

一、题目：多功能科学计算器

二、内容：

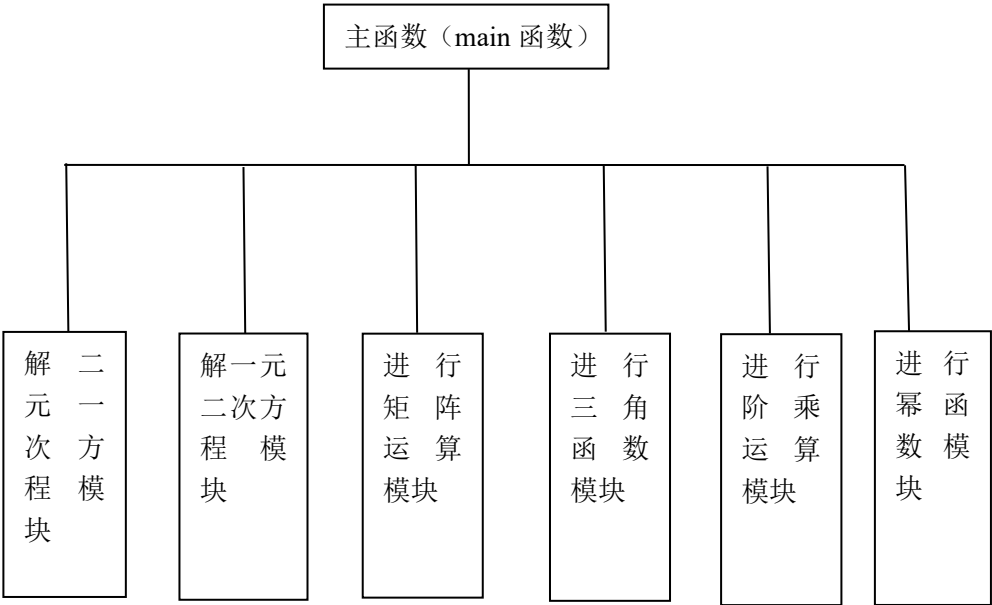
（1）概述或引言

普通计算器的使用已经不满足当下了，应当设计出有更多功能的计算器系统。

为了完成多功能科学计算器的使用，首先明确系统的定义。多功能科学计算器系统就是拥有一些普通计算器没有的计算功能，通过数据分析，来方便大家计算。该程序是基于 C 语言，采用面向过程程序设计思想，通过设计业务流程，将设计中待实现的功能划分成若干个子模块，并逐个编程实现，最后通过主函数调用，实现模块功能。

此次设计的程序也有很好地表达了作业的基本要求，整个程序也能简单不报错地运行，基本上已经实现了解二元一次方程、解一元二次方程、矩阵运算、三角函数运算、阶乘运算、幂函数运算，这些运算也已经都可以运行成功，但稍微有些许不足的地方就是只有在解二元一次方程的部分提供程序运行时出现的任何异常情况的处理方法，其他部分暂时未给出运行出现的异常处理方法，不过用户按要求输入的话一般的报错率都是很低的，基本上都能运算成功。

（2）程序概要设计



(3) 程序详细设计

用 switch 语句实现多分支选择结构时，如果用户输入的字符与规定选择都不相同，因加入执行 default 后面的语句，输出“输入数据有错”的信息。在每个 case 后的语句中，最后都有一个 break 语句来结束。

源代码：

```
default:
    printf("输入错误！请重新输入！\n");    //如果输入其他字符，则发出警告
    break;
```

使用三角函数的计算时，注意正切函数部分，当角度 $x = (\pi/2) + k\pi, k \in \mathbb{Z}$ 时，该正切值应当不存在，因暂无找到更好的办法，故程序暂时只强调了 90 度和 270 度。

源代码：

```
printf("请输入角度：\n");
scanf("%lf",&a);
j=tan(a*3.14/180);
if(a==90||a==270)
{
    printf("%.2f 的 tan 值不存在\n\n",a);
}
else
    printf("%.2f 的 tan 值为%.2f\n\n",a,j);
```

(4) 程序使用说明

1. 菜单模块

主函数模块首先调用 printMenu 函数，完成菜单的显示，然后根据用户输入调用不同的子函数实现相应的功能。

源代码：

```
void PrintMenu()    //定义 printMenu 函数
{
    printf("\n");
    printf("*****多功能科学计算器*****\n");
    printf("***    1.解二元一次方程                **\n");
    printf("***    2.解一元二次方程                **\n");
    printf("***    3.进行矩阵相加、相减、相乘运算        **\n");
    printf("***    4.进行三角函数                    **\n");
    printf("***    5.进行阶乘运算                        **\n");
    printf("***    6.进行幂函数                        **\n");
    printf("***    0.退出程序                            **\n");
    printf("*****\n");
}
```

