C语言程序设计实验报告

学号： 姓名： 魏弘博 专业班级：

**实验十一 数组**

【实验目的】

1. 掌握一维数组的概念及定义、初始化、赋值和输入/输出的方法。
2. 掌握二维数组的概念及定义、初始化、赋值和输入/输出的方法。
3. 理解数组的概念和存储特点。

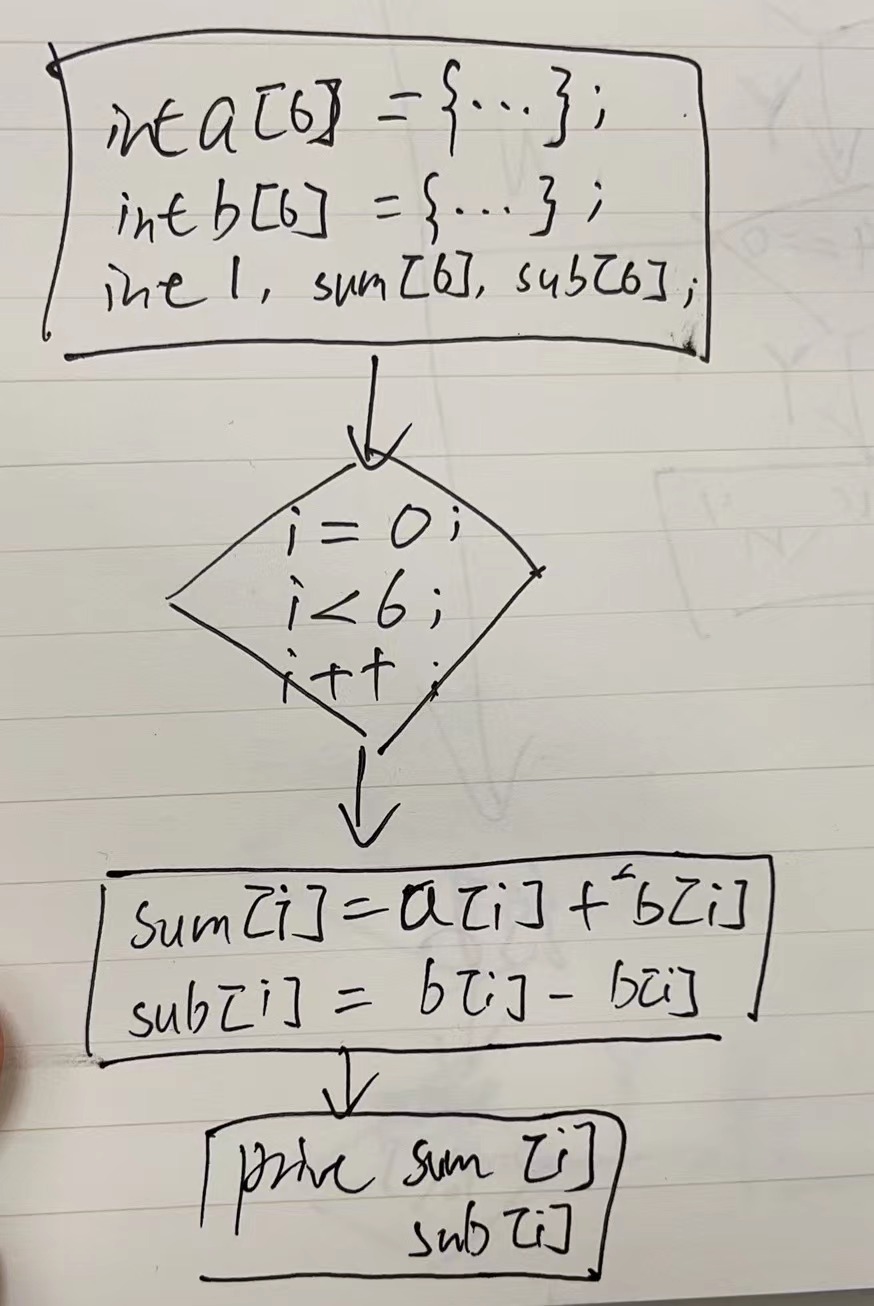
【实验内容】

一、基础编程题

1、编写程序，求矩阵a和b相加减,并输出结果(矩阵a、b均为一维矩阵)

a= 1 2 3 4 5 6 , b= 1 2 3 4 5 6

（1）程序分析设计思路（使用标准流程图符号完成流程图）



（2）源代码（不能截图，只能复制粘贴）

#include "stdio.h"

int main()

