**数据结构**

**课程设计报告**

设计题目：数组应用

专 业 网络工程 \_

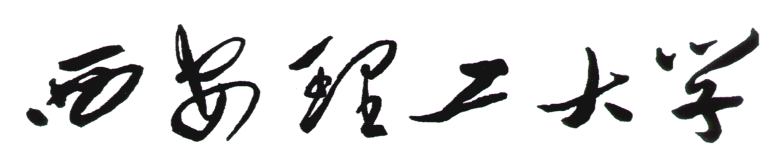
班 级 2016级1班 \_

学 生 卢 家 玺 \_

学 号 3150931005 \_

指导教师 宋昕 / 朱磊 \_

起止时间 2016年7月4日 -7月8日 \_



2016年 春 季 学期

**目录**

1. 课题简介.............................................................................3
2. 设计方案.............................................................................3

2.1总体思路.....................................................................3

2.2 课程设计内容及流程图.................................................4

3.源代码.................................................................................5

4.测试.....................................................................................8

4.1 调试过程中遇到的问题及解决方案.................................8

4.2 调试结果与分析...........................................................9

5.总结体会............................................................................10

6.参考文献............................................................................10

1. 具体任务

读入数组下标，求出数组A靠边元素之和；求从A[0][0]开始的互不相临的各元素之和；当m=n时，分别求出两条对角线上的元素之和，否则打印出m!=n的信息。

要求：1）界面友好，函数功能划分好

2）总体设计应画一流程图

3）程序要加必要的注释

4）要提供程序测试方案

5）要经得起测试，宁可功能少，也要能运行

1. 软件环境

Windows10环境

CodeBlocks

WPS文字

1. 算法设计思想

分步实施：

1. 初步完成总体设计，将功能分成5个部分，然后让用户在array函数里确定这个动态的二维数组

Int \*\*Create2DArr(int m,int n) //为指针创建空间

void Array() //输入数组函数，输出数组函数，并根据用户选择计算相应的功能

int sumside(q,m,n) //计算靠边元素和的函数

int diagcater(q,m,n) //计算对角线元素和的函数

int sumupthedata (q,m,n) //计算不相邻元素和的函数

int menu\_select() //菜单函数

2.输出函数：按照用户输入数组的行和列为界，用两个循环变量挨个输出数据。

3.输入函数：用户自己按照先行后列，自左至右的顺序依次输入自己的数组。

按顺序输入，只需以用户输入的行和列为界，用两个循环变量i和j控制输入数据，然后调用输出函数显示用户输入信息。

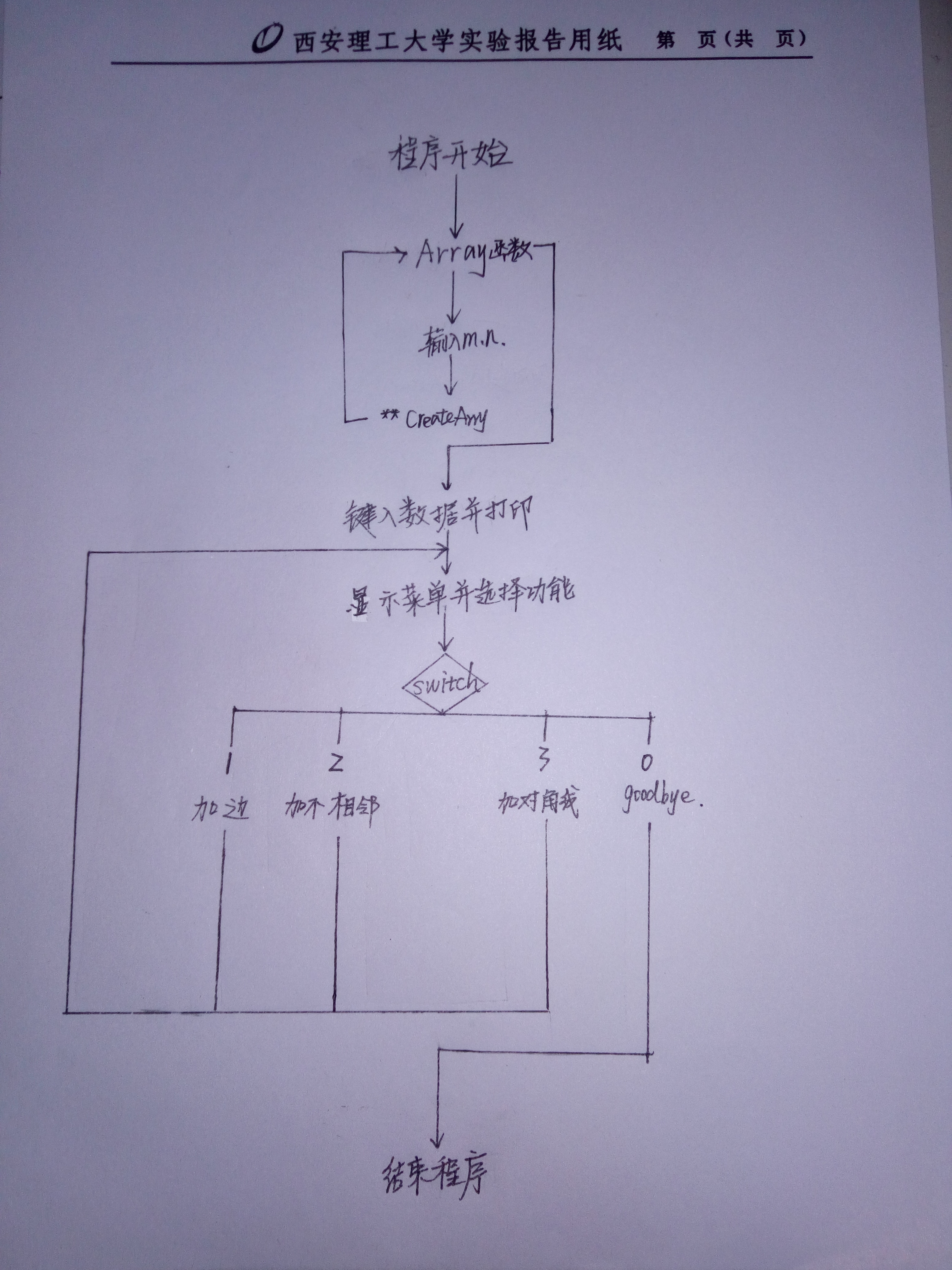
4.加边函数：需要利用两个循环变量，同样以行和列为界限，进行选择。若当前元素的行或列等于0或等于行减1或列减1，则说明该元素在第一行或第一列，或最后一行，或最后一列，即靠边元素。定义变量s来记录靠边元素的和，每选择出一个元素则加上刚才的和，最后通过参数传递给主函数。

