



实验 1 简单判定性问题求解

一、实验学时

完成本实验需4学时。

二、实验目的

1、阅读程序题

(1) 掌握C语言数据类型，熟悉如何定义一个整型、字符型的变量，以及对它们赋值的方法；

(2) 掌握不同的类型数据之间赋值的规律；

(3) 掌握数据在内存中的存储方式；

(4) 学会输入、输出函数的基本格式和使用方法；

(5) 学会使用有关算术运算符、逻辑运算符、关系运算符，以及包含这些运算符的表达式。

2、编程题

(1) 如何运用if-else判定性结构进行程序设计；

(2) 如何运用switch判定性结构进行程序设计。

3、调试题

(1) 熟悉C程序的编辑、编译、连接和运行的过程。

三、实验指导

为了达到最佳的实验效果，以下提供几条适于编程的指导意见，可供参考。

1、阅读程序题应先运用自己在课堂所学的知识，推导出结果，在上机时输入计算机，印证自己推导的结果，注意观察数据在内存中的存储方式、含不同种运算符表达式的输出结果。

2、编程题必须首先画出流程图，并反复思考判断程序设计的正确性，完成程序的设计。要注意简单判定性问题的结构选择。

3、调试题应明确程序的调试、测试是一项非常烦琐的工作，也是非常重要的工作。对于初学者来说应该建立良好的习惯，在调试程序的时候，应该尽可能考虑到程序运行时各种可能情况。

四、实验内容



1、阅读程序题

(1) main()

```
{
    /*定义字符型变量*/
    char c1,c2;      /*向字符变量赋以整数*/
    c1=97;
    c2=98;
    printf("%c %c\n",c1,c2); /*以字符形式输出*/
    printf("%d %d\n",c1,c2); /*以整数形式输出*/
}
```

该程序的输出结果是 a b(换行)97 98。

思考：可否改成int c1, c2；输出结果是？ 相同

(2) main()

```
{
    int a=7,b=5;
    printf("%d\n",b=b/a);
}
```

该程序的输出结果是 0。

思考：若将printf语句中%d变为%f，可否输出分式的值？ 可以

(3) main()

```
{
    int a=9;
    a+=a-=a+a; /*包含复合的赋值运算符的赋值表达式*/
    printf("%d\n",a);
}
```

该程序的输出结果是 -18。

思考：赋值表达式 $a += a - = a + a$ 的求解步骤？

第一步： $a = a - (a + a) = -9$ 第二步 $a = a + a = 18$

(4) main()

```
{
    int k=-1;
    printf("%d,%u\n",k,k);
}
```



思考：-1在内存中的存储形式？

```
(5) main()
{
    unsigned x1;
    int b=-1;
    x1=b; /*有符号数据传送给无符号变量*/
    printf("%u",x1);
}
```

思考：不同类型的整型数据相互赋值时，规则是什么？

```
(6) main()
{
    int a=3,b=4,c=5;
    printf("%d\n",(! (a<b))&&(c=c+1));
    printf("%d",c);
}
```

思考：若将第一个printf语句中&&变为||，结果将如何？

```
(7) main()
{ int a=1,b=2,c=3;
  if(c=a) ; printf("%d\n",c);
  else printf("%d\n",b);
}
```

2、编程序并上机调试运行(包括题目及要求、流程图、程序清单、测试数据及结果)

(1) 将一个百分制成绩转换成等级制成绩，百分制与等级制的对应关系如下表：

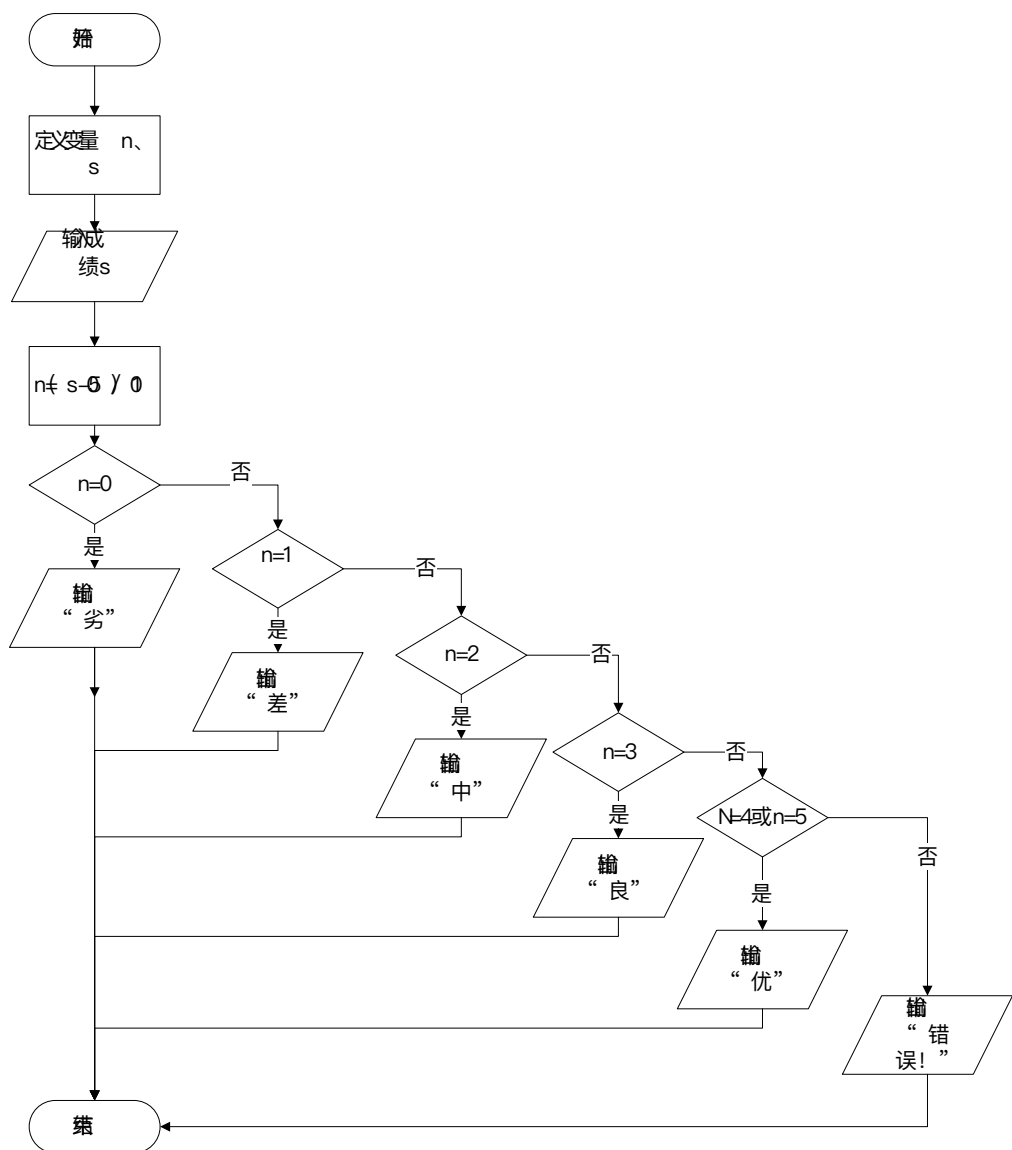
百分制	90-100	80-89	70-79	60-69	<60
等级制	优	良	中	差	劣 



- 1) 用switch语句实现该功能；
- 2) 用键盘输入百分制成绩，输入前要有提示信息，对于不合理的数据应输出错误信息；
- 3) 输出结果应包含百分制成绩和成绩等级；
- 4) 分别输入成绩-10, 30, 60, 68, 75, 83, 95, 102, 查看记录运行结果。

<p style="text-align: center;">流程图</p> <p style="text-align: center;">见下页</p>	<p style="text-align: center;">程序代码</p> <pre> #include< #include<stdlib.h> void main() { float score; int level; printf("请输入成绩:\n"); scanf("%f",&score); if (score>100 score<0) printf("成绩输入错误! \n"); else { level=(int)((score-50)/10); switch(level) { case 5: printf("优\n");break; case 4: printf("优\n");break; case 3: printf("良\n");break; case 2: printf("中\n");break; case 1: printf("差\n");break; default: printf("劣\n");break; } } } </pre>
<p>测试数据: 100, 50, -20 输出结果: 优, 劣, 成绩输入错误</p>	





(2) 银行整存整取存款不同期限的月息利率分别为：

输入存款的本金和年 月息利率 = 的利息与本金的合 利息 = 本金 × 月息利	$\left\{ \begin{array}{ll} 0.63\% & \text{期限} = 1\text{年} \\ 0.66\% & \text{期限} = 2\text{年} \\ 0.69\% & \text{期限} = 3\text{年} \\ 0.75\% & \text{期限} = 5\text{年} \\ 0.84\% & \text{期限} = 8\text{年} \end{array} \right.$	限 = 1年 限 = 2年 限 = 3年 限 = 5年 限 = 8年	限，求到期时能从银行得到 计。（利息的计算公式为： 率 × 12 × 存款年限。
--	--	--	--

程序代码



见下页 流程图

测试数据:

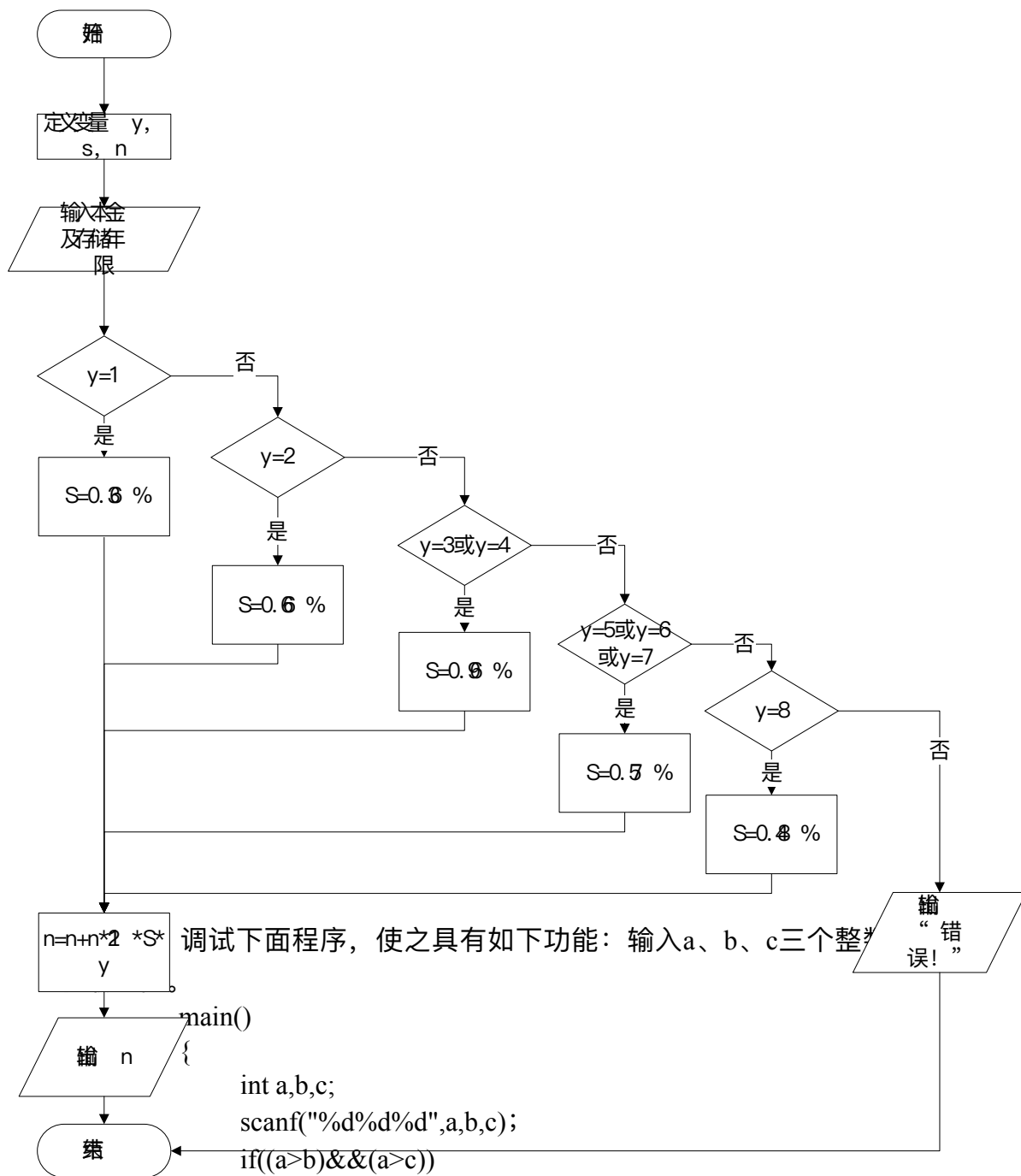
4,100 04, -10

输出结果:

133.12 本金金额不能为负!

```
#include<stdio.h>
#include<stdlib.h>
void main()
{
    int nYear;
    float fInput;
    double fS,fOut;
    printf("输入存储年份: \n");
    scanf("%d",&nYear);
    printf("输入本金数额: \n");
    scanf("%f",&fInput);
    if (fInput<0)
        printf("本金金额不能为负! \n");
    else
    {
        switch(nYear)
        {
            case 1: fS=0.63/100;break;
            case 2: fS=0.66/100;;break;
            case 3: ;
            case 4: fS=0.69/100;break;
            case 5: ;
            case 6: ;
            case 7: fS=0.75/100;break;
            case 8: fS=0.84/100;;break;
            default:printf("输入错误! ");return;}
        fOut=fInput+fInput*12*nYear*fS;
        printf("合计: %f\n",fOut);
    }
}
```





调试下面程序，使之具有如下功能：输入a、b、c三个整数

```

main()
{
    int a,b,c;
    scanf("%d%d%d",a,b,c);
    if((a>b)&&(a>c))
        if(b<c)
            printf("min=%d\n",b);
        else
            printf("min=%d\n",c);
    if((a<b)&&(a<c))
        printf("min=%d\n",a);
}
  
```



程序中包含有一些错误，按下述步骤进行调试。

- 1) 设置断点。
- 2) 通过单步执行，观测变量，发现程序中的错误。

程序调试记录

1、程序出现的错误及改正

第四行，abc前都加&，在%d%d%d中两个%d之间加空格；{ }括住六到九行；把第十行if改成else if；

2、程序的改进

1.每个<或>后加一个=；

2.在程序末尾添加如下指令

```
else if((a<=b)&&(a>=c)|| (a>=b)&&(a<=c))
{
    if (b<=c)
        printf("min=%d\n",b);
    else
        printf("min=%d\n",c);
}
```





实验2 循环结构及其应用

一、实验学时

完成本实验需4学时。

二、实验目的

- 1、熟练掌握while, do-while, for语句实现循环的方法, 包括如何正确地设定循环条件, 如何控制循环的次数。
- 2、掌握break语句与continue语句的使用方法。
- 3、熟悉循环的嵌套以及从循环体内退出循环的处理。
- 4、理解什么是循环程序结构。

三、实验指导

为了达到最佳的实验效果, 以下提供几条适于模块化编程的指导意见, 可供参考。

1、阅读程序题应先运用自己在课堂所学的知识, 推导出结果, 在上机时输入计算机, 印证自己推导的结果, 注意循环语句的一般使用方法, 以及复合运算符的使用。

2、编程题必须首先画出流程图, 并反复思考判断程序设计的正确性, 以面向过程的、模块化设计方法完成程序设计。

【例题解析】

编写一个程序, 输出半径为1到10的圆的面积, 若面积在40到90之间则予以输出, 否则, 不予输出。

1) **解题分析:** 解决问题的关键算法为求圆的面积并按要求输出。循环求圆的面积area, 若 $area < 40$, 则用continue语句提前结束本次循环并开始新一轮循环, 若 $area > 90$, 则用break语句跳出循环。

2) **算法流程图如图2-1:**

3) **综合分析**

需要从循环体中提前跳出循环, 或者在满足某种条件下, 不执行循环中剩下的语句而立即从头开始新一轮循环, 这时就要用到break和continue语句。



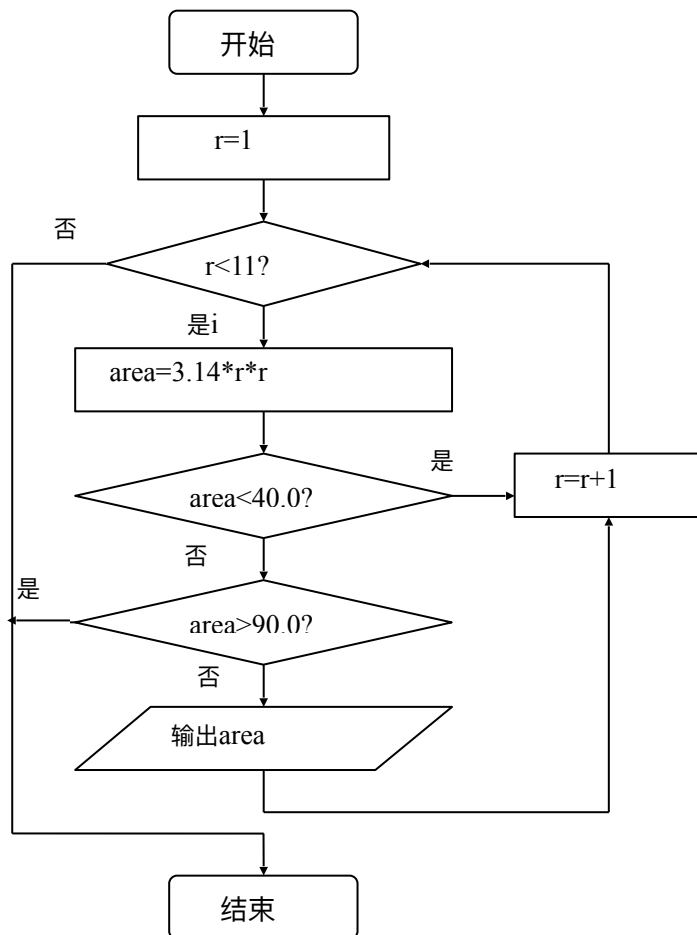


图2-1 输出半径为1到10且面积在40 到90之间的圆面积的流程图

4) 程序代码

```

#include <stdio.h>
main (void){
    float area=0.0,pi=3.14;
    int r;
    for(r=1;r<11;r++){
        area=pi*r*r;
        if (area<40)
            continue;
        if (area>90)
            break;
        printf("area=%5.2f\n",area);
    }
}

```

四、实验内容



1、阅读程序题。

1.

```
#include<stdio.h>
main( ){
    int x=-1;

    do{

        x=x*x;

    } while(!x);

    printf("%x=\n",x);

}
```

该程序的输出结果是 1=。

2.

```
#include<stdio.h>

main( ){
    int num=0;
    while(num<=2){
        num++;
        printf("%d\n",num);
    }
}
```

该程序的输出结果是 1 (换行) 2 (换行) 3。

3.

```
#include<stdio.h>
main( ){
    int a,b;
    for (a=1,b=1 ; a<=100 ; a++) {
        if (b>=20) break;
        if (b%3==1) { b+=3 ; continue ; }
        b-=5;
    }
    printf("%d\n",a);
}
```



该程序的输出结果是 8。

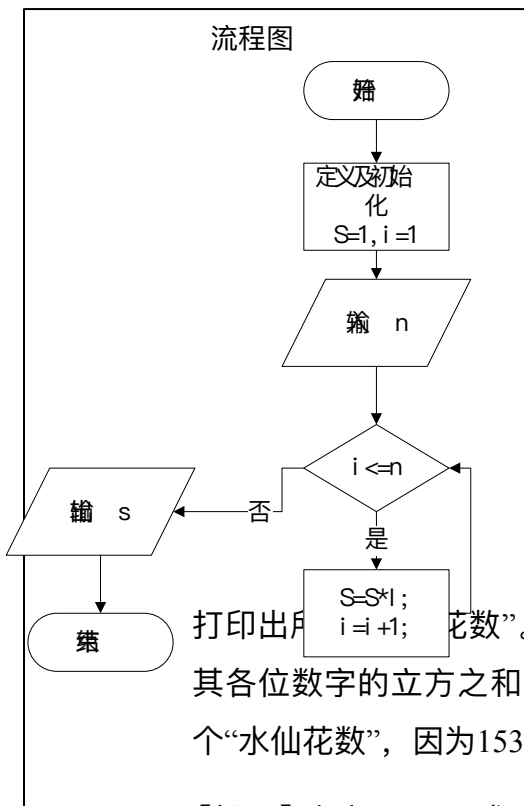
4.

```
#include<stdio.h>
main( ){
    int y=2,a=1;
    while (y--!=1)
    do {
        a*=y ;
        a++ ;
    } while (y--);
    printf("%d,%d\n",a,y);
}
```

该程序的输出结果是 1, -2。

2、编程序并上机调试运行(包
测试数据及结果)

1. 输入整数n, 求 n! (1<



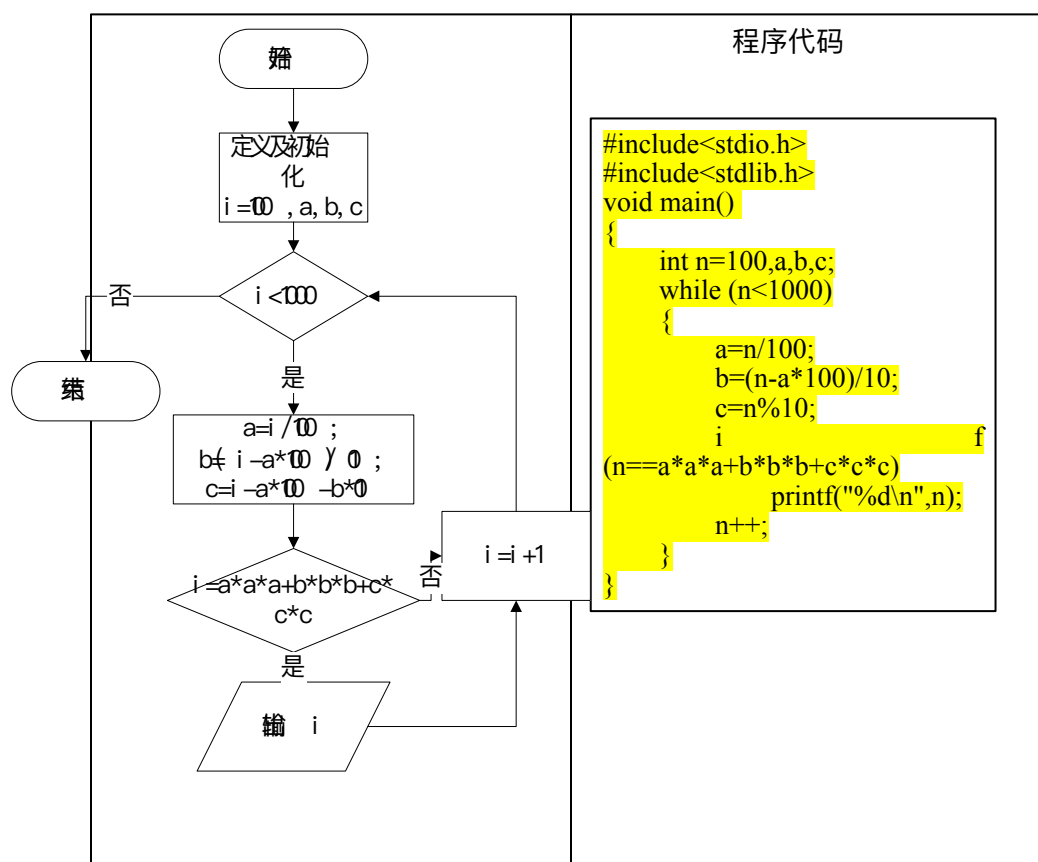
```
#include<stdio.h>
#include<stdlib.h>
void main()
{
    int i=1,n,nResult=1;
    printf("请输入n(0<n<20):\n");
    scanf("%d",&n);
    if (n<=0||n>=20)
        printf("输入错误\n");
    else
    {
        while (i<=n)
        {
            nResult*=i;
            i++;
        }
        printf("%d\n",nResult);
    }
}
```