{

int a[6]={1,2,3,4,5,6},b[6]={1,2,3,4,5,6},i,sum[6],sub[6];

for(i=0;i<6;i++)

{

sum[i]=a[i]+b[i];

printf("%d ",sum[i]);}

printf("\n");

for(i=0;i<6;i++)

{

sub[i]=a[i]-b[i];

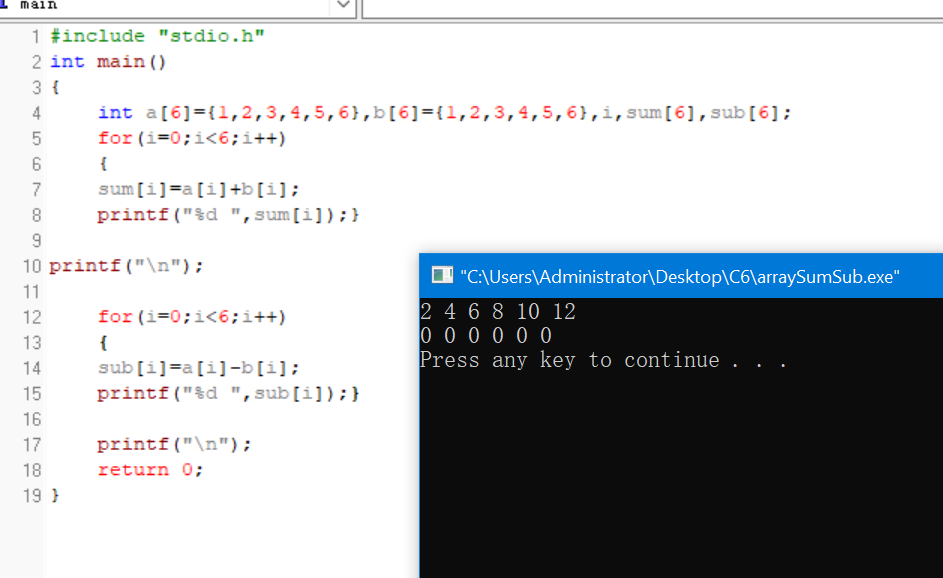
printf("%d ",sub[i]);}

printf("\n");

return 0;