运行结果：

```

*****多功能科学计算器*****
**      1. 解二元一次方程      **
**      2. 解一元二次方程      **
**      3. 进行矩阵相加、相减、相乘运算  **
**      4. 进行三角函数        **
**      5. 进行阶乘运算        **
**      6. 进行幂函数          **
**      0. 退出程序            **
*****

```

2. 解二元一次方程模块

print_A 函数的功能是解二元一次方程，通过让用户分别输入两个方程里未知数 x, y 的两个系数和一个常数，然后通过解方程组求出 x, y 的值，而后输出该值。

源代码：

```

void print_A()    //定义 print_A 函数
{
    int a,b,m,c,d,n;
    double x=0,y=0;
    printf("请依次输入第一个方程里 x, y 的系数 a b 和常数 m: \n");
    scanf("%d %d %d",&a,&b,&m);    //输入三个整数
    printf("请依次输入第二个方程里 x, y 的系数 c,d 和常数 n: \n");
    scanf("%d %d %d",&c,&d,&n);    //输入三个整数
    printf("%dx + %dy=%d\n",a,b,m);    //输出带有 x, y 的系数 a,b 和常数 m 的方程
    printf("%dx + %dy=%d\n",c,d,n);    //输出带有 x, y 的系数 c,d 和常数 n 的方程
    x=(m*d-b*n)/(a*d-b*c);
    y=(m*c-a*n)/(b*c-a*d);    //解方程组求出 x, y
    printf("\nx=%.2f\ty = %.2f\n",x,y);    //输出 x, y 的值
}

```

运行结果：

```

请依次输入第一个方程里x, y的系数a b和常数m:
2 1 5
请依次输入第二个方程里x, y的系数c,d和常数n:
1 1 3
2x + 1y=5
1x + 1y=3

x=2.00  y = 1.00

```

3. 解一元二次方程模块

print_B 函数的功能是解一元二次方程，通过让用户分别输入方程里的三个参数，而后通过公式法求出两个根的值。

源代码：

```

void print_B()    //定义 print_B 函数
{
    int a, b, c;
    double x1,x2,delta;

```

```

printf("请依次输入一元二次方程的三个参数 a b c: \n");
scanf("%d %d %d", &a, &b, &c);    //输入三个整数
delta = b*b - 4*a*c;    //通过公式法求出 delta
x1 = (-b + sqrt(delta))/(2 * a);
x2 = (-b - sqrt(delta))/(2 * a);    //通过公式法求出方程的两个根
if (delta>0)
{
    printf("一元二次方程有两个解\n");
    printf("一元二次方程的第一个解, x1 = %.2f\n",x1);
    printf("一元二次方程的第二个解, x2 = %.2f\n",x2);
}
else if (delta ==0)
{
    printf("一元二次方程有两个相同的解\n");
    printf("一元二次方程的解为 x1 = x2 = %.2f\n",x1);
}
else
{
    printf("一元二次方程无解\n");
}
}

```

运行结果:

① 两个解不一样的情况

```

请依次输入一元二次方程的三个参数a b c:
2 4 1
一元二次方程有两个解
一元二次方程的第一个解, x1 = -0.29
一元二次方程的第二个解, x2 = -1.71

```

② 两个解一样的情况

```

请依次输入一元二次方程的三个参数a b c:
2 4 2
一元二次方程有两个相同的解
一元二次方程的解为x1 = x2 = -1.00

```

③ 无解的情况

```

请依次输入一元二次方程的三个参数a b c:
2 2 1
一元二次方程无解

```

4. 矩阵运算模块

print_C 函数的功能是进行矩阵相加、相减、相乘运算，通过让用户先输入矩阵的维数，再分别输入两个矩阵的元素，然后输出用户输入的两个矩阵，再输出菜单选项让用户选择运算，最后根据用户选择的运算选项（相加、相减、相乘）进行计算。

