



計算機程序設計基礎（C）

實驗報告

學生姓名 許采婷

學 號 8208230832

專業班級 計通 2308 班

指導教師 楊迎澤

學 院 計算機學院

实验六 综合实验

1.实验目的要求

- (1) 掌握选择、循环、数组、字符串、函数和链表的综合应用。
- (2) 针对实际工程应用问题，能利用所学知识点设计算法进行编程实现，锻炼动手实践和解决问题能力，提升创新思维。
- (3) 培养良好的逻辑思维能力和严谨的分析问题能力。

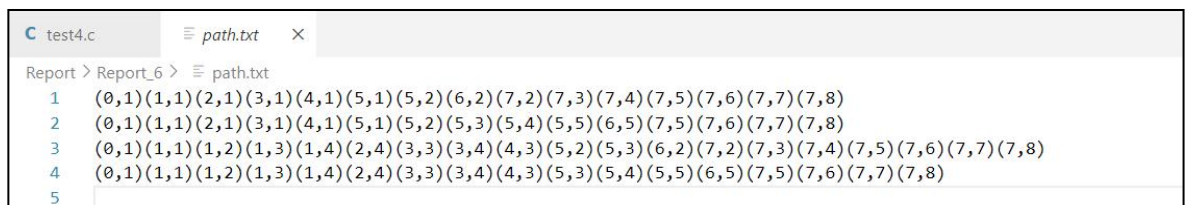
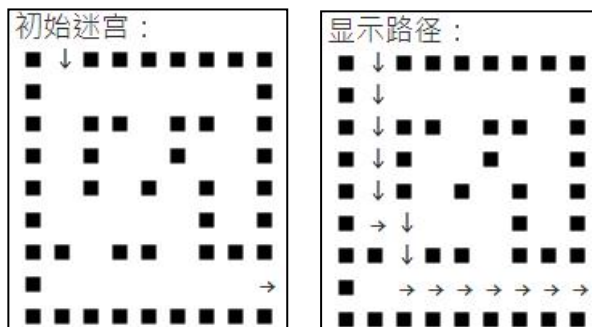
2.实验内容

2.1 老鼠迷宫问题

说明：(1) 利用 2 维数组构建迷宫

- (2) 实现老鼠从入口到出口的所有路径，并将路径保存到文件
- 选择某一种路径，动态实现老鼠在迷宫中的运行轨迹

(1) 实验运行结果：



(2) 源代码：

```
// 老鼠迷宫问题

#include <stdio.h>
#include <stdlib.h>
#define M 9

// 迷宫初始化[全局变量]
// 0:路 1:位置 2:牆 3:無 4:左
// 5:下 6:右 7:無 8:上 9:無
int maze[M][M] =
{
    {2,5,2,2,2,2,2,2,2},
    {2,0,0,0,0,0,0,0,2},
    {2,0,2,2,0,2,2,0,2},
    {2,0,2,0,0,2,0,0,2},
    {2,0,2,0,0,2,0,0,2},
    {2,0,2,0,0,2,0,0,2},
    {2,0,2,0,0,2,0,0,2},
    {2,0,2,0,0,2,0,0,2},
    {2,0,2,0,0,2,0,0,2}
};
```

```

        {2,0,2,0,2,0,2,0,2},
        {2,0,0,0,0,0,2,0,2},
        {2,2,0,2,2,0,2,2,2},
        {2,0,0,0,0,0,0,0,6},
        {2,2,2,2,2,2,2,2,2}
};

int path[M*M][2] = {0};          // 存儲經過的路徑座標
int start1 = 1, start2 = 1;      // 假定[1][1]是入口
int end1 = 7, end2 = 7;         // 假定[7][7]是出口
int pathNum = 0;                // 走過的位置編號
int flag = 1;                   // 第幾個路徑

void rewrite(int list[M*M][2], int n){ // 第一次寫入

    FILE *fp = fopen("path.txt", "w");

    for(int i=0; i<n; i++){

        fprintf(fp, "(%d,%d)", list[i][0], list[i][1]);
    }

    fprintf(fp, "\n");
    fclose(fp);
}

void write(int list[M*M][2], int n){    // 寫入

    FILE *fp = fopen("path.txt", "a+");

    for(int i=0; i<n; i++){

        fprintf(fp, "(%d,%d)", list[i][0], list[i][1]);
    }

    fprintf(fp, "\n");
    fclose(fp);
}

void visit(int i, int j){

    maze[i][j] = 1;

    if(i==end1 && j==end2 && flag==1){ // 判斷是否到達出口位置，到達直接輸出[第一個]

        flag++; // 只要輸出一個路徑

        printf("\n 顯示路徑：\n");

        for(int m=0; m<M; m++){

            maze[i][j] = 6;