}

}（3）运行结果截图（只截取相应运行结果截图）



2、编写程序，设计一个函数，完成如下功能：

自定义一个10位数组，并使用选择排序法进行排序，选择排序如下图：

初始态 { 54 45 36 81 72 18 63 27 }

第一趟 18 { 45 36 81 72 54 63 27 }

第二趟 18 27 { 36 81 72 54 63 45 }

第三趟 18 27 36 { 81 72 54 63 45 }

第四趟 18 27 36 45 { 72 54 63 81 }

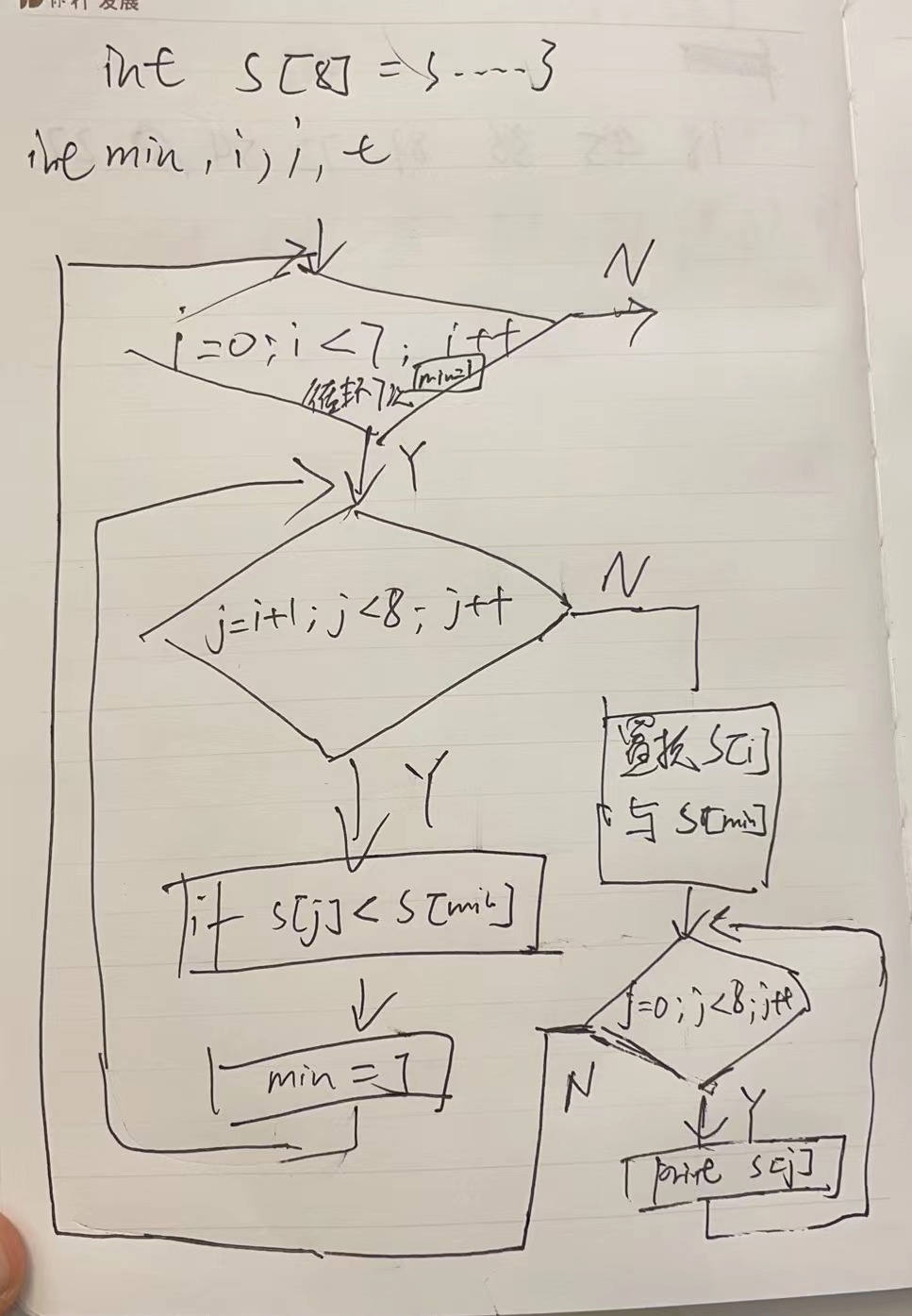
第五趟 18 27 36 45 54 { 72 63 81 }

第六趟 18 27 36 45 54 63 { 72 81 }

第七趟 18 27 36 45 54 63 72 { 81 }

选择排序（Selection sort）是一种简单直观的排序算法。它的工作原理是：第一次从待排序的数据元素中选出最小（或最大）的一个元素，存放在序列的起始位置，然后再从剩余的未排序元素中寻找到最小（大）元素，然后放到已排序的序列的末尾。以此类推，直到全部待排序的数据元素的个数为零。

（1）程序分析设计思路（使用标准流程图符号完成流程图）



（2）源代码（不能截图，只能复制粘贴）

#include"stdio.h"

int main()

{

int s[8]={54,45,36,81,72,18,63,27};

int min,i,j,t;

for(i=0;i<7;i++)