源代码:

```
void print_C()    //定义 print_C 函数
{
    int n,m,c=0;
    int a[100][100],b[100][100];
    int fc[100][100];
    int i,j,k;
    printf("*输入矩阵*\n");
    printf("请输入矩阵的维数:");
    scanf("%d",&n);    //输入一个整数
    m=n*n;    //矩阵的个数 m
    printf("请输入第一个矩阵 a（共有%d 个元素）:\n",m);    //让用户输入第一个矩阵
    do    //循环体结构开始
    {
        m--;
        for(i=0;i<n;i++)
            for(j=0;j<n;j++)
            {
                scanf("%d",&a[i][j]);    //输入 a[i][j]的值
            }
    }while(m<0);    //当 m 小于 0 时，停止执行循环体

    printf("请输入第二个矩阵 b（共有%d 个元素）:\n",m);    //让用户输入第二个矩阵
    do    //循环体结构开始
    {
        m--;
        for(i=0;i<n;i++)
            for(j=0;j<n;j++)
            {
                scanf("%d",&b[i][j]);    //输入 b[i][j]的值
            }
    }while(m<0);    //当 m 小于 0 时，停止执行循环体

    printf("您输入的矩阵 a=\n");    //输出用户输入的矩阵 a
    for(i=0;i<n;i++)
        for(j=0;j<n;j++)
        {
            printf("%10d",a[i][j]);    //输入 a[i][j]的值
            if(j==n-1)
                printf("\n");
        }

    printf("您输入的矩阵 b=\n");    //输出用户输入的矩阵 b
    for(i=0;i<n;i++)
        for(j=0;j<n;j++)
        {
```

```

        printf("%10d",b[i][j]);    //输入 a[i][j]的值
        if(j==n-1)
            printf("\n");
    }

while(1)
{
    printf("\n");
    printf("1 矩阵相加\n");
    printf("2 矩阵相减\n");
    printf("3 矩阵相乘\n");
    printf("0 退出系统\n");
    printf("\n");
    printf("请选择运算 0~3:");
    scanf("%d",&c);    //输入 0~3 中的一个整数
    if(c==0)
    {
        break;
    }
    switch(c)
    {
        case 1:
            for(i=0;i<n;i++)
                for(j=0;j<n;j++)
                {
                    fc[i][j]=0;
                    fc[i][j]=a[i][j]+b[i][j];
                }
            printf("矩阵和为: \n");    //输出两个矩阵相加后的矩阵和
            for(i=0;i<n;i++)
                for(j=0;j<n;j++)
                {
                    printf("%10d",fc[i][j]);
                    if(j==n-1)
                        printf("\n");    //当 j 的值为 n-1 时，循环结束并转行
                }
            break;
        case 2:
            for(i=0;i<n;i++)
                for(j=0;j<n;j++)
                {
                    fc[i][j]=0;
                    fc[i][j]=a[i][j]-b[i][j];
                }
            printf("矩阵差为: \n");    //输出两个矩阵相减后的矩阵差
            for(i=0;i<n;i++)

```

```

        for(j=0;j<n;j++)
        {
            printf("%10d",fc[i][j]);
            if(j==n-1)
                printf("\n");    //当 j 的值为 n-1 时，循环结束并转行
        }
        break;
case 3:
    for (i=0;i<n;i++)
        for (j=0;j<n;j++)
        {
            fc[i][j]=0;
            for (k=0;k<n;k++)
                fc[i][j]+=a[i][k]*b[k][j];
        }
    printf("矩阵积为: \n");    //输出两个矩阵相乘后的矩阵积
    for(i=0;i<n;i++)
        for(j=0;j<n;j++)
        {
            printf("%10d",fc[i][j]);
            if(j==n-1)
                printf("\n");    //当 j 的值为 n-1 时，循环结束并转行
        }
        break;
default:
    printf("输入错误！请重新输入！\n");    //如果输入其他字符，则发出警告
    break;
}
}
}

```

运行结果:

① 运行输入输出

```

*输入矩阵*
请输入矩阵的维数:2
请输入第一个矩阵a（共有4个元素）:
1 2 3 4
请输入第二个矩阵b（共有3个元素）:
1 2 3 4
您输入的矩阵a=
    1      2
    3      4
您输入的矩阵b=
    1      2
    3      4

```

② 运算过程

```
1 矩阵相加
2 矩阵相减
3 矩阵相乘
0 退出系统

请选择运算0~3:1
矩阵和为:
      2      4
      6      8

1 矩阵相加
2 矩阵相减
3 矩阵相乘
0 退出系统

请选择运算0~3:2
矩阵差为:
      0      0
      0      0

1 矩阵相加
2 矩阵相减
3 矩阵相乘
0 退出系统

请选择运算0~3:3
矩阵积为:
      7      10
     15     22

1 矩阵相加
2 矩阵相减
3 矩阵相乘
0 退出系统

请选择运算0~3:4
输入错误! 请重新输入!

1 矩阵相加
2 矩阵相减
3 矩阵相乘
0 退出系统

请选择运算0~3:0

*****多功能科学计算器*****
```

