1. m个人的成绩存放在数组score中，编写函数fun，它的功能是：
2. 求得低于平均分的人数
3. 求得低于平均分的分数
4. 将上述两种值带回主函数

#include <stdio.h>

double getEqual(int \* score)

{

    int sum = 0;

    for(int i = 0;i<5;i++)

    {

        sum+=\*(score+i);

    }

    double equal = sum/5.0;

    return equal;

}

void func(int \* score,double equal)

{

     printf("低于平均分的分数有:");

     int number = 0;

     for(int i =0;i<5;i++)

    {

        if(\*(score+i) < equal)

        {

            printf("%d,",\*(score+i));

           number+=1;

        }

    }

     printf("\n低于平均分的人数为:%d\n",number);

}

int main(){

    int score[5]={85,66,87,57,61};

    int equal = getEqual(score);

    func(score,equal);

    return 0;

}

1. 请编写函数void fun(int x, int pp[], int \*n)，他的功能是：
2. 求出能整除x但不是偶数的各整数
3. 按从小到大的顺序存放在pp所指的数组中
4. 这些除数的个数通过形参n返回

#include <stdio.h>

#define NUM 6

void fun(int x, int pp[], int \*n)

{

    int newPP[NUM]={0};

    int temp = 0;

    int j =0;

     printf("能整除x的数有:");

    for(int i =0;i<NUM;i++)

    {

        if(pp[i]%x==0)

        {

            printf("%d,",pp[i]);

            j++;

        }

    }

    \*n=j;

    for(int i =0;i<NUM;i++)

    {

        for(j =i+1;j<NUM;j++)

        {

            if(pp[i]>pp[j])

            {

                temp=pp[i];

                pp[i]=pp[j];

                pp[j]=temp;

            }

        }

    }

}

int main(){

    int pp[NUM]={3,67,45,69,34,65};

    int x = 3;

    int \* n =&x;

    fun(x,pp,n);

    printf("\n有%d个数\n",\*n);

    printf("排序后为:");

    for(int i =0;i<NUM;i++)

    {

        printf("%d,",pp[i]);

    }

    return 0;

}

1. 写一个函数，输入一个四位数字，要求输出这四个数字字符，但每两个数字间空一个空格，如输入2021，输出’2 0 2 1’

#include <stdio.h>

#include<math.h>

void fun(int num)

{

    char str[8]="\0";

    for(int i = 0;i<4;i++)

    {

       int temp =pow(10,3-i);

       str[2\*i]=num/temp+'0';

       str[2\*i+1]=' ';

       num=num-(num/temp)\*temp;//初始化下一次的num

    }

    str[7]='\0';

    printf("\n%s\n",str);

}

int main(){

    fun(2021);

    return 0;

}

1. 要求使用指针，将一个3\*3的整形矩阵转置

#include"stdio.h"

#include<Windows.h>

void trans(int (\*s)[3], int x, int y);        //定义转置函数

int main()

{

    int a[3][3], \*p, i, j;

    printf("请输入: ");

    for (int i =0;i<3; i++)

    {

        for(int j =0;j<3;j++)

        {

            int input= 0;

            scanf("%d",&input);

            a[i][j]=input;

        }

    };    //输入矩阵元素

    printf("转置前:");

    for (i=0; i<3; i++)

    {

        printf("\n");                                     //输出矩阵

        for (j=0; j<3; j++)

        printf("%d ", \*(\*(a+i)+j));

    }

    trans(a, 3, 3);

    printf("\n转置后:");                                                         //调用转置函数

    for (i=0; i<3; i++)                                        //输出矩阵

    {

        printf("\n");                                     //输出矩阵

        for (j=0; j<3; j++)

        printf("%d ", \*(\*(a+i)+j));

    }

    system("pause");

return 0;

}

//转置函数

void trans(int (\*s)[3], int x, int y)

{

    int i, j, t;

    for (i=0; i<x; i++)

        for (j=0; j<y; j++)

        {

            if(i!=j&&i<j)

            {

                t=\*(\*(s+i)+j);

                 \*(\*(s+i)+j)=\*(\*(s+j)+i);

                 \*(\*(s+j)+i)=t;

            }

        }

}

1. 定义一个结构体变量，包括年月日，计算某日在某年中是第几天，注意闰年问题

#include "stdio.h"

#include <Windows.h>

typedef struct Date

{

    int year;

    int month;

    int day;

} Date;

int numberOfday(Date date)

{

    if (date.month <= 2)

    {

        return date.day + (date.month - 1) \* 31;

    }

    int num = 0;

    if (date.month % 2 == 0) //偶数月

    {

        num = (date.month / 2 - 1) \* 30 + (date.month / 2) \* 31;

    }

    else //奇数月

    {

        num = (date.month / 2 - 1) \* 30 + (date.month / 2 + 1) \* 31;

    }

    //判断是否是闰年

    if (date.year / 400 == 0 || (date.year / 4 == 0 && date.year / 100 != 0))

    {

        num += 29;

    }

    else

    {

        num += 28;

    }

    num += date.day;

    return num;

}

int main()

{

    Date date;

    printf("请输入年月日");

    scanf("%d", &date.year);

    scanf("%d", &date.month);

    scanf("%d", &date.day);

    int num = numberOfday(date);

    printf("%d\n", num);

    system("pause");

    return 0;

}

1. 有一磁盘文件“employee”，内存放职工的数据，每个职工数据包括职工姓名，职工号，性别，年龄，住址，工资，健康状况，文化程度。今要求将职工名，工资的信息单独抽出来另建一个简明的职工工资文件。

#include "stdio.h"

#include <Windows.h>

int main()

{

    char name[20]={""};

    char sex[20]={""};

    int zgh=0;

    int wage=0;

    FILE \*fp;

    FILE \*mfp;

    if((fp=fopen("C:\\Users\\czh\\Desktop\\SFLesson\\lesson1\\lessonWork\\code\\employee.txt","r")) !=NULL

    &&(mfp=fopen("C:\\Users\\czh\\Desktop\\SFLesson\\lesson1\\lessonWork\\code\\data.txt","w"))!=NULL)

    {

        while(!feof(fp))

        {

          fscanf(fp,"zgname:%s zgh:%d sex:%s wage:%d\n",name,&zgh,sex,&wage);

          fprintf(mfp,"zgname:%s wage:%d\n",name,wage);

        }

    }

    fclose(fp);

    fclose(mfp);

    system("pause");

    return 0;

}