{

min=i;

for(j=i+1;j<8;j++)if(s[j]<s[min]) min=j;

t=s[i];

s[i]=s[min];

s[min]=t;

for(j=0;j<8;j++)printf("%d ",s[j]);

printf("\n");

}

printf("\nFinal result is:\n");

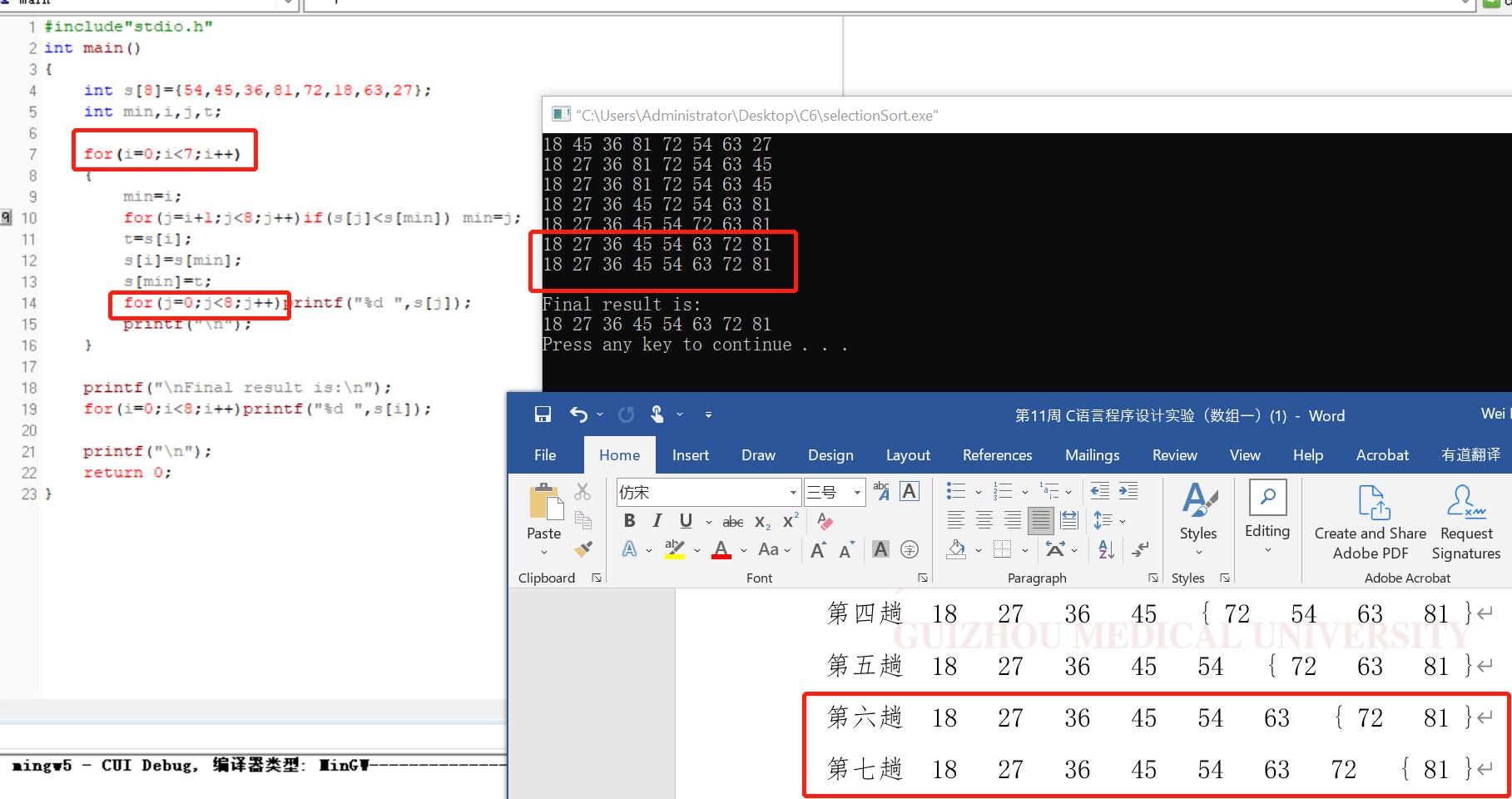
for(i=0;i<8;i++)printf("%d ",s[i]);

printf("\n");

return 0;