【提示】根据题目要求只要分别求出一个三位数的个位、十

位、百位上的数字，然后判断是否满足（某一三位数 $a=a$ 的百



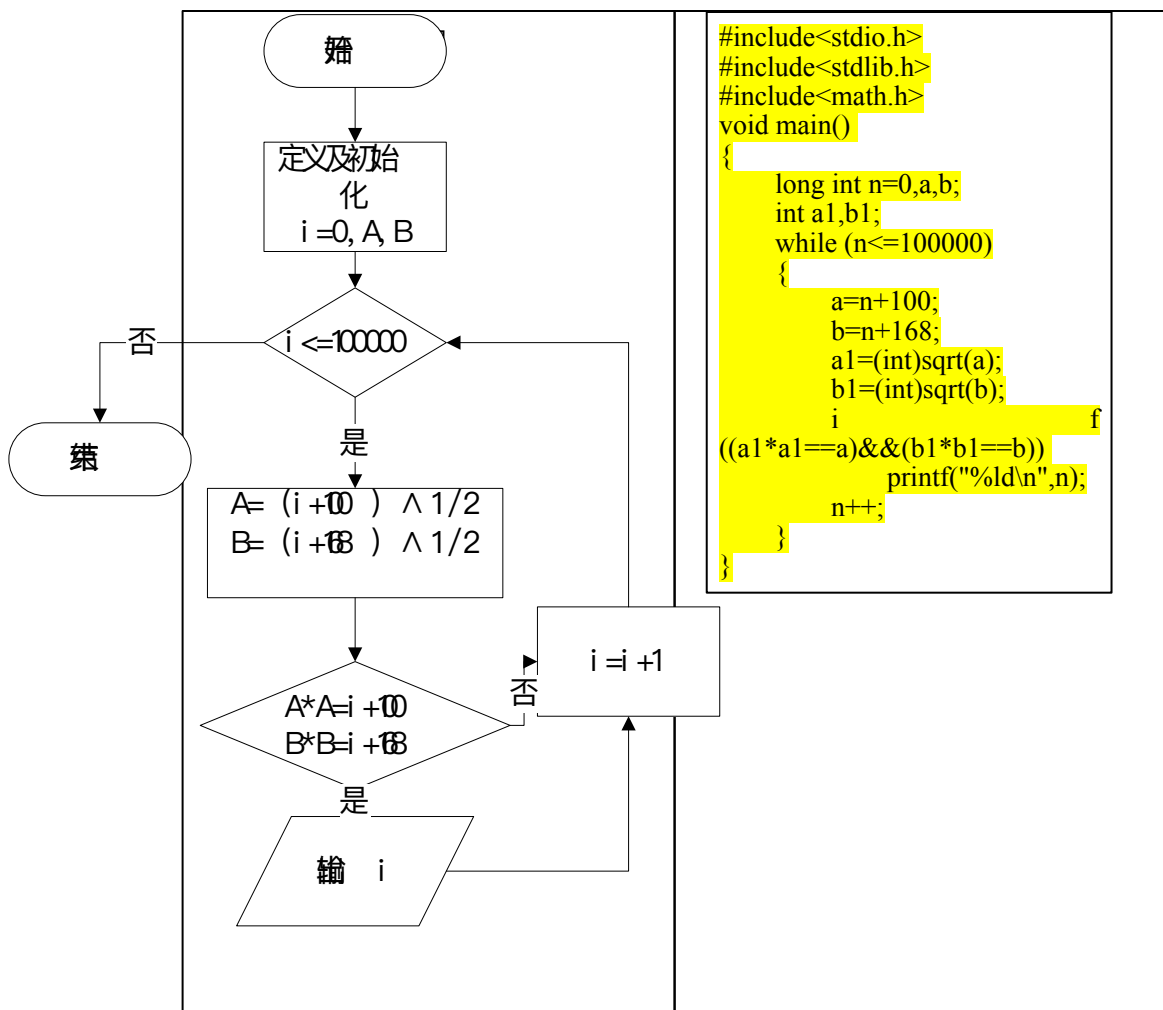
位的立方+a的十位的立方+a的个位的立方) 这个公式，满足这个条件的三位数就是“水仙花数”。



3. 一个整数，它加上100后是一个完全平方数，再加上168又是一个完全平方数，请问该数是多少？

【提示】在10万以内判断，先将该数加上100后再开方，再将该数加上268后再开方，如果开方后的结果满足如下条件，即是结果。





3、程序选做题

1. 猴子吃桃问题。猴子第一天摘下若干个桃子，当即吃掉了一半，还不过瘾，又多吃了一个。第二天早上又将剩下的桃子吃掉一半，又多吃了一个。以后每天早上都吃了前一天剩下的总数一半零一个。到第10天早上想再吃时，见只剩一个桃子了。求第一天共摘了多少桃子。

【提示】 使用逆向推导的方式，设后一天的桃子为N个，则前一天的桃子为2N+1个。



第二题

```
#include<stdio.h>
#include<stdlib.h>
void main()
{
    int a=0,b=0,c=0,d;
    while (a<=15)
    {
        for(b=0;b<=25;b++)
        {
            d=14*a+8*b;
            c=100-a-b;
            if (d==200)
                printf("%d,%d\n",a,b,c);
            else
                continue;
        }
        a++;
    }
}
```

其前面一天的桃子为 $(N+1) * 2$ 个，以此类推。

2. “百钱买百鸡”问题：公鸡每只5元，母鸡每只3元，小鸡每只1元；用100元买100只鸡，问公鸡，母鸡，小鸡各买多少只？

【提示】设公鸡 x 只，母鸡 y 只，小鸡 z 只；则有 $1 \leq x \leq 18$ ， $1 \leq y \leq 31$ ， $1 \leq z \leq 98$ ，且同时满足 $5x + 3y + z/3 = 100$ ， $x + y + z = 100$ ， $z \% 3 = 0$ 这三个条件。通过数学运算可改变条件，从而可用不同的方法来解决此问题，请尝试不同的循环次数解决该问题。

五、程序调试记录

要求：写出题号，并编写顺序号。如四（1）出现的错误

(1)

(2)

1、程序出现的错误及改正

2、程序的改进





实验3 模块化设计与应用

一、实验学时

完成本实验需要4学时。

二、实验目的

1、阅读程序题

- (1) 掌握函数实参和形参的对应关系以及“值传递”的方式；
- (2) 理解函数调用的过程以及函数的返回值；
- (3) 弄清楚带参函数和带参的宏之间的异同。

2、编程题

- (1) 深刻理解模块化程序设计的思想及如何进行模块划分；
- (2) 掌握定义函数的方法以及函数原型的声明；
- (3) 灵活运用函数调用(有参函数和无参函数、有返回值的函数和无返回值的函数)。

三、实验指导

为了达到最佳的实验效果，下面提出几点指导意见或要求以供参考：

1、阅读程序题应先运用自己在课堂所学的知识，推导出结果。上机时输入计算机，验证自己推导的结果是否正确。若不正确，应分析原因。

2、编程题注意事项：

- (1) 一个模块(或函数)应有一个入口点和一个退出点。
 - (2) 一般每个模块(或函数)只执行一个任务。不要将无关的任务放于同一模块中，只将完成同一任务的语句组合在一起。
 - (3) 先画流程图，后写程序代码；
 - (4) 变量和函数标示符尽量“见名知义”；
 - (5) 程序中应有适当注释。
- 3、应记录调试程序时出现的错误，以便经验的积累。

四、实验内容

1、阅读程序题

- (1) 下面程序的输出结果是 5.8。




```

#include <stdio.h>
int f(int m)
{
    int k=2;
    k++;
    return k+m;
}

void main( )
{
    int s;
    s=f(2);
    printf("%d, %d\n", s, f(s));
}

```

(2) 下面程序的输出结果是 777。

<pre> #include <stdio.h> int f(int a) { int b=0; int c=3; b=b+1; c=c+1; return(a+b+c); } </pre>	<pre> void main() { int a=2,i; for(i =0;i <3;i++) printf("%d",f(a)); } </pre>
---	---

(3) 下面程序的输出结果是 3 (空格) 5。

<pre> #include <stdio.h> void swap(int *a, int *b) { int *t; t=a;a=b;b=t; } </pre>	<pre> { int x=3,y=5,*p=&x,*q=&y; swap(p,q); printf("%d %d\n",*p,*q); } </pre>
--	---

void main()

(4) 下面程序的输出结果是 12。



```
#include <stdio.h>
#define M(x,y,z) x*y+z
void main()
{
    int a=1,b=2,c=3;
    printf("%d\n", M(a+b,b+c,c+a));
}
```

(5) 下面程序的输出结果是 19。

```
#include <stdio.h>
int M(int x, int y, int z)
{
    int p;
    p=x*y+z;
    return(p);
}

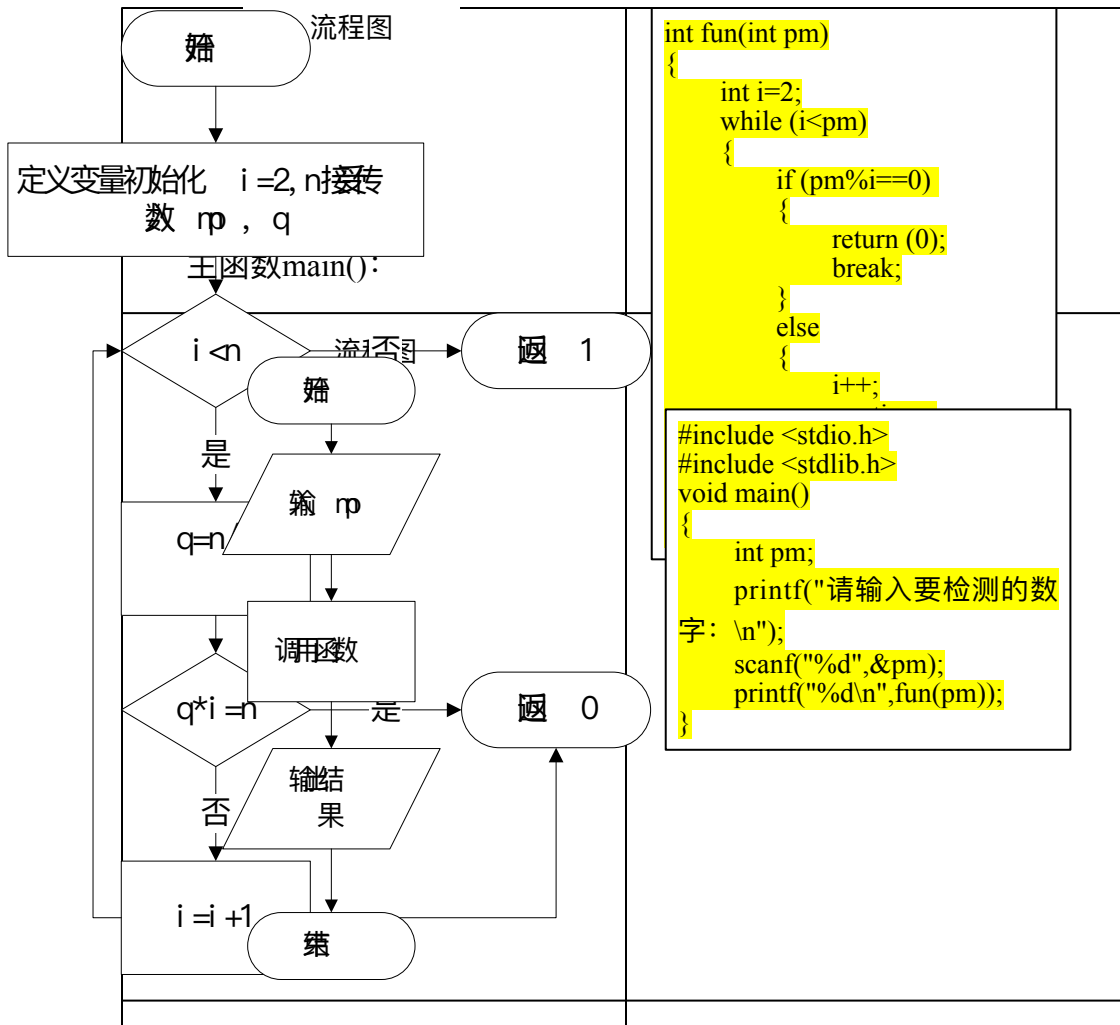
void main()
{
    int a=1,b=2,c=3;
    printf("%d\n", M(a+b,b+c,c+a));
}
```

