本能遵守各项规章制度，能机械地了解设计内容，能基本按步骤完成设计。遇到问题常常缺乏解决的办法，在别人启发下能作些简单处理，但效果不理想，能基本完成设计报告。

#### (5) 不及格

设计态度不端正，不能遵守各项规章制度，出勤不足 60%，不能掌握设计技能，遇到问题时说不明原因。在教师指导下也较难达到设计要求，设计报告不合要求。

### 2 项目组成成绩考核

(1) 项目设计规范、合理。

(2) 软件项目人机界面友好、功能完善、可以正常运行。

(3) 项目技术文档规范、翔实。

(4) 项目分工合理、能够按计划完成项目开发。

### 3 项目组成员个人成绩考核

- (1) 积极参加项目各个阶段的工作。
- (2) 所负责的模块设计规范、功能完善、文档齐全。
- (3) 考勤合格。
- (4) 按时独立完成工作日志。
- (5) 项目组长额外有加分。

每一位学生的成绩由两部分组成，即：项目组成绩+个人成绩。其中项目组成绩占 50%，个人成绩占 50%。