}

（3）运行结果截图（只截取相应运行结果截图）



3、编写程序，设计一个函数，完成功能：把某年某月的第几天转换成该年的第几天。

（提示：首先判断所输入的年份是否为闰年。平年2月28天，闰年2月29天。 该年的第几天 = 该年该月之前的各月份天数和 + 该年该月的天数。）

设：

yearday: 存放该年的第几天变量。

year: 存放年变量。

month: 存放月变量。

day: 存放天变量。

month\_tab[ ][ ]: 存放闰年与非闰年各月的天数。

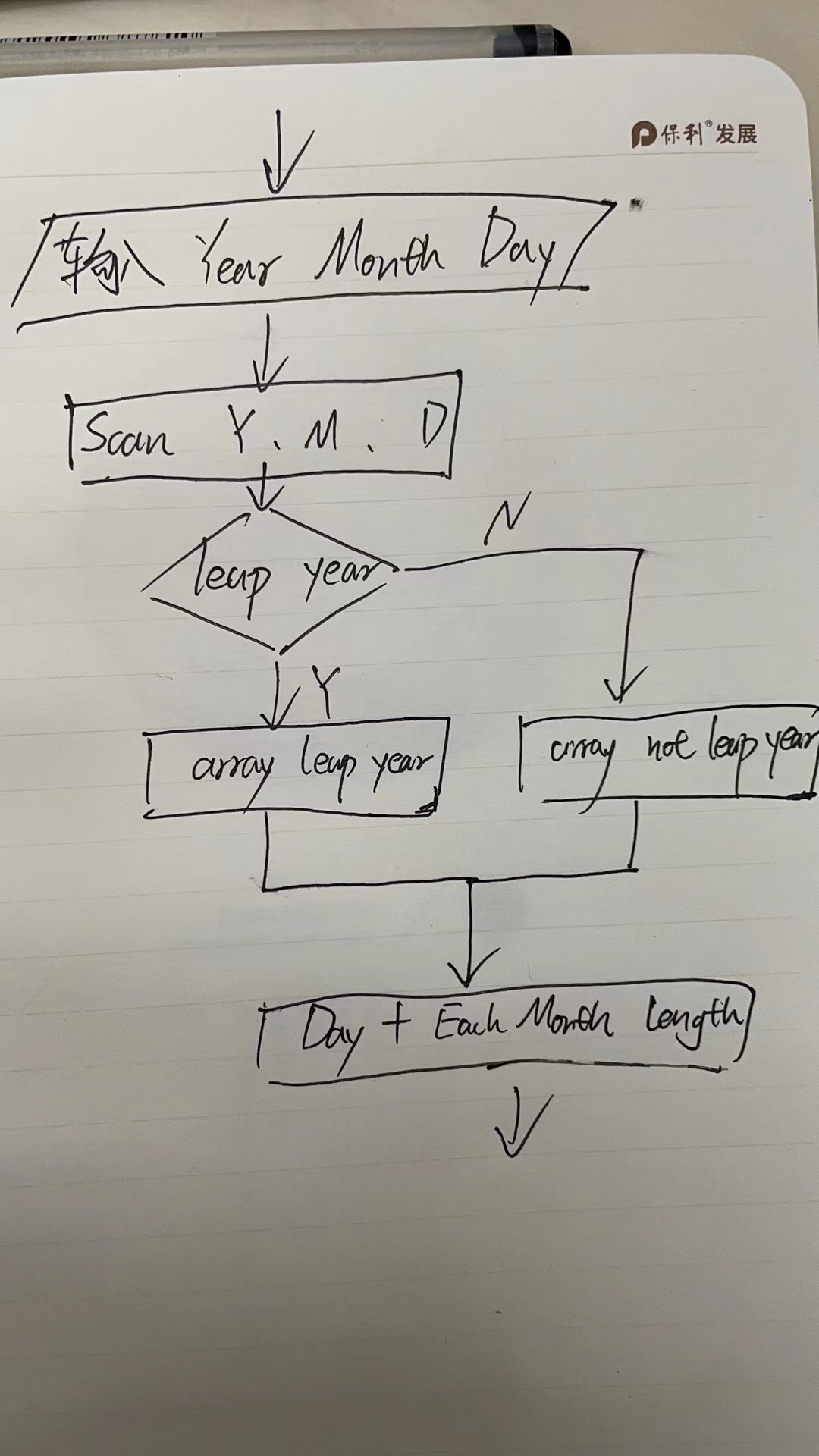
month\_tab[0]: 存放非闰年各月的天数。

month\_tab[1]: 存放闰年各月的天数。

leap: 闰年变量。

static int month\_tab[ 2 ] [ 13 ]= { {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}, {0, 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31} };

1. 程序分析设计思路（使用标准流程图符号完成流程图）



（2）源代码（不能截图，只能复制粘贴）