2、编程序并上机调试运行。

(1) 请设计一个函数fun()，它的功能是：判断pm是否是素数。若pm是素数，返回1；若不是素数，返回0。pm的值由主函数从键盘读入。

函数fun()：





测试数据及输出结果：

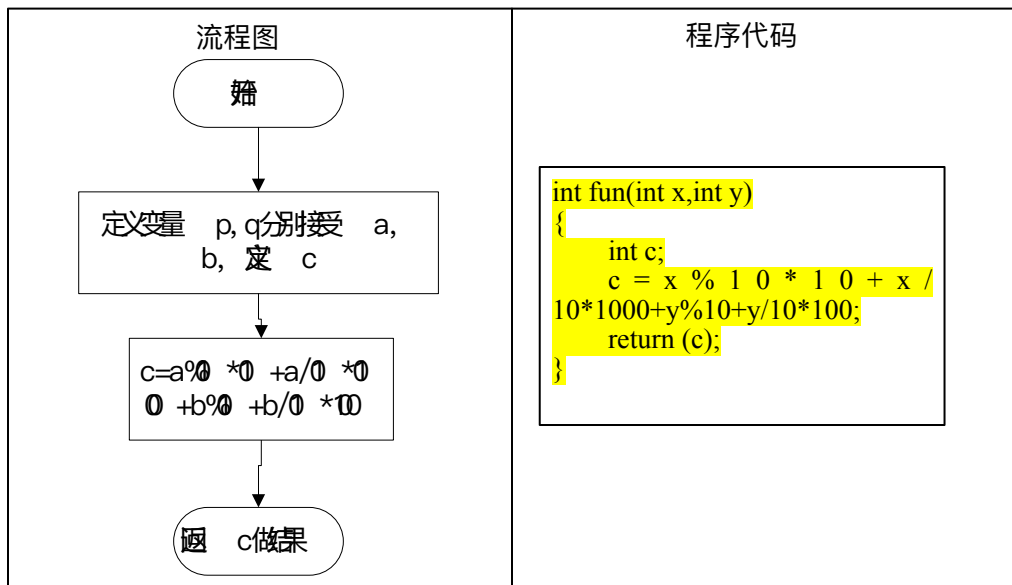
```

33 0
37 1
        
```

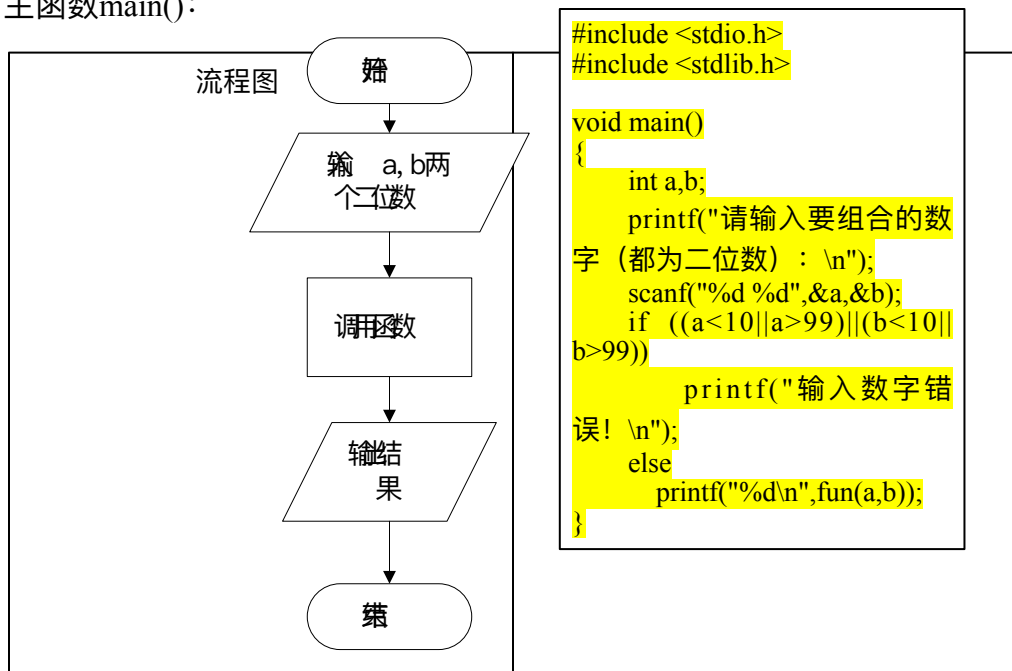
(2) 请设计一个函数fun(), 它的功能是：将两个两位数的正整数a、b合并形成一个整数放在c中。合并的方式是：将a数的十位和个位数依次放在c数的千位和十位上， b数的十位和个位数依次放在c数的百位和个位上。例如，当a=45，b=12。调用该函数后，c=4152。



函数fun():



主函数main():



测试数据及输出结果:

```

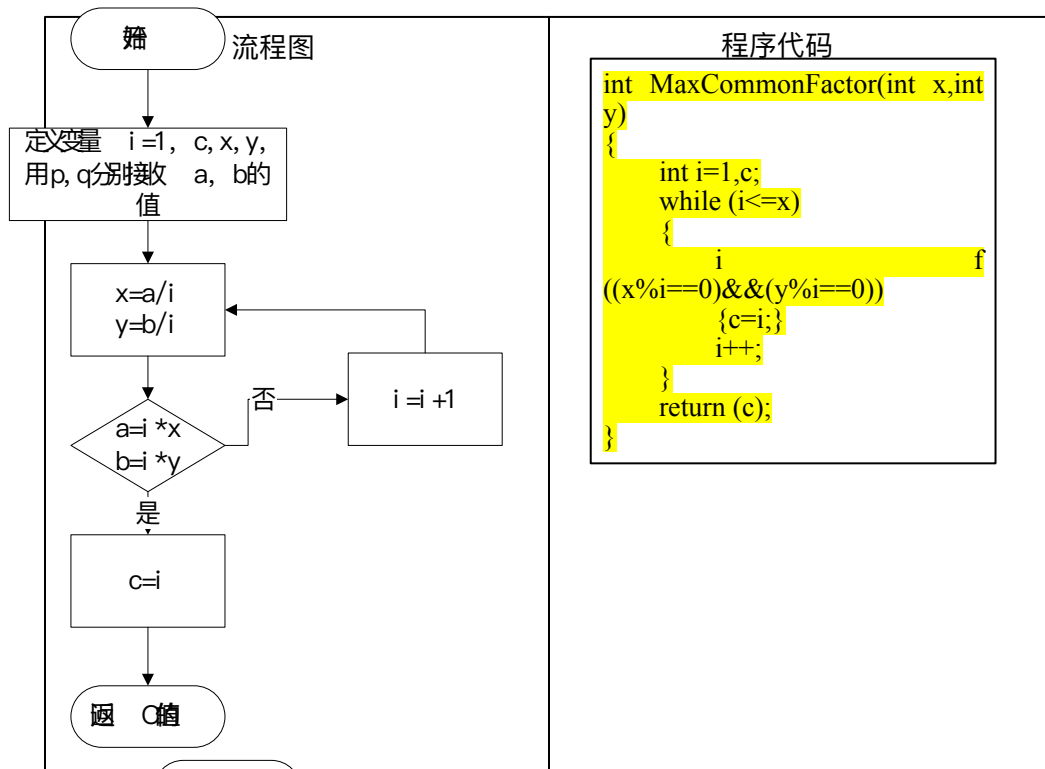
9 21 输入数字错误!
21 23      2123
    
```

(3) 请设计两个函数MaxCommonFactor()和MinCommonMultiple(), 它们的功能是: 前者用于求两个正整数的最大公约数, 后者用于求两个

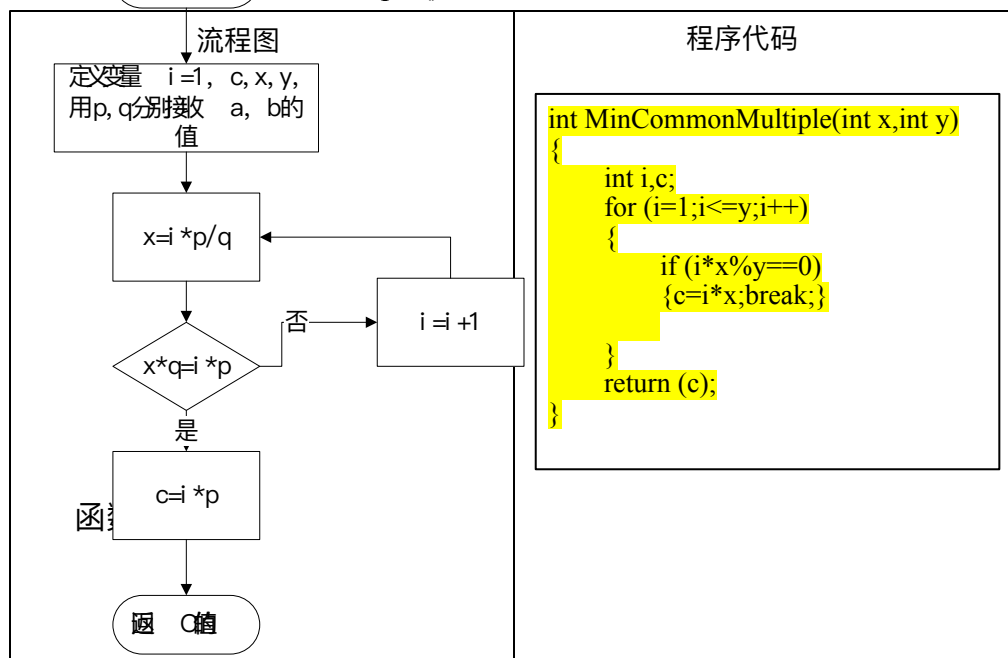


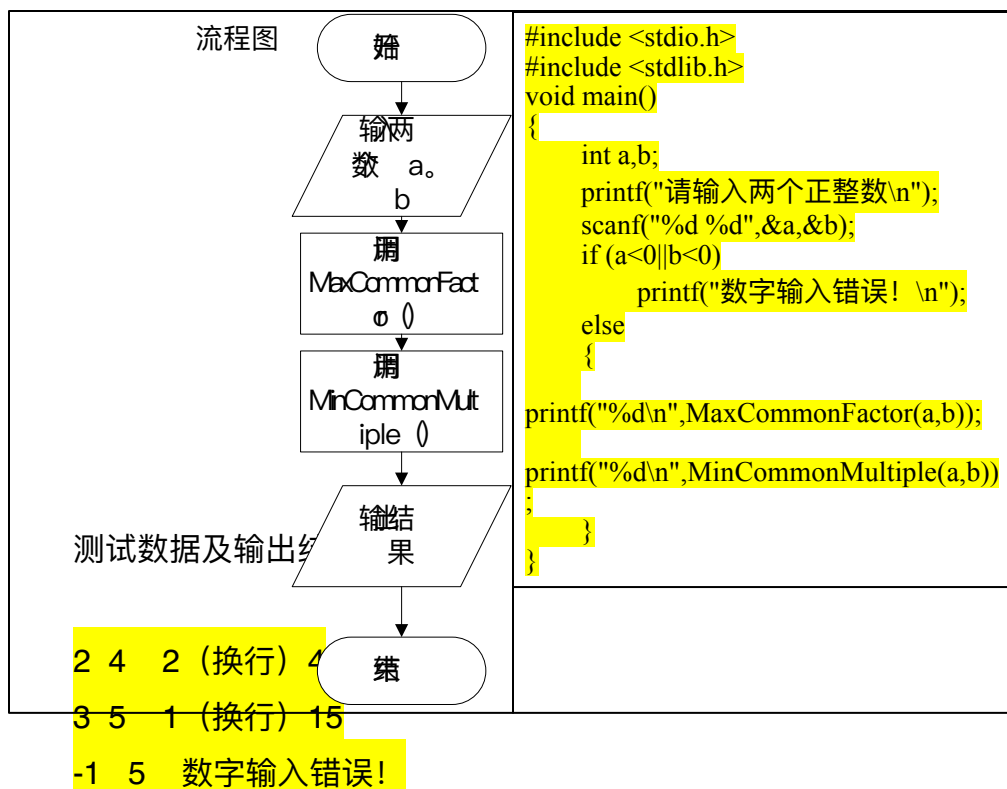
正整数的最小公倍数。

函数MaxCommonFactor():



函数MinCommonMultiple():





五、程序调试记录

要求记录以下内容：

- 1、程序出现的错误及改正
- 2、程序的改进





实验4 相同数据类型集合

一、实验学时

完成本实验需4学时。

二、实验目的

1、阅读程序题

- (1) 掌握一维和二维数组的定义、赋值和输入输出
- (2) 学会使用一维数组、二维数组
- (3) 学会使用多维数组、字符数组
- (4) 学会使用字符串处理函数

2、编程题

- (1) 如何运用数组进行程序设计
- (2) 了解字符串处理函数

三、实验指导

为了达到最佳的实验效果，以下提供几条适于模块化编程的指导意义，可供参考。

1、阅读程序题应先运用自己在课堂所学的知识，推导出结果，在上机时输入计算机，印证自己推导的结果，注意数组下标的使用方法。

2、编程题必须首先画出流程图，并反复思考判断程序设计的正确性，注意数组下标不要越界（为了加深认识，自己可以尝试一下下标越界的情况）。

3、调试程序要有耐心，反复的调试过程，虽然表面看起来费时费力，但是你一定受益匪浅。

四、实验内容

1、阅读程序题

(1) main()

```
{char ch[7]={"65ab21"};
int i,s=0;
for(i=0;ch[i]>='0'&&ch[i]<='9';i+=2)
s=10*s+ch[i]-'0';
```



```
printf("%d\n",s);}

```

该程序的输出结果是 6。

(2) #define MAX 5

```
main()

```

```
{int a[MAX],i;

```

```
for(i=0;i<5;i++)

```

```
    a[i]=i;

```

```
    printf("\n*****a*****\n");

```

```
for(i=0;i<5;i++)

```

```
{

```

```
    printf("  a[%d]=",i);

```

```
    printf("%d\n",a[i]);

```

```
}
```

```
}
```

该程序的输出结果是

(3) #include "stdio.h"

```
#define N 10

```

```
main()

```

```
{

```

```
    int i,j,temp;

```

```
    int a[N+1];

```

```
    int count=0;

```

```
    printf(" input  %d  data : \n",N);

```

```
    for (i=1;i<=N;i++)

```

```
        scanf("%d",&a[i]);

```

```
    printf("*****sort course*****\n");

```

```
    for(i=1;i<=N;i++)

```

```
{

```

```
    count++;

```

```
    for(j=1;j<=N-i;j++)

```

```
        if(a[j]>a[j+1])

```

```
        { temp=a[j];

```

```
          a[j]=a[j+1];

```

```
*****a*****
a[0]=0
a[1]=1
a[2]=2
a[3]=3
a[4]=4

```




```

        a[j+1]=temp;
    }
    printf("%3d:",count);
    for(j=1;j<=N;j++)
        printf(" %d",a[j]);
    printf("\n");
}
printf("the result is:\n");
for(i=1;i<=N;i++)
    printf("%d",a[i]); }