5.对角线函数：需要给用户选择，计算正对角线或副对角线，正对角线需要判断循环变量是否相等再求和，副对角线则可以用列递减的方式进行求和。

6.不相邻函数：通过挨个遍历，循环变量每次加1的思想，不相邻元素中间间隔一个元素，则遍历时行变量需要每次加2，但列变量遍历时还是需要递增1，因为不是隔列遍历。

当列数为奇数和偶数时也不同，奇数时，偶数行从第2个元素开始加，奇数行从第1个元素开始；偶数列时，每行都从第一个元素开始求和，需要分类讨论。

1. 主函数：直接引用Array函数，在函数内部进行各项的功能
2. 流程图



1. 源代码

#include <stdio.h>

#include <stdlib.h>

int main()

{

printf("please create an array first!\n");

Array();

return 0;

}

int \*\*Create2DArr(int m,int n)

//m is line,n is column 给一个动态二维数组的指针

{

int \*\*a,i;

a = malloc(m\*sizeof(int \*));

for(i = 0;i < m;i++)

a[i] = malloc(n\*sizeof(int \*));

return a;

}

void Array()

//输入一个二维数组，打印一个二维数组并且将其结果显示

{

int choice = 0;

int m = 0,n = 0;

int \*\*q;

int i,j;

printf("please input the number of the hang.");

scanf("%d",&m);

printf("please input the number of the lie.");

scanf("%d",&n);

q = Create2DArr(m,n);

printf("please input your data\n");

for(i = 0;i < m;i++)

{

for(j = 0;j < n;j++)

{

scanf("%d",&q[i][j]);

}

}

printf("the array looks like this:\n");

for(i = 0;i < m;i++)

{

for(j = 0;j < n;j++)

{

printf(" %d",q[i][j]);

}

printf("\n");

}

for(;;)

//菜单和switch函数循环

{

switch(menu\_select())

{

case 1:

printf("the answer(side) is:");

printf("%d\n",sumside(q,m,n));

break;

case 2:

printf("the answer is(not close):");

printf("%d\n",sumupthedata(q,m,n));

break;

case 3:

printf("the answer is(diagonal):");

printf("%d\n",diagcater(q,m,n));

break;

case 0:

printf("good bye!");

exit(0);

//输出0时跳出循环，结束程序。

}

}

}

int menu\_select()

//菜单加选择

{

char a;

do{

printf(" sum up the side(input:1)\n");

printf(" sum up the data which not close by(input:2)\n");

printf(" sum up the diagonal data(input:3)\n");

printf(" if you want to stop(input:)0\n");

a = getchar();

a = getchar();//得到输入的数据

}while(a < '0'||a > '3');

return(a-'0');//不符合的方案

}

int sumside(int \*\*q, int m, int n)