#include "stdio.h"

int main()

{

static int month\_tab[2][13]=

{ {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31},

{0, 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31} };

int i,year,month,day,sumDay=0;

printf("Enter Year Month Day with number in form Y M D: ");

scanf("%d %d %d",&year,&month,&day);

if(year%4==0&&year%100!=0||year%400==0)

i=1;

else

i=0;

for(month;month>0;month--)

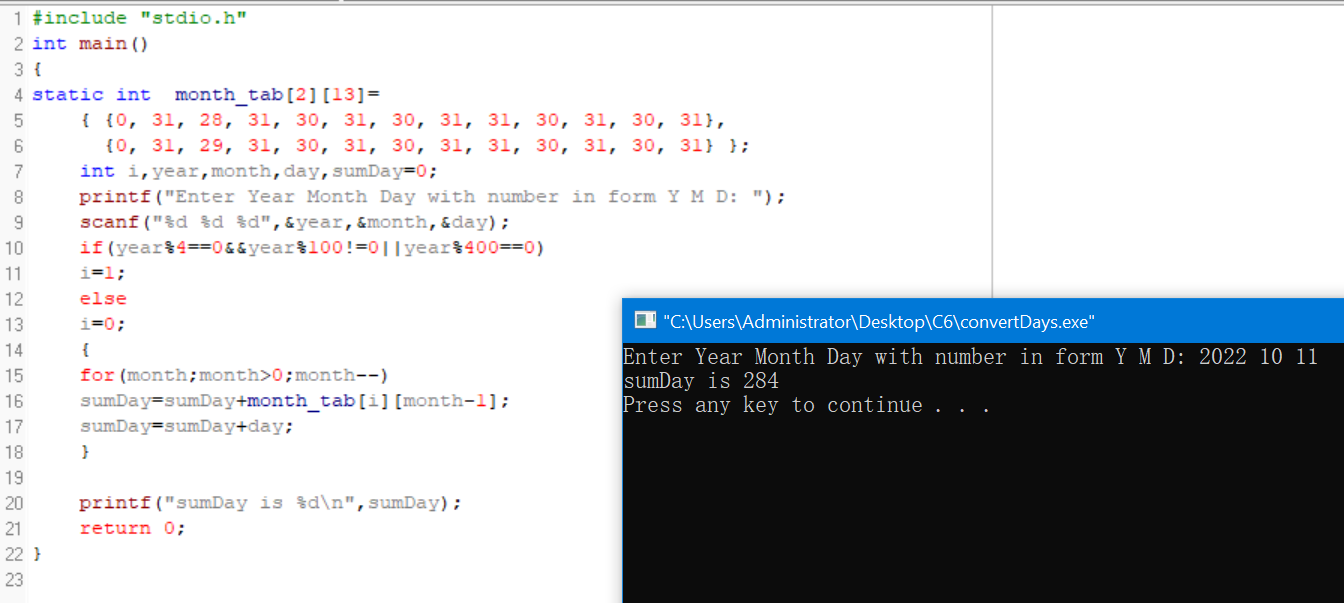
sumDay=sumDay+month\_tab[i][month-1];

sumDay=sumDay+day;

printf("sumDay is %d\n",sumDay);

return 0;