```

该程序的输出结果是 冒泡排序显示过程，并最终把输入的十个数字由小到大无空格排列。

(4) #include "stdio.h"

```
#define N 20
```

```
main()
```

```
{
```

```
    int f[N],i;
```

```
    f[0]=1;
```

```
    f[1]=1;
```

```
    for(i=2;i<N;i++)
```

```
        f[i]=f[i-1]+f[i-2];
```

```
        printf("\n-----Fibonacci-----
```

```
        \n");
```

```
    for(i=0;i<N;i++)
```

```
    {
```

```
        if(i%4==0)    printf("\n
```

```
        printf("f[%d]=%-5d
```

```
    }
```

```
}
```

该程序的运行结果为_____

(5) #include "stdio.h"

```
main()
```

```
{
```

```
    int i,j, temp;
```

```

-----Fibonacci-----
f[0]=1  f[1]=1  f[2]=2  f[3]=3
f[4]=5  f[5]=8  f[6]=13  f[7]=21
f[8]=34  f[9]=55  f[10]=89  f[11]=144
f[12]=233  f[13]=377  f[14]=610  f[15]=987
f[16]=1597  f[17]=2584  f[18]=4181  f[19]=6765

```

```

int a[3][3]={11,12,13},{21,22,23},{31,32,33}};
printf("---matrix a---\n");
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
        printf("%3d",a[i][j]);
    printf("\n\n");
}
for(i=0;i<3;i++)
    for(j=0;j<i;j++)
        { temp=a[i][j];
          a[i][j]=a[j][i];
          a[j][i]=temp;
        }
printf("---matrix a changed---\n");
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++) printf("%3d",a[i][j]);
    printf("\n\n");
}

```

该程序的输出结果是_____

```

(6) #include "stdio.h"
main()
{ char s[40];
int i;
printf("input string:");
for(i=0;(s[i]=getchar())!='\n';i++)
s[i]='\0';
printf("\noutput string:");
for(i=0;s[i]!='\0';i++)
{ if(s[i]>='a'&& s[i]<='z')
    s[i]=s[i]-32;
}
}

```

```

---matrix a---
11 12 13
21 22 23
31 32 33

---matrix a changed---
11 21 31
12 22 32
13 23 33

```



```
else if(s[i]>='A'&& s[i]<='Z')
```

```
    s[i]=s[i]+32;
```

```
    printf("%c",s[i]);
```

```
}
```

} 该程序的运行结果为 将输入大写字母转变为小写，小写字母

转变为大写。

2、编程序并上机调试运行(包括题目测试数据及结果)

(1) 编写一个3*4矩阵，找出每行最大值并交换。具体要求如下：

(a) 使用二维数组存放该3*4矩阵

(b) 定义并初始化该二维数组；

(c) 输出原矩阵和变换后的矩阵

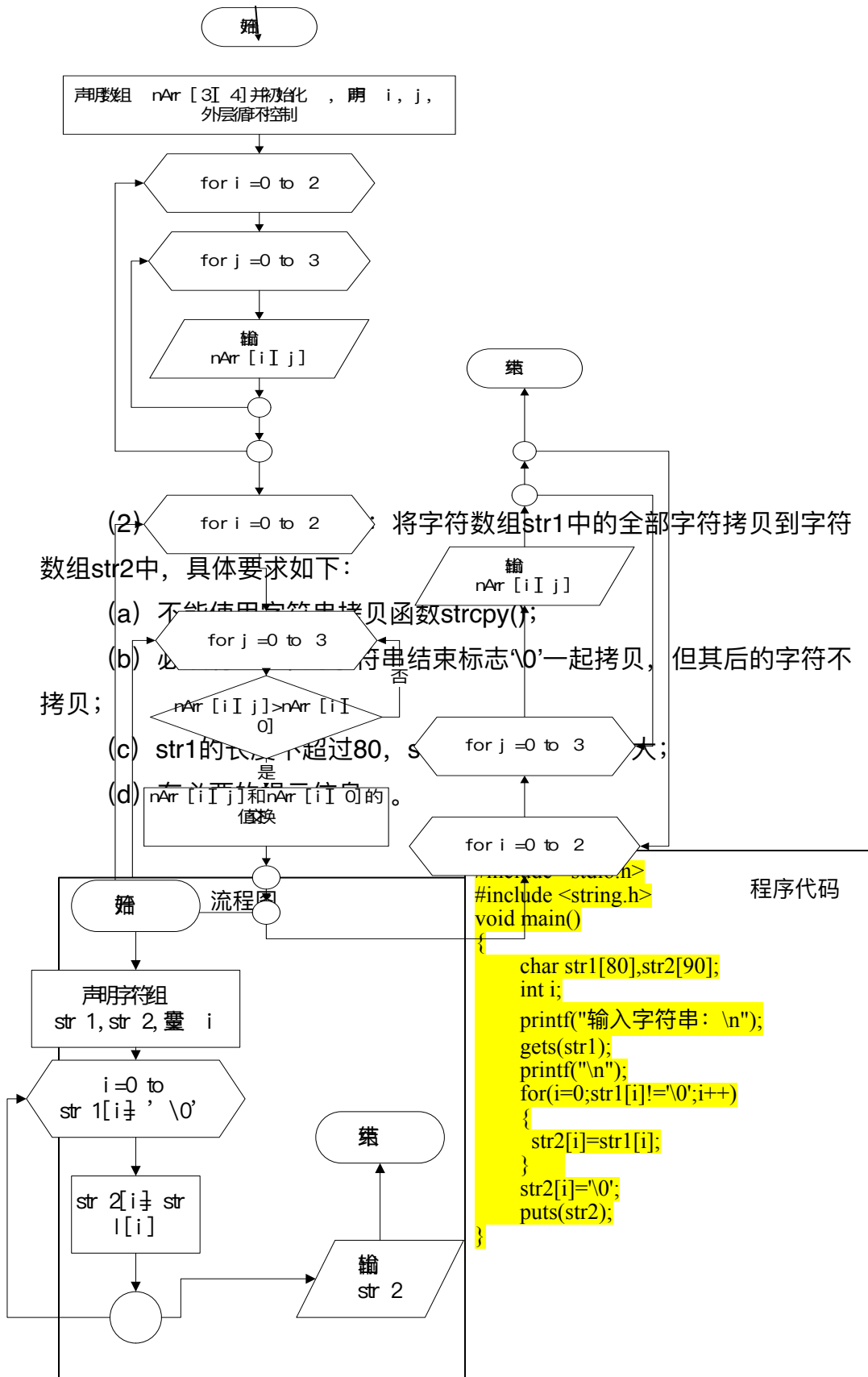
(d) 有必要的提示信息。

流程图

见下页

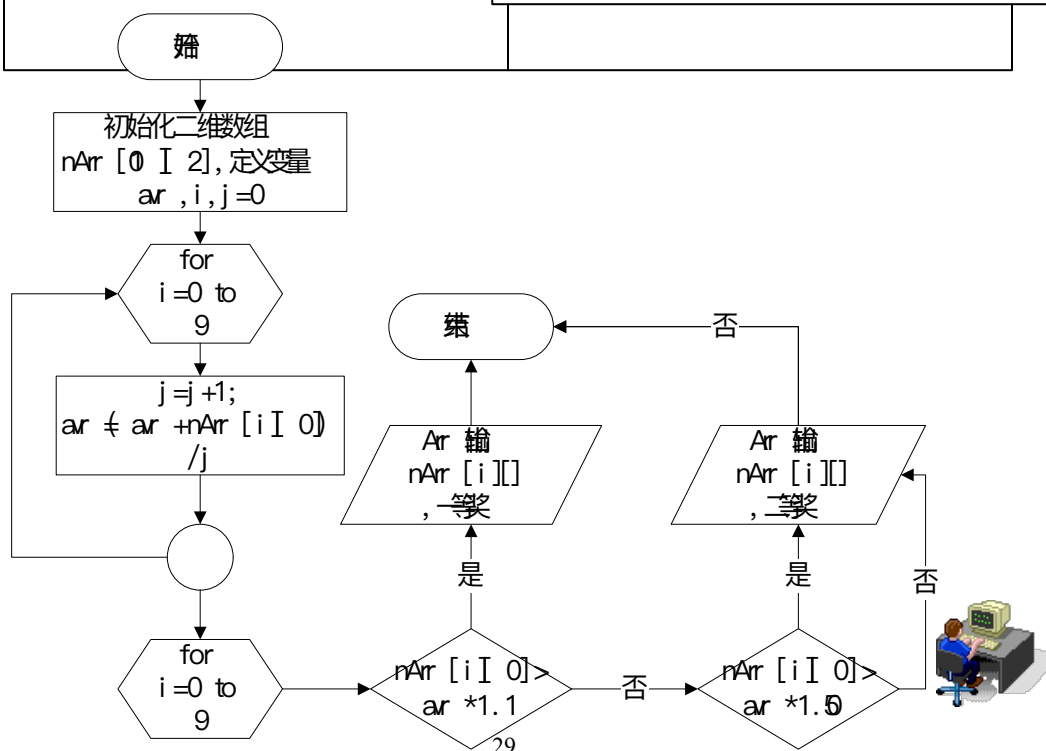
```
#include "stdio.h"
main()
{
    int s[3][4];
    int i=0,j=0,nHelp;
    printf("请按顺序输入12个数字，以空格
或回车分隔：\n");
    while (i<=2)
    {
        for (j=0;j<=3;j++)
        {scanf("%d",&s[i][j]);}
        i++;
    }
    printf("\n");
    for(i=0;i<=2;i++)
    {
        for(j=0;j<=3;j++)
        {printf(" %d",s[i][j]);}
        printf("\n");
    }
    printf("\n\n");
    for(i=0;i<3;i++)
    {
        for(j=1;j<4;j++)
        {
            if(s[i][j]>s[i][0])
            {
                nHelp=s[i][0];
                s[i][0]=s[i][j];
                s[i][j]=nHelp;
            }
        }
    }
    for(i=0;i<=2;i++)
    {
        for(j=0;j<=3;j++)
        {
            printf(" %d",s[i][j]);
        }
        printf("\n");
    }
}
```





(3) 设某班共有10名学生，为了平定某门课程的奖学金，按照规定超过全班平均成绩10%者发给一等奖，超过全班平均成绩5%者发给二等奖。编写程序，输出学生学号、成绩和奖学金等级。

流程图	程序代码
<p>见下页</p>	<pre> void Output(int nArr[][2]) { int AverS=0; int i; for(i=0;i<10;i++) AverS+=nArr[i][1]; AverS=AverS/10; printf("学号 成绩 奖学金等级\n"); for(i=0;i<10;i++) {if (nArr[i][1]>AverS*1.1) printf("%d %d 一等奖\n",nArr[i][0],nArr[i][1]); else if ((nArr[i][1]>(AverS*1.05))&&(nArr[i][1]<AverS*1.1)) printf("%d %d 二等奖\n",nArr[i][0],nArr[i][1]); } } </pre>



主函数中调用代码：

```
void main()
{
    int nArr[10][2];
    int i,j;
    printf("请按照学号、成绩的顺序依次输入数据：\n");
    for(i=0;i<10;i++)
    {for(j=0;j<2;j++)
        {scanf("%d\n",&nArr[i][j]); }}
    Output(nArr);
}
```

五、程序调试记录

(1) 调试下面程序。

```
#define N 10
main()
{ int i,num,data[]={12,15,23,29,30,31,34,45,56,70};
    /* num存放被查找的整数,数组data存放有序数列*/
    int low=0,high=N-1,mid;
    /*low、high和mid分别标记查找区间的下界和上界及中间位置*/
    printf("\nplease input num :\n");
    scanf("%d",&num); /*输入要查找的整数*/
    printf("the sorted numbers are:\n");
    for(i=1;i<N;i++)
        printf("%d ",data[i]); /*输出有序数列*/
    while(low<=high) /*使用折半法查找数据*/
    { mid=(low+high)/2;
        if(num==data[mid])
            printf("\nFind %d,it is data[%d]!",num,mid);
            break; /*若num等于数列中间位置的数据则查找成功*/
            else if(num>data[mid])
```



```

        low=mid+1;
        /*若num小于中间位置的数据, low等于mid-1*/
    else
        high=mid-1;
        /*若num大于中间位置的数据, low等于mid+1*/
    }
    if(low>high)
        printf("\n %d is not in data[]",num);
        /*若low大于high,则查找失败*/
    }

```

程序中包含有一些错误, 按下述步骤进行调试。

程序调试记录

1、程序出现的错误及改正

1.用{}把十四十五行括起来

2.第十行i=1改成i=0;

3.十九到二十三行注释有问题;

2、程序的改进

1.把七八九三行与十十一两行换一下位置, 方便用户操作;





