

§ 补充资料.C 方式如何将输出放入文件中

例 1: 从键盘读取 1 个 2*3 的矩阵, 转置后输出到屏幕上

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

int main()
{
    int a[2][3], b[3][2];
    int i, j;

    /* 输入并转置 */
    for (i = 0; i<2; i++) {
        for (j = 0; j<3; j++) {
            scanf("%d", &a[i][j]);
            b[j][i] = a[i][j];
        }
    }

    /* 输出 */
    for (j = 0; j<3; j++) {
        for (i = 0; i<2; i++)
            printf("%6d", b[j][i]);
        printf("\n");
    }

    return 0;
}
```

假设键盘输入 1 2 3
4 5 6

则输出为: 1 4
2 5
3 6

例 2: 从文件 m.txt 中读取 1 个 2*3 的矩阵, 转置后输出到屏幕上

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

int main()
{
    int a[2][3], b[3][2];
    int i, j;
    FILE *fin; //fin 为变量名
    //以读方式打开文件
    fin = fopen("m.txt", "r");
    if (fin == NULL) {
        printf("打开文件失败\n");
        return -1;
    }

    /* 输入并转置(观察 scanf 和 fscanf 的差别) */
    for (i = 0; i<2; i++) {
        for (j = 0; j<3; j++) {
            fscanf(fin, "%d", &a[i][j]);
            b[j][i] = a[i][j];
        }
    }

    /* 输出 */
    for (j = 0; j<3; j++) {
        for (i = 0; i<2; i++)
            printf("%6d", b[j][i]);
        printf("\n");
    }

    fclose(fin); //关闭输入文件
    return 0;
}
```

假设 m.txt 中是 1 2 3
4 5 6

则输出为: 1 4
2 5
3 6

从文件中读取数据的方法见补充资料

例 3: 从键盘读取 1 个 2*3 的矩阵, 转置后输出到文件 r.txt 中

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

int main()
{
    int a[2][3], b[3][2];
    int i, j;
    FILE *fout; //fout 为变量名

    //以写方式打开文件
    fout = fopen("r.txt", "w");
    if (fout == NULL) {
        printf("打开文件失败\n");
        return -1;
    }

    /* 输入并转置 */
    for (i = 0; i<2; i++) {
        for (j = 0; j<3; j++) {
            scanf("%d", &a[i][j]);
            b[j][i] = a[i][j];
        }
    }

    /* 输出(观察 fprintf 和 printf 的差别) */
    for (j = 0; j<3; j++) {
        for (i = 0; i<2; i++)
            fprintf(fout, "%6d", b[j][i]);
        fprintf(fout, "\n");
    }

    fclose(fout); //关闭输出文件
    return 0;
}
```

假设键盘输入 1 2 3 (空格) 4 5 6

则屏幕上无任何输出, 打开同目录下的 r.txt, 内容为:

```
1 4
2 5
3 6
```

例 4: 从文件 m.txt 中键盘读取 1 个 2*3 的矩阵, 转置后输出到另一个文件 r.txt 中

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

int main()
{
    int a[2][3], b[3][2];
    int i, j;
    FILE *fin; //fin 为变量名
    FILE *fout; //fout 为变量名
    //以读方式打开文件
    fin = fopen("m.txt", "r");
    if (fin == NULL) {
        printf("打开文件失败\n");
        return -1;
    }

    //以写方式打开文件
    fout = fopen("r.txt", "w");
    if (fout == NULL) {
        printf("打开文件失败\n");
        fclose(fin); //注意此句!!!
        return -1;
    }

    /* 输入并转置(观察 scanf 和 fscanf 的差别) */
    for (i = 0; i<2; i++) {
        for (j = 0; j<3; j++) {
            fscanf(fin, "%d", &a[i][j]);
            b[j][i] = a[i][j];
        }
    }

    /* 输出(观察 fprintf 和 printf 的差别) */
    for (j = 0; j<3; j++) {
        for (i = 0; i<2; i++)
            fprintf(fout, "%6d", b[j][i]);
        fprintf(fout, "\n");
    }

    fclose(fin); //关闭输入文件
    fclose(fout); //关闭输出文件
    return 0;
}
```

运行后不需要键盘输入, 屏幕上无任何输出, 从 m.txt 中读数据, 打开同目录下的 r.txt 中, 内容为:

```
1 4
2 5
3 6
```

向文件中写入多个数据的方法

- 1、用 `fprintf` 可以向文件中写入格式数据
- 2、生成的文件与源程序在相同目录下【如果反复运行，则后次输出覆盖前次】
- 3、例 4 中红色字体为输出文件，蓝色字体为输入文件，请比较不同，主要有两点
【打开方式：`fopen` 的第 2 个参数不同】
【使用方法：读方式打开的文件用 `fscanf` 读，写方式打开的文件用 `fprintf` 写】

操作说明：

- 1、如果在 VS 的集成环境下运行，则生成的数据文件和源程序在同一目录下
- 2、如果编译成 `exe` 文件后，直接双击 `exe` 文件名运行，则生成的数据文件和 `exe` 在同一个目录下
- 3、如果编译成 `exe` 文件后，在 `cmd` 下输入 `exe` 文件名/鼠标拖曳 `exe` 文件名运行，则生成的数据文件是在 `cmd` 窗口的“起始位置”目录下（不一定是 `exe` 所在目录）
- 4、具体请参考上学期的“输入输出重定向及管道运算符”文档，特别是“错误及不建议用法”