}}（3）运行结果截图（只截取相应运行结果截图）



二：拓展编程题

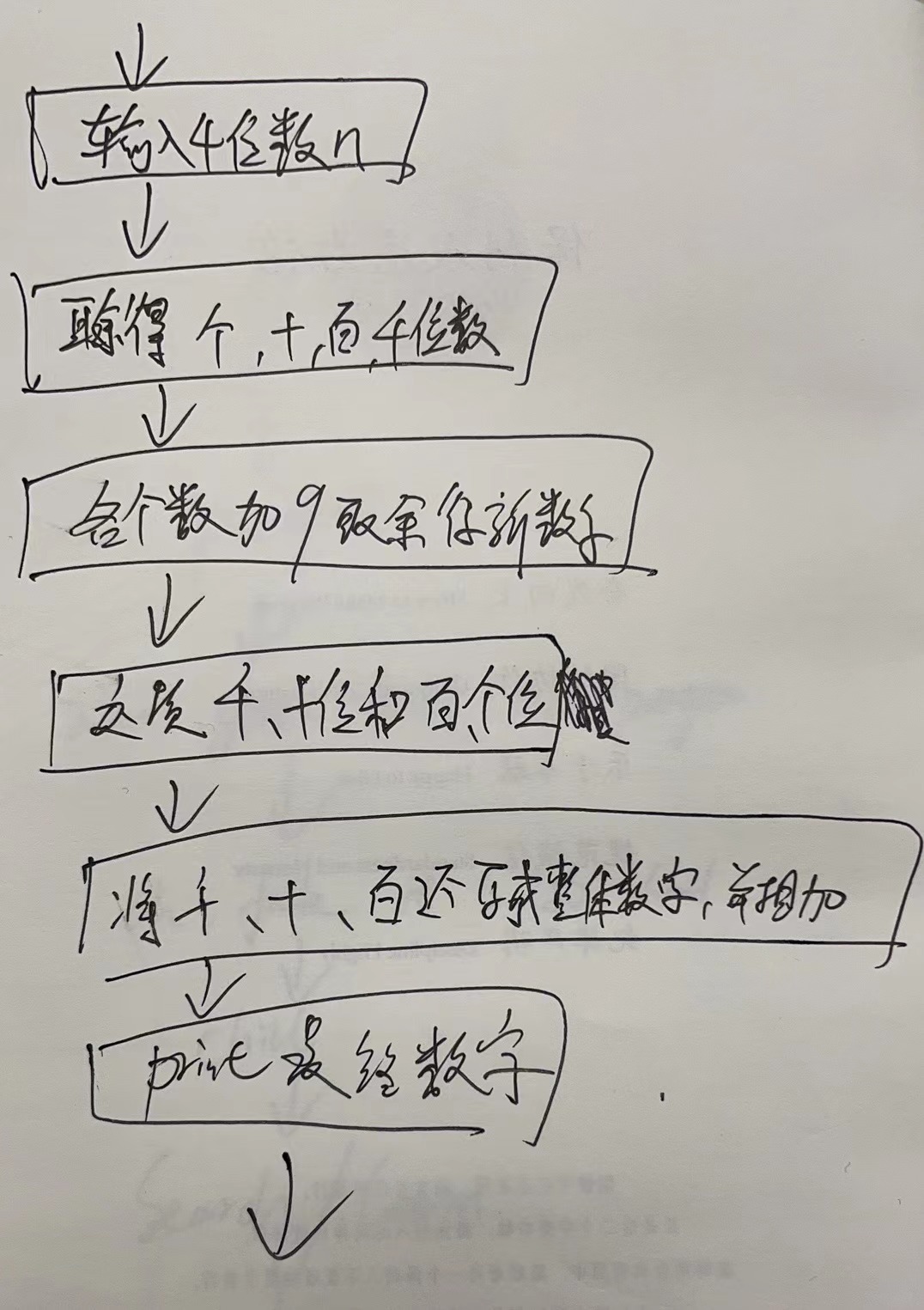
1、数字加密：输入一个四位数，将该数每一位上的数字加9，然后除以10取余，作为该位上的新数字，最后将千位和十位上的数字互换，百位和个位上的数字互换，组成加密后的新四位数。例如输入1257，经过加9取余后得到的新数字0146，在经过两次换位后得到4601。试编写相应程序。