实验5 深入模块化设计与应用

一、实验学时

完成本实验需4学时。

二、实验目的

1、阅读程序题

- (1) 学会使用函数嵌套调用的使用方法
- (2) 学会使用一维数组做为函数参数
- (3) 递归调用的设计与使用
- (4) 学会使用指针做为函数的参数

2、编程题

- (1) 如何运用指针进行程序设计
- (2) 如何以一维数组、二维数组做为函数参数进行程序设计
- (3) 冒泡排序、选择排序的灵活应用

三、实验指导

为了达到最佳的实验效果，以下提供几条适于模块化编程的指导意见，可供参考。

1、阅读程序题应先运用自己在课堂所学的知识，推导出结果，在上机时输入计算机，印证自己推导的结果，注意观察函数嵌套调用、一维数组做为函数参数、指针做为函数的参数、递归调用的使用方法。

2、编程题必须首先画出流程图，并反复思考判断程序设计的正确性，以面向过程的、模块化设计方法完成程序设计。要注意变量设置、函数参数及返回值在数据传递或共享中的重要作用。

四、实验内容

1、阅读程序题

- (1)

```
int func(int a, int b)
{ return (a+b);}
void main()
{ int x=2,y=5,z=8,r;
  r=func(func(x,y),z);
```




```
printf("%d\n",r);
}
```

该程序的输出结果是 15。

(2) int f(int b[], int n)

```
{ int i,r;
  r=1;
  for(i=0;i<=n;i++) r=r*b[i];
  return r;
}
```

void main()

```
{ int x,a[]={2,3,4,5,6,7,8,9};
  x=f(a,3);
  printf("%d\n",x);
}
```

该程序的输出结果是 120。

(3) long fib(int n)

```
{ if(n>2) return(fib(n-1)+fib(n-2));
  else return(2);
}
```

void main()

```
{ printf("%d\n",fib(3));}
```

该程序的输出结果是 4。

(4) int f(char *s)

```
{ int k=0;
  while(*s) k=k*10+*s++-'0';
  return(k);
}
```

以f("1234")方式调用该函数，结果 1234。

(5) void sort(int *b,int n)

```
{ int i,j,k,t;
  for(i=0;i<n-1;i++){
    k=i;
    for(j=i+1;j<n;j++)
```



```

        if(*(b+j)<*(b+k)) k=j;
    if(k!=i)
    { t=*(b+i);*(b+i)=*(b+k);*(b+k)=t;}
    }
}

void main()
{   int *p,i,a[10];
    p=a;
    for(i=0;i<10;i++)
        scanf("%d",p++);
    p=a;
    sort(p,10);
    for(p=a,i=0;i<10;i++,p++)
        printf("%d",*p);
    printf("\n");
}

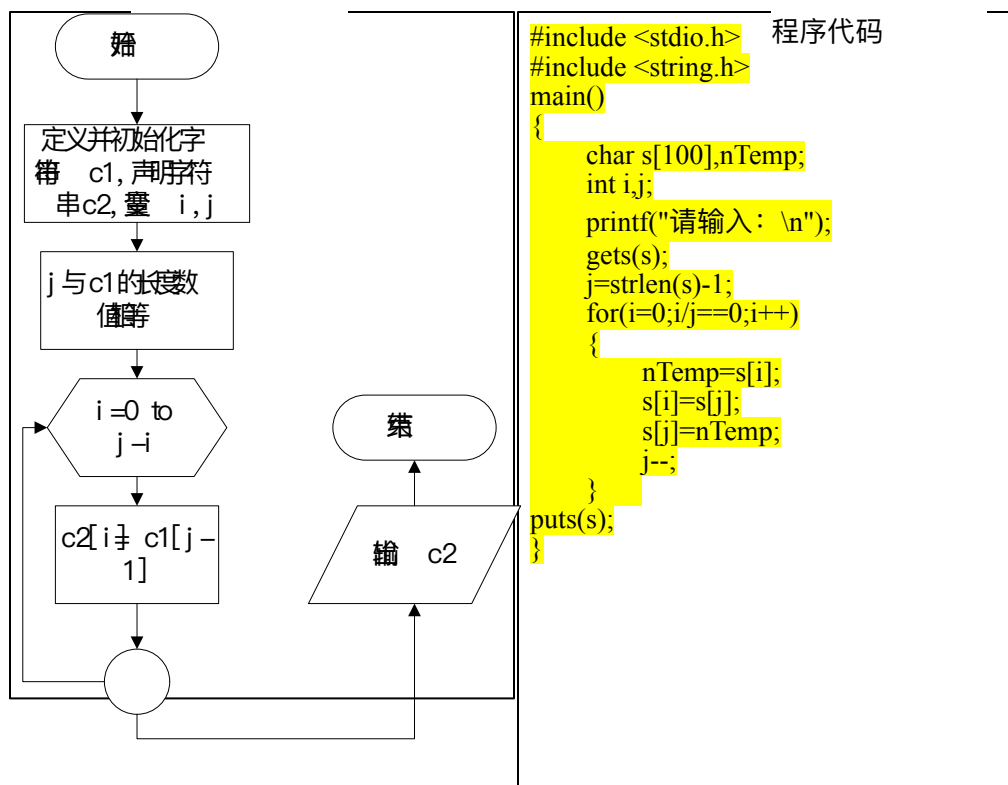
```

该程序的输出结果是 将输入数字按照从小到大顺序排列后组成一个新数字。

2、编程序并上机调试运行(包括题目及要求、流程图、程序清单、测试数据及结果)

(1) 编写一个函数fun(char *s)，函数的功能是把字符串中的内容逆置。例如，字符串中原有的内容为abcdefg，则调用该函数后，字符串中的内容为gfedcba。

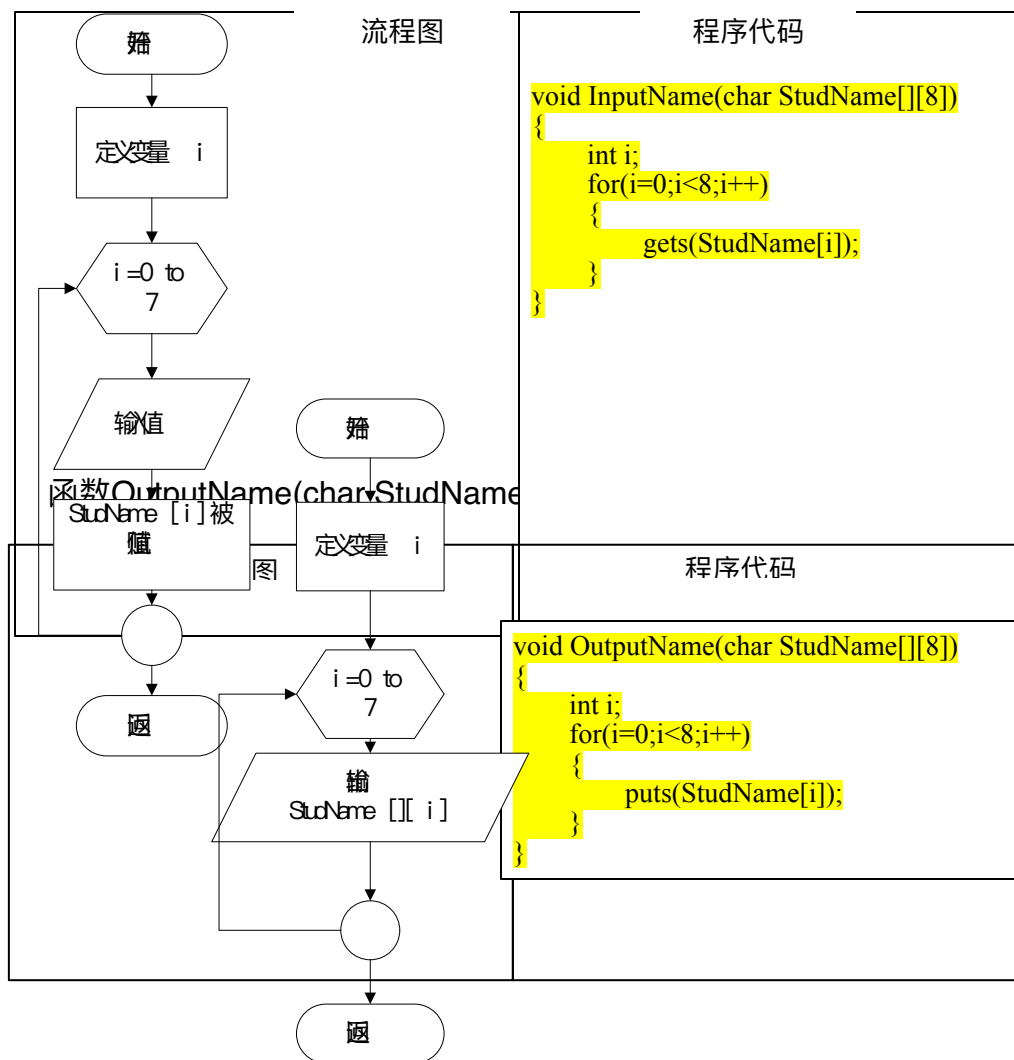




(2) 构造函数InputName(char StudName[][8]), 将8名同学的姓名存入二维数组中; 构造函数BubbleSort(char StudName[][8]), 用冒泡排序算法按学生的姓名进行排序; 构造函数OutputName(char StudName[][8]), 将排序后的结果输出。

函数InputName(char StudName[][8])





函数BubbleSort(char StudName[][8])

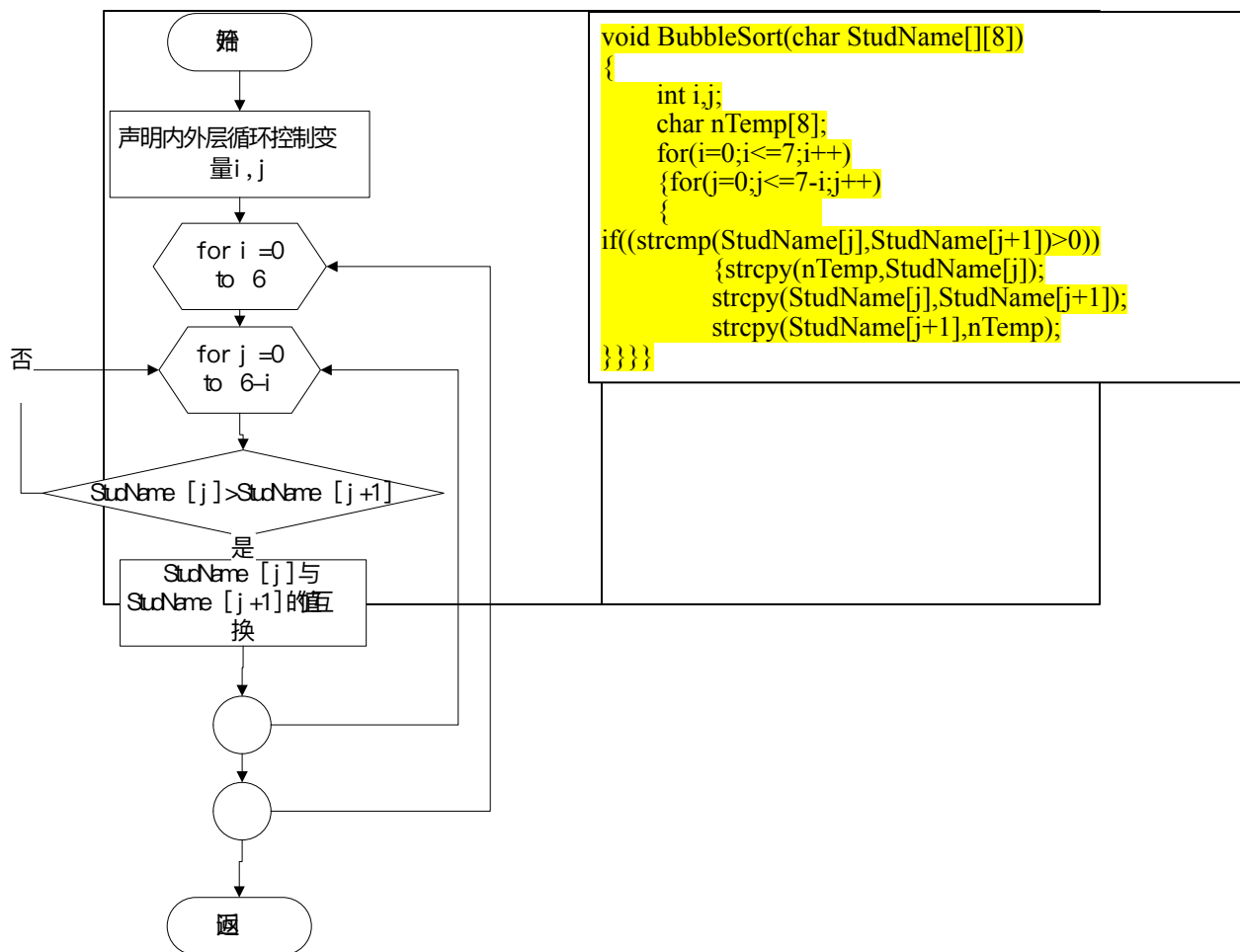
主函数中调用代码：

```

#include<stdio.h>
#include<stdlib.h>

```





```

#include<string.h>
void main()
{
    char StudName[2][8];
    InputName(StudName);
    BubbleSort(StudName);
    OutputName(StudName);}