//加边函数

{

int i = 0,sum = 0,j;

for(j = 0;j < n; j++)

sum +=q[i][j];

if(m == 1)

return sum;

while((++i) < m - 1)

{

sum += q[i][0] + q[i][n -1];

}

for(j = 0;j < n;j++)

sum += q[i][j];//i已经循环到最后了

return sum;

}

int sumupthedata(int \*\*q, int m, int n)

//加和不相邻数据函数

{

int i, j, sum = 0;

//偶数行

for(i = 0; i < n; i += 2)

for(j = 0;j < n; j += 2)

sum += q[i][j];

//奇数行

for(i = 1; i < m; i += 2)

for(j = 1;j < n; j += 2)

sum += q[i][j];

return sum;

}

int diagcater(int \*\*q, int m, int n)

//对角线函数

{

int i,sum = 0;

if(m != n)

{

//printf("m != n\n");打印不出来，无用。

return -1;

}

for(i = 0;i < m; i++)

{

if(i == n - 1 - i )//如果只有一行一列，或加到最中间

sum += q[i][i];

else

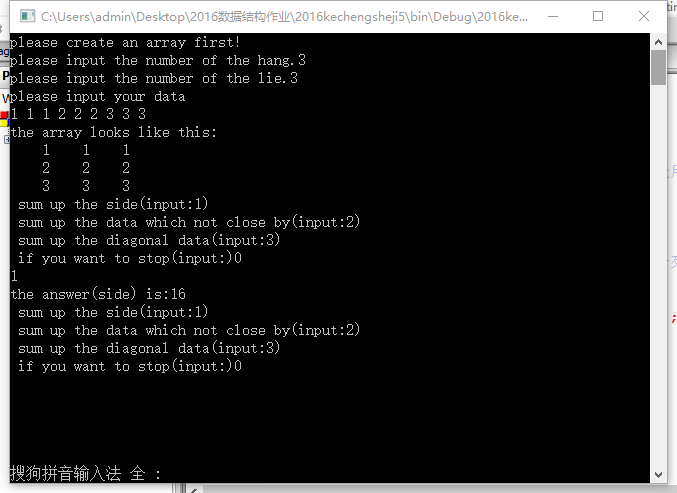
sum += q[i][i] + q[i][n - 1 -i];//加同一行对称的两边数据

}

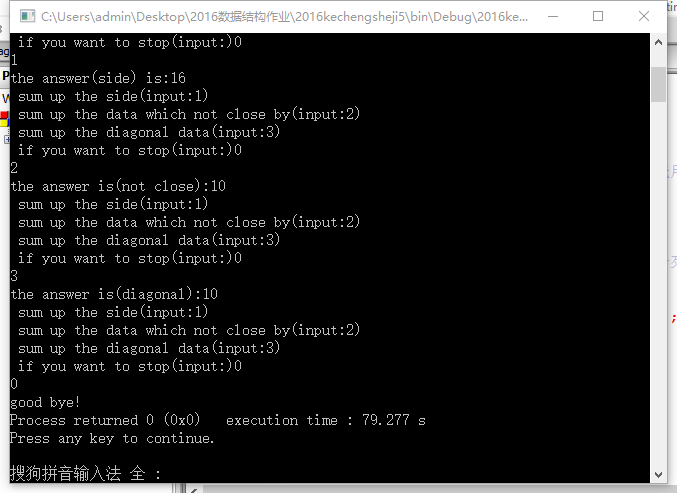
return sum;

}

1. 运行结果
2. 输入行数列数及其依次输入数组的数据，程序会自动生成二维数组
3. 按照菜单输入1会求出边相加之和，然后菜单会继续弹出等待命令



1. 输入2，求出不相邻数据相加之和
2. 输入3，求出对角线数据相加之和
3. 输入0，程序提醒goodbye,并继而键任意键退出



1. 收获及体会

在最开始的时候，思考不全面，在面对课题的时候不能整体思考，导致将函数框架书写错误。在计算边元素求和时，将I ,j 的过程没有想清楚。编程过程中深深感觉到自己知识的欠缺，对编程过程的不熟悉。虽然最后在大家的帮助下完成了本次课程设计，但是让我更加明白编程需要非常扎实的基本功。我应该在今后学习过程中更加细心，认真，多动脑。本次的课程设计让我受益匪浅，虽然期间遇到很多困难，程序进行的也很不容易，但是让我看到了编写程序中间的乐趣！希望自己可以在以后完成本课题更高级的内容。

八．参考文献

《C语言入门经典》 、《数据结构教程》