输入输出示例

1257

4601

（1）程序分析设计（使用标准流程图符号完成流程图）



（2）源代码（不能截图，只能复制粘贴）

#include "stdio.h"

int main()

{

int x,n,num[4],i,t;

printf("Enter 4 digits number: ");

scanf("%d",&x);

for(i=3;i>=0;i--)

{

n=x%10;

x=(x-n)/10;

num[i]=n;

num[i]=(num[i]+9)%10;

}

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

{

t=num[i];

num[i]=num[i+2];

num[i+2]=t;

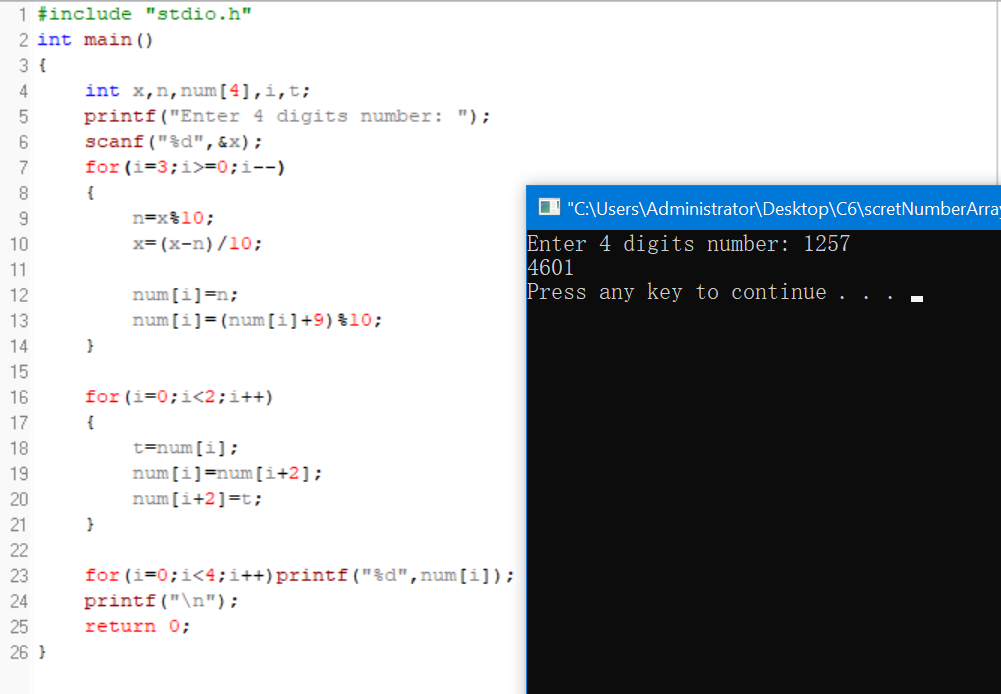
}

for(i=0;i<4;i++)printf("%d",num[i]);

printf("\n");

return 0;

}（3）运行结果截图（只截取相应运行结果截图）



【总结报告】

（分析本次实验程序设计思路、运行情况及存在的问题，包括本次实验所取得的经验，若编程过程中出现错误，应分析错误原因）