```

测试数据:

a b d g e f c s

输出结果:

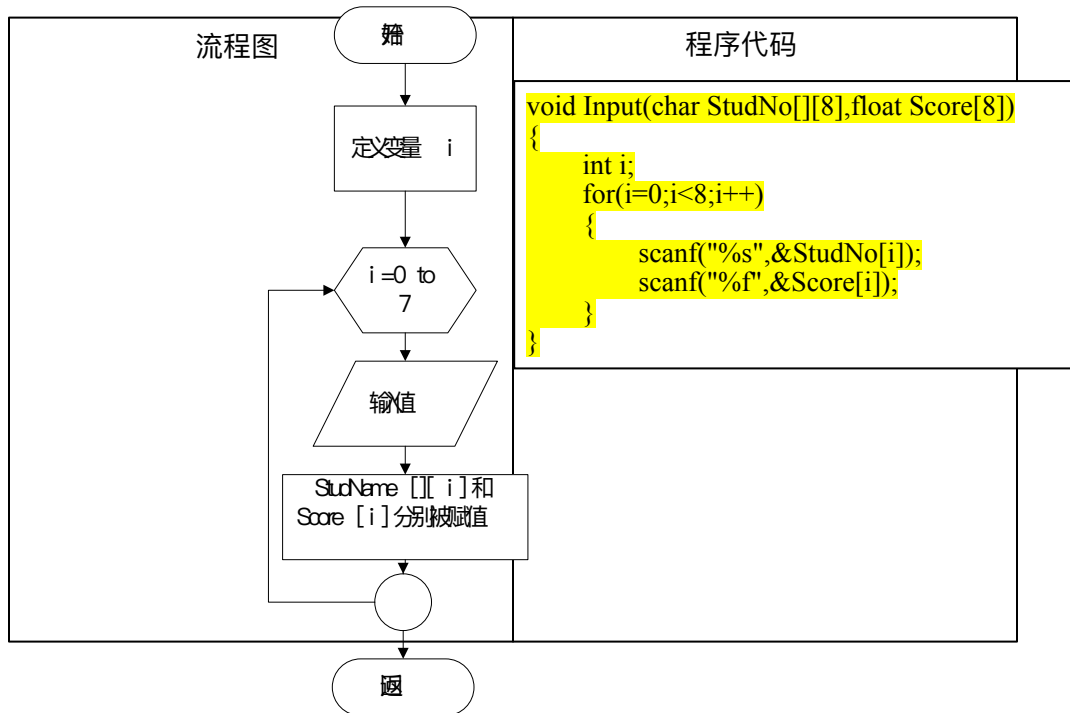
a b c d e f g s

(3) 构造函数Input(char StudNo[][8], float Score[8]), 将8名同学的姓名存入二维数组中; 构造函数SelectSort(float Score[8]), 用选择排



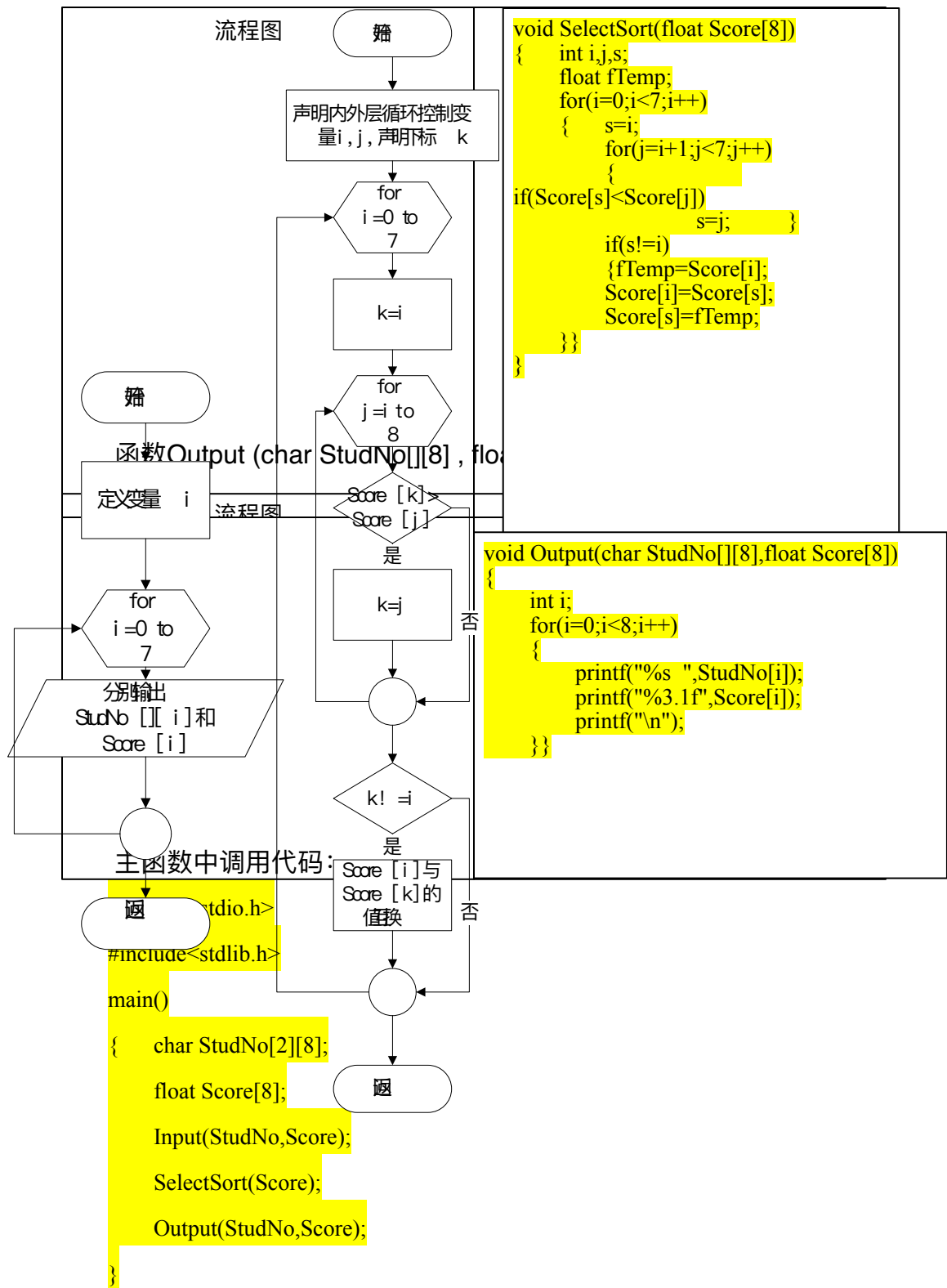
序算法按学生的成绩进行排序；构造函数Output (char StudNo[][8] , float Score[8])，将排序后的结果输出。

函数Input(char StudNo[][8], float Score[8])



函数SelectSort(float Score[8])





测试数据：

a 12 b 1 c 23 d 21 e 4 f 34 g 7 h 23

输出结果：

a 1(换行) b 4 (换行)c 7 (换行)d 12(换行) e 21(换行) f 23(换行) g 23(换行)
h 34

五、程序调试记录

要求：写出题号，并编写顺序号。如四（1）出现的错误

（1）

（2）

1、程序出现的错误及改正

2、程序的改进



实验6 构造数据类型

一、实验学时

完成本实验需3学时。

二、实验目的

1、阅读程序题

- （1）学会结构体与共用体存储空间的计算
- （2）学会结构体与共用体的嵌套定义

2、编程题

- （1）学会结构体变量的定义和引用
- （2）学会结构体数组的定义和引用



(3) 学会结构体作为函数返回值及函数参数

三、实验指导

作为一种用户自定义数据类型，必须首先进行结构体和共用体类型的定义（不分配空间），在定义了其类型之后才可以定义该结构体（共用体）类型的变量、数组或指针（分配空间）。

结构体变量定义之后，即可像简单数据类型变量一样来使用。

在实际编程中，要具体体会：

1. 结构体变量、共用体变量的成员引用方法。
2. 结构体变量、共用体变量的存储空间计算方法。
3. 结构体、共用体与指针、数组、函数的关系。

四、实验内容

1、阅读程序题

(1) typedef union

```
{
    long i;int k[5];char c;
}DATE;
struct date
{
    int cat;DATE cow;double dog;
}too;
DATE max;
main()
{
    printf("%d",sizeof(struct date)+sizeof(max));
}
```

该程序的输出结果是： 52。

(2) main()

```
{
    union bt
    {
        int k;
        char c[2];
    }a;
    a.k=-7;
    printf("%o,%o\n",a.c[0],a.c[1]);
}
```



该程序的输出结果是： 3777777777771,37777777777。

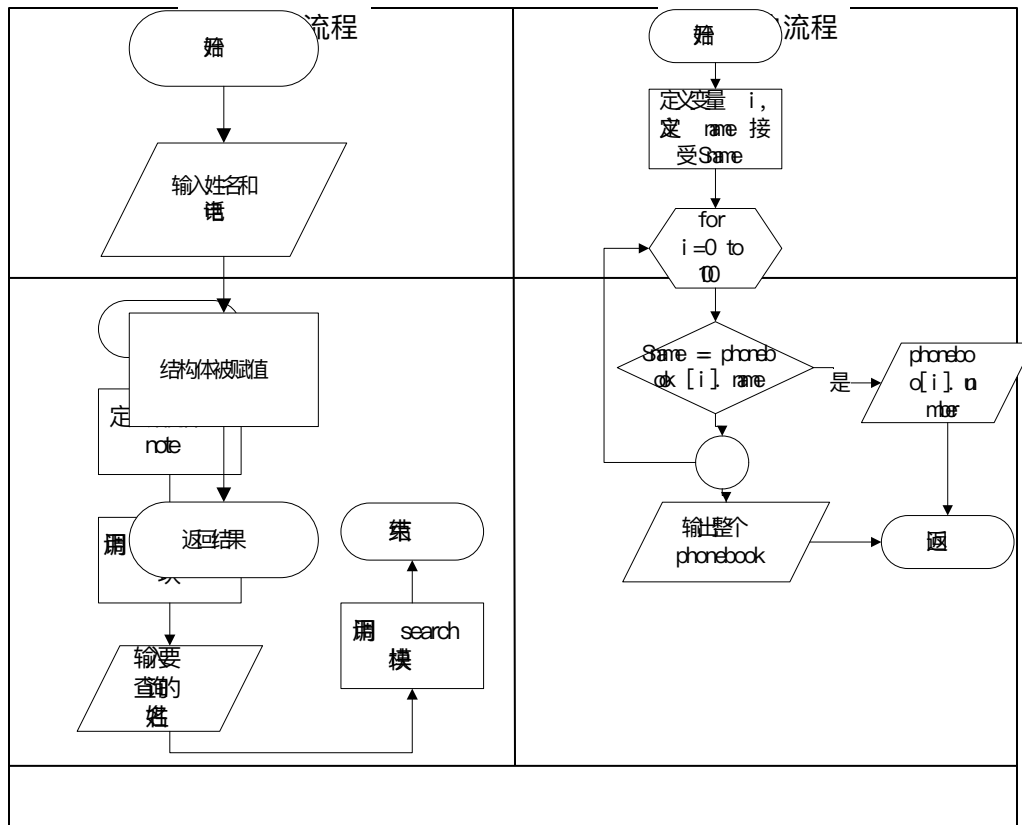
```
(3) main()
{
    union example
    {
        struct
        {
            int x;
            int y;
        } in;
        int a;
        int b;
    } e;
    e.a=1; e.b=2;
    e.in.x=e.a*e.b;
    e.in.y=e.a+e.b;
    printf("%d,%d",e.in.x, e.in.y);
}
```

该程序的输出结果是 4,8。

2、编程序并上机调试运行(包括题目及要求、流程图、程序清单、测试数据及结果)

(1) 制作一个简单的通讯录，输入姓名和电话，将其保存在结构体数组中，通过输入姓名，可查询到某个人的电话，如果姓名为空则列出所有人的姓名与电话。





主程序流程图

程序清单



<pre> #include <stdio.h> #include <stdlib.h> #include <string.h> struct data{ char name[3]; int number[7]; }*phonebook; void Input() { int i,j,s; for(i=0;i<101;i++) { for(j=0;j<7;j++) {phonebook[i].number[j]=0; phonebook[i].name[j]='\0'; }} printf("请输入通讯录中记录数：\n"); scanf("%d",&s); printf("Input the information(7位number+换行+name+#):\n"); for(i=0;i<s;i++) { for(j=0;j<7;j++) scanf("%d",&phonebook[i].number[j]); for(j=0;j<3;j++) scanf("%c",&phonebook[i].name[j]); }} int Search(struct data *x,char y[3]) { int i,find=0; for(i=1;i<101;i++) { if (strcmp(x[i].name,y)==0) find=i; } return (find); } </pre>	<pre> void main() { phonebook = (struct data *)malloc(sizeof(struct data *)*101); char Sname[3]; int nCount,j; for(j=0;j<7;j++) Sname[j]='\0'; Input(); printf("Input the name which is expected to be found:\n"); gets(Sname); nCount=Search(phonebook,Sname); if (nCount==0) { for (j=1;j<101;j++) { puts(phonebook[j].name); printf(" "); printf("%d\n",phonebook[j].number); }} else { puts(phonebook[nCount].name); printf(" "); printf("%d\n",phonebook[nCount].number); } free(phonebook); } </pre>
---	--