5. 三角函数运算模块

print_D 函数的功能是进行三角函数的运算,通过让用户先选择运算,再输入三角函数的角度,而后通过*3.14/180 转角度方法运算出最终的三角函数式子。

源代码:

```
void print_D()    //定义 print_D 函数
{
    int i;
    while(1)
    {
        double a,j;
        printf("三角函数的运用\n");
        printf("1.正弦函数 sin\n");
        printf("2.余弦函数 cos\n");
        printf("3.正切函数 tan\n");
        printf("4.余切函数 cot\n");
        printf("5.正割函数 ses\n");
        printf("6.余割函数 csc\n");
        printf("0.退出系统\n");
        printf("请选择 0~6:");    //输入 0~6 中的一个整数
        scanf("%d",&i);
        if(i==0)
        {
            break;
        }
        switch(i)
        {
            case 1:
                printf("请输入角度: \n");
                scanf("%lf",&a);
                j=sin (a*3.14/180);
                printf("%.2f 的 sin 值为%.2f\n\n",a,j);
                break;
            case 2:
                printf("请输入角度: \n");
                scanf("%lf",&a);
                j=cos (a*3.14/180);
                printf("%.2f 的 cos 值为%.2f\n\n",a,j);
                break;
            case 3:
                printf("请输入角度: \n");
                scanf("%lf",&a);
                j=tan (a*3.14/180);
                if(a==90||a==270)
                {
                    printf("%.2f 的 tan 值不存在\n\n",a);
                }
                else
                    printf("%.2f 的 tan 值为%.2f\n\n",a,j);
            }
        }
    }
}
```

```

        break;
    case 4:
        printf("请输入角度: \n");
        scanf("%lf",&a);
        j=1/tan (a*3.14/180);
        printf("%.2f 的 cot 值为%.2f\n\n",a,j);
        break;
    case 5:
        printf("请输入角度: \n");
        scanf("%lf",&a);
        j=1/cos (a*3.14/180);
        printf("%.2f 的 ses 值为%.2f\n\n",a,j);
        break;
    case 6:
        printf("请输入角度: \n");
        scanf("%lf",&a);
        j=1/sin (a*3.14/180);
        printf("%.2f 的 csc 值为%.2f\n\n",a,j);
        break;
    }
}
}

```

运行结果:

```

三角函数的运用
1. 正弦函数sin
2. 余弦函数cos
3. 正切函数tan
4. 余切函数cot
5. 正割函数ses
6. 余割函数csc
0. 退出系统
请选择0~6:6
请输入角度:
15
15.00的csc值为3.87

```

```

三角函数的运用
1. 正弦函数sin
2. 余弦函数cos
3. 正切函数tan
4. 余切函数cot
5. 正割函数ses
6. 余割函数csc
0. 退出系统
请选择0~6:3
请输入角度:
90
90.00的tan值不存在

```

6. 阶乘运算模块

print_E 函数的功能是进行阶乘的运算，通过让用户输入要计算的阶乘数，而后通过循环相乘求出最终的值。

源代码：

```
void print_E()    //定义 print_E 函数
{
    int i=1,n;
    double x=1;
    printf("请输入要计算的阶乘数 n: ");
    scanf("%d",&n);    //输入一个整数
    while(i<=n)
    {
        x=x*i;
        i++;    //循环体
    }
    printf("n 的阶乘 x=%.2f\n",x);    //输出 x 的值
}
```

运行结果：

```
请输入要计算的阶乘数n: 10
n的阶乘x=3628800.00
```

7. 幂函数运算模块

print_F 函数的功能是进行幂函数的运算，通过让用户输入要计算的幂函数的底数和幂，而后通过数学函数方法求出最终的幂函数式子。

源代码：

```
void print_F()    //定义 print_F 函数
{
    double x, y, z;
    printf("请输入幂函数的底数 x 和幂 y: \n");
    scanf("%lf %lf",&x,&y);    //输入两个浮点型数据
    z = pow( x, y );
    printf( "%.2f^%.2f = %.2f\n", x, y, z );    //输出最终的幂函数式子
}
```

运行结果：

```
请输入幂函数的底数x和幂y:
2 2
2.00^2.00 = 4.00
```

（5）总结

这个计算机程序也许对别人来说是简简单单的，但对我来说却是一次很大的突破。经过多天的努力编写修修补补，程序的系统运行也能很好使用了，也成功的都能使用更多功能的科学计算。整个程序过程都挺清晰明了的，暂时没有任何很大错误或问题的地方，每个子函数也均能正常运行。

三、参考文献

作者：谭浩强

书名[M]：C 程序设计（第五版）

出版地：北京清华大学学研大厦 A 座

ISBN 978-7-302-48144-7

出版社：清华大学出版社

出版年：1991.7.1