```

```

        for(int n=0; n<M; n++){

            if(maze[m][n] == 2)
                printf("■ ");
            else if(maze[m][n] == 4){

                printf("←");

                path[pathNum][0] = m;
                path[pathNum][1] = n;
                pathNum++;
            }
            else if(maze[m][n] == 5){

                printf("↓");

                path[pathNum][0] = m;
                path[pathNum][1] = n;
                pathNum++;
            }
            else if(maze[m][n] == 6){

                printf("→");

                path[pathNum][0] = m;
                path[pathNum][1] = n;
                pathNum++;
            }
            else if(maze[m][n] == 8){

                printf("↑");

                path[pathNum][0] = m;
                path[pathNum][1] = n;
                pathNum++;
            }
            else
                printf(" ");
        }

        printf("\n");
    }

    rewrite(path, pathNum); // 每次的第一次寫入要先清空
    pathNum = 0;
}
else if(i==end1 && j==end2 && flag!=1){ // 判断是否到达(5,5)出口
位置，到达直接输出

    for(int m=0; m<M; m++){

        maze[i][j] = 6;

        for(int n=0; n<M; n++){

```

```

        if(maze[m][n] == 4){

            path[pathNum][0] = m;
            path[pathNum][1] = n;
            pathNum++;
        }
        else if(maze[m][n] == 5){

            path[pathNum][0] = m;
            path[pathNum][1] = n;
            pathNum++;
        }
        else if(maze[m][n] == 6){

            path[pathNum][0] = m;
            path[pathNum][1] = n;
            pathNum++;
        }
        else if(maze[m][n] == 8){

            path[pathNum][0] = m;
            path[pathNum][1] = n;
            pathNum++;
        }
    }
}

write(path, pathNum);
pathNum = 0;
}

```

// 分析老鼠可以在迷宫移动的方向，并递归求下一步。

// 8 上 5 下 4 左 6 右

```
if(maze[i-1][j] == 0){ // 向上
```

```

    maze[i][j] = 8;
    visit(i-1, j);
}

```

```
if(maze[i+1][j] == 0){ // 向下
```

```

    maze[i][j] = 5;
    visit(i+1, j);
}

```

```
if(maze[i][j-1] == 0){ // 向左
```

```

    maze[i][j] = 4;
    visit(i, j-1);
}

```

```
if(maze[i][j+1] == 0){ // 向右
```

```

    maze[i][j] = 6;
    visit(i, j+1);
}

```

```

    }

    //若代码运行到这一步，则证明前面走的路径并不能到达出口，则返回，把走过的
    位置重新写作 0
    maze[i][j] = 0;
}

int main(){

    int i, j;

    printf("初始迷宫：\n");

    for(i=0; i<M; i++){    // 打印初始化迷宫

        for(j=0 ;j<M; j++){

            if(maze[i][j] == 2)
                printf("■ ");
            else if(maze[i][j] == 5)
                printf("↓");
            else if(maze[i][j] == 6)
                printf("→");
            else
                printf(" ");

        }

        printf("\n");
    }

    visit(start1,start2);    // 调用 visit 函数，把输出内容放在 visit 函数
    中，让所有路径进行遍历

    return 0;
}

/*
-----
    参考網站: https://www.cnblogs.com/tanssss/p/12517473.html
-----
*/

```