五、程序调试记录

要求：写出题号，并编写顺序号。如四（1）出现的错误



(1)

(2)

1、程序出现的错误及改正

2、程序的改进





实验7 综合设计与应用

一、实验学时

完成本实验需4学时。

二、实验目的

- 1、熟悉数组、指针和函数的综合编程方法；
- 2、链表的建立和使用；

三、实验指导

程序1中，用到了一维数组作为函数参数，程序2用到了指向二维数组的行指针作函数参数，程序3用到了指向二维数组的列指针作函数参数，程序4用到了动态内存分配实现动态数组。程序5涉及链表的建立、输出及节点的插入。

四、实验内容

1、编程序并上机调试运行(要求给出流程图、程序清单、测试数据及运行结果)

假设每班人数最多不超过40人，具体人数由键盘输入，试编程打印最高分及其学号。

程序1：用一维数组和指针变量作函数参数，编程打印某班一门课成绩的最高分和学号。

程序2：用二维数组和指针变量作函数参数，编程打印3个班学生（假设每班4个学生）的某门课成绩的最高分，并指出具有该最高分成绩的学生是第几个班的第几个学生。

程序3：用指向二维数组第0行第0列元素的指针作函数参数，编写一个能计算任意m行n列的二维数组中的最大值，并指出其所在的行列下标值的函数，利用该函数计算3个班学生（假设每班4个学生）的某门课成绩的最高分，并指出具有该最高分成绩的学生是第几个班的第几个学生。

程序4：编写一个能计算任意m行n列的二维数组中的最大值，并指出其所在的行列下标值的函数，利用该函数和动态内存分配方法，计算任意m个班、每班n个学生的某门课成绩的最高分，并指出具有该最高分



成绩的学生是第几个班的第几个学生。

程序5：按如下方法定义一个时钟结构体类型：

```
struct line
{ int num ;
  struct line *next;
};
```

然后，请建立一个有9个节点的链表，要求链表节点的成员num的值依次分别为1-9的整数，每建立一个节点都将之插入到原头节点前面，使新节点变成头节点，最后输出num值为偶数的节点。

程序1程序清单：

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
void Outputmax(int x[40],long int *a)
{
    int s=0,j;
    for(j=1;j<40;j++)
    {
        if(x[s]<x[j])
            s=j;
    }
    printf("最高分是： %d\n",x[s]);
    printf("该生学号： %d\n",*(a+s));
}
```

```
void main()
{
    long int number[40],*pointer;
    int i,j,Score[40];
    printf("输入班级人数 (<40) \n");
    scanf("%d",&i);
    pointer=number;
    printf("输入学号和成绩\n");
    for(j=0;j<i;j++)
    {
        scanf("%d %d\n",pointer+j,&Score[j]);
    }
    Outputmax(Score,pointer);
}
```

程序2程序清单：

```
#include<stdio.h>
int main()
{
    int i,j,score[3][4],max,*m,s=0;
    int fun(int a[],int *m);
    m=&s;
    for(i=0;i<3;i++)
    for(j=0;j<4;j++)
    scanf("%d",&score[i][j]);
    for(i=0;i<3;i++)
    {max=fun(score[i],m);
```

```
printf("%d班%d号:%d\n",i,*m+1,max);}}
int fun(int a[],int *m)
{
    int j,max;
    max=a[0];
    for(j=0;j<4;j++)
    if(max<a[j])
    { max=a[j];
      *m=j; }
    return(max);}
```

```
#include<stdio.h>
void Outputmax(int x[3][4],int *a)
{   int   j,i,r,s,m,n;
r=0;s=0;
printf("请输入要查找的班级数和每班人数: \n");
scanf("%d %d",&m,&n);
for(i=0;i<m;i++)
{for(j=0;j<n;j++)
if (*a<*(a+i*n+j))
{a=a+i*n+j;
r=i;
s=j;}}
printf("最高分是: %d\n",*(a+i*4+j));
printf("该生是%d班第%d个学生",r+1,s+1);}
```

```
void main()
{
    int nScore[3][4],*pointer,i,j;
    pointer=nScore[0];
    printf("依次输入成绩\n");
    for(i=0;i<3;i++)
        for(j=0;j<4;j++)
            scanf("%d\n",&*pointer+i*4+j);
    Outputmax(nScore,pointer);
}
```

程序4程序清单·

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int Outputmax(int *a,int p,int q)
{   int   s=0,r=0,j,i;
for(i=0;i<p;i++)
{   for(j=0;j<q;j++)
{   if (*(a+r*p+s)<*(a+i*p+j)))
{r=i;s=j;}}
}
return(r+1,s+1,*(a+r*p+s));
}
```

```
void main()
{   int i,j,m,n,Score[40][40],*pointer;
printf("依次输入成绩\n");
for(i=0;i<3;i++)
    for(j=0;j<4;j++)
        scanf("%d %d\n",&Score[i][j]);
printf("输入要查找的班级及每班人数: \n");
scanf("%d %d",&m,&n);
pointer=(int *)malloc(sizeof(int)*m*n);
printf("最高分是%d班%d位同学的分\n",Outputmax(pointer,m,n));}
```

程序5程序清单：




```

#include<stdio.h>
#include<stdlib.h>
#include<string.h>
struct line{
    int num;
    struct line *next;
}*pHead,*pNew,*pFind;
void main()
{
    int j;
    j=9;
    pHead = NULL;
    pHead=(struct line *)malloc(sizeof(struct line *));

```

```

    pHead->num=j;
    for(j=8;j>0;j--)
    {
        pNew=(struct line *)malloc(sizeof(struct line
    *));
        pNew->num=j;
        pNew->next=pHead;
        pHead=pNew;
    }
    pFind=pHead;
    while(pFind!=NULL)
    {
        if((pFind->num)%2==0)
            printf("%d",pFind->num);
    }
}

```

五、程序调试记录

1、程序出现的错误及改正

2、程序的改进





实验8 数据永久性存储用

一、实验学时

完成本实验需3学时。

二、实验目的

1、阅读程序题

学习文件指针的概念，熟悉并掌握各种基本文件操作函数，以及如何进行文件基本操作。

2、编程题

(1) 如何运用文件指针进行文件的打开、关闭操作；了解常用的文件打开模式；了解文件存储的相对路径和绝对路径的区别

(2) 了解各种文件打开方式、字符读写、字符串读写、数据块读写等不同函数之间的区别以及实现方式。

3、作为全书最后一章，本章的实验内容结合了前面几章的知识，培养学生的综合编程能力。

三、实验指导

本章内容相对固定，没有给初学者更多的灵活发挥空间，因此相对容易掌握；但同时本章涉及到的基本文件操作函数较多，而且要求能够准确运用。因此本章实验需要注意以下几点。

1、上机前要求熟悉各种文件读写函数，包括参数、返回值、调用方式以及功能，必要的时候需要一点点“死记硬背”（本课程大部分内容没有这种要求）。在上机时通过读程序和实际编程，巩固对上述函数的功能和使用方法的了解。

2、编程题的测试数据要求学生自己构建，在构建文件时要注意多尝试各种可能发生的情况，例如特殊字符；中英文字符混合；空行等等，观察程序的运行结果。

四、实验内容

1、阅读程序题

(1)

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



```

main( )
{ FILE *fp; int i,k=0,n=0;
  fp=fopen("d1.dat","w");
  for(i=1;i<4;i++) fprintf(fp,"%d",i);
  fclose(fp);
  fp=fopen("d1.dat","r");
  fscanf(fp,"%d%d",&k,&n); printf("%d %d\n",k,n);
  fclose(fp);
}

```

执行后输出结果是 123 0。

(2) 有以下程序（提示：程序中fseek(fp, -2L*sizeof(int), SEEK_END);

语句的作用是使位置指针从文件尾向前移2*sizeof(int)字节)

```

#include <stdio.h>
main( )
{ FILE *fp; int i,a[4]={1,2,3,4},b;
  fp=fopen("data.dat","wb");
  for(i=0;i<4;i++) fwrite(&a[i],sizeof(int),1,fp);
  fclose(fp);
  fp=fopen("data.dat","rb");
  fseek(fp,-2L*sizeof(int),SEEK_END);
  fread(&b,sizeof(int),1,fp);
  fclose(fp);
  printf("%d\n",b);
}

```

执行后输出结果是 3。

(3)

```

#include <stdio.h>
void WriteStr(char *fn,char *str)
{ FILE *fp;
  fp=fopen(fn,"w");fputs(str,fp);fclose(fp);
}

```



```
main()
{
    WriteStr("t1.dat","start");
    WriteStr("t1.dat","end");
}
```

程序运行后，文件t1.dat中的内容是 end。

2、编程序并上机调试运行(包括题目及要求、流程图、程序清单、测试数据及结果)

(1) 某英汉词典文件包含N个记录($N>1$)，每个记录有两个字段：一个是英文单词，另一个是相应的汉语解释。各个记录按英文单词的词典顺序排列，各英文单词并不重复。

要求编写一个用于维护、更新该英汉词典文件的函数。维护、更新的方法是：首先读入英汉词典文件，并将读入的N个英文单词和相应的汉语解释依次存放在一个结构体数组中。输入一个英文单词及其汉语解释，然后在该数组中查找输入的英文单词，若找到，则用输入的汉语解释更新原有的解释；若找不到，则需要将输入的英文单词及其汉语解释插入到该数组的适当位置，使各记录仍按英文单词的词典顺序排列。最后将数组输出，形成新的英汉词典文件，算法结束。

测试数据：

输出结果：



流程图	程序代码

(2) 将若干个原始文件合并成一个文件，然后键盘输入k值，在合并文件中恢复出其中第k个原始文件。要求采用三个函数实现：文件合并函数、文件恢复函数、主函数。

提示：合并文件中先顺序存储各原始文件，然后顺序存储各原始文件的控制信息，即文件名、文件长度和在合并文件中的位置(偏移量)。其结构为：

```
typedef stmct
{ char file_name[256]; /*原始文件名*/
  long length; /*原始文件长度(字节数)*/
  long offset; /*原始文件在合并文件中的位置(偏移量)*/
```



}FileInfo;

最后通过控制信息恢复出原始文件。

```
#include "string.h"
#include "stdio.h"
#include "stdlib.h"
struct FileInfo{
char filename[256];
long length;
long offset;
char* content;};
#define MaxFile 10
int combine()
{static char buffer[1024];
static struct FileInfo infos[MaxFile];
int number=0;
int c;
int error=0;
char* p;
p=buffer;
printf("input filenames separate
with space:\n");
do{c=getc(stdin);
if(' '==c || '\n'==c){
*p=0;FILE* handle=fopen(buffer, "r");
if(! handle ){
printf("can't open:%s\n",buffer);
return 0;}
FileInfo* curlInfo=&infos[number];
strcpy ( curlInfo->filename,buffer);
errrororl
=fseek(handle,0,SEEK_END);
curlInfo->length=ftell(handle);
errrororl
=fseek(handle,0,SEEK_SET);
curlInfo->content=(char*)malloc(curlInfo->length);
fread ( curlInfo->content, curlInfo->length,1,handle);
fclose(handle);
if(error){
printf("error op file:
%s\n",buffer);
return 0;}
number++;
p=buffer;}
else * (p++)=c;
}while(c!='\n');
FILE* fout=fopen("out","w");
fwrite(&number,sizeof(number),1,fout);
int offset=sizeof(number)+number*sizeof(FileInfo);
for(int i=0;i<number;i++){
infos[i].offset=offset;
offset+=infos[i].length;}
fwrite(infos,sizeof(FileInfo),number,fout);
for(int i=0;i<number;i++){
fwrite(infos[i].content,infos[i].length,1,fout);}
fclose(fout);
return 1;}
int recoverK(const char* fname,int k){
static FileInfo infos[MaxFile];
FILE* handle=fopen(fname,"rb");
int number=0;
fread(&number,sizeof(number),1,handle);
if(k>number || k<=0){
printf("bad k\n");
return 0;}
fread(infos,sizeof(FileInfo),number,handle);
FileInfo* kinfo=&infos[k-1];
void* buffer=malloc(kinfo->length);
fseek (handle,kinfo->offset,SEEK_SET);
fread (buffer,kinfo->length,1,handle);
FILE* khandle=fopen(kinfo->filename,"w");
fwrite (buffer,kinfo->length,1,khandle);
fclose(khandle);fclose(handle);
return 1;
}
int _tmain(int argc, _TCHAR* argv[])
{
combine();
printf("input k:");
int k;scanf("%d",&k);
recoverK("out",k);
return 0;
}
```

五、程序调试记录

要求：写出题号，并编写顺序号。如四（1）出现的错误

（1）

（2）

1、程序出现的错误及改正

2、程序的改进

