
百题魔鬼训练营详解（基础篇）

资料属个人整理，有些题次还未完成；代码仅供参考；

如要获取完整版，老秦微信：13467535874

1. 输入输出和基本计算

1.1 将一个四位数，反顺序输出

题目描述：

将一个四位数，反顺序输出

输入描述：

一行，输入一个整数 n ($1000 \leq n \leq 9999$)

输出描述：

针对每组的输入，反向输出对应的四位数

示例：

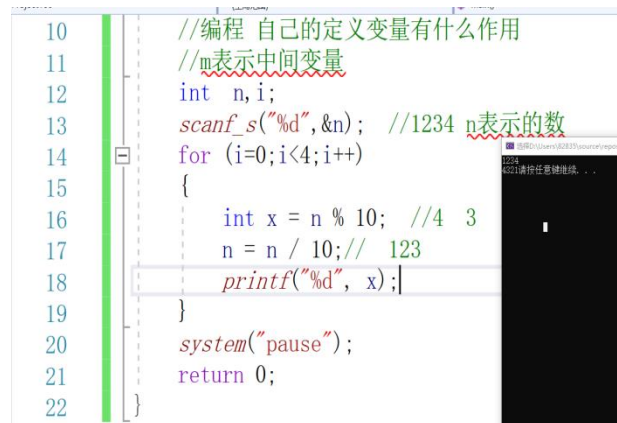
输入：

2345

输出：

5432

运行截图：



```
10 //编程 自己的定义变量有什么作用
11 //m表示中间变量
12 int n, i;
13 scanf_s("%d", &n); //1234 n表示的数
14 for (i=0; i<4; i++)
15 {
16     int x = n % 10; //4 3
17     n = n / 10; // 123
18     printf("%d", x);
19 }
20 system("pause");
21 return 0;
22 }
```

代码：

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

int main()
{
    int n, i;
    scanf_s("%d", &n); //1234 n表示的数
    for (i=0; i<4; i++)
    {
        int x = n % 10; //4 3
        n = n / 10; // 123
        printf("%d", x);
    }
}
```

```
        system("pause");

    return 0;

}
```

1.2 计算 BMI 指数

题目描述：

问题：计算 BMI 指数（身体质量指数）。BMI 指数（即身体质量指数，简称体质指数又称体重，英文为 *Body Mass Index*，简称 *BMI*），是用体重公斤数除以身高米数平方得出的数字，是目前国际上常用的衡量人体胖瘦程度以及是否健康的一个标准。主要用于统计用途，当我们需要比较及分析一个人的体重对于不同高度的人所带来的健康影响时，BMI 值是一个中立而可靠的指标。

输入描述：

一行，两个整数，分别表示体重（公斤），身高（厘米），中间用一个空格分隔。

输出描述：

一行，BMI 指数（保留两位小数）。

示例：

输入：

70 170

输出:

24.22

运行截图:

代码:

```
#include <stdio.h>

#include <stdlib.h>

int main()
{
    //体重 身高
    float h, m;

    scanf_s("%f%f", &h, &m);

    m = m / 100; //厘米到米的转化
    printf("%.2f", h / (m * m));

    system("pause");

    return 0;
}
```

1.3 大小写字母转换

题目描述：

实现字母的大小写转换。多组输入输出。

输入描述：

多组输入，每一行输入大写字母。

输出描述：

针对每组输入输出对应的小写字母

示例：

输入：

A

B

输出：

a

b

运行截图：

代码：

1.4 计算球的体积

题目描述：

给定一个球体的半径，计算其体积。其中球体体积公式为 $V = 4/3 * \pi r^3$ ，其中 $\pi = 3.1415926$ 。

输入描述：

一行，用浮点数表示的球体的半径。

输出描述：

一行，球体的体积，小数点后保留 3 位。

示例：

输入：

3.0

输出：

113.097

运行截图：

代码：

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

#define PI 3.1415926

int main()
{
    // V = 4/3 *  $\pi$  r3，其中  $\pi = 3.1415926$ 。
    // 应该定义几个变量？
```

```
double r, v;

scanf_s("%lf", &r);

v = (4 / 3.0) * PI * r * r * r;

printf("%.3lf\n", v);

system("pause");

return 0;

}
```

1.5 字符金字塔

题目描述：

输入一个字符，用它构造一个三角形金字塔。

输入描述：

输入只有一行，一个字符。

输出描述：

该字符构成的三角形金字塔。

示例：

输入：

1

输出:

```
1
1 1
1 1 1
1 1 1 1
1 1 1 1 1
```

运行截图:

代码:

1.6 总成绩和平均成绩

题目描述:

依次输入一个学生的 3 科成绩，在屏幕上输出该学生的总成绩以及平均成绩。

输入描述:

```
一行，3 科成绩，成绩之间用一个空格隔开。
```

输出描述:

```
一行，总成绩和平均成绩（小数点后保留两位），用一个空格隔开
```

示例:

输入:

```
79.5 80.0 98.0
```

输出:

```
257.50 85.83
```

运行截图：

代码：

1.7 进制计算

题目描述：

输入一个十六进制数 a ，和一个八进制数 b ，输出 $a+b$ 的十进制结果

输入描述：

一个十六进制数 a ，和一个八进制数 b ，中间间隔一个空格。

输出描述：

$a+b$ 的十进制结果。

示例：

输入：

0x12 05

输出：

23

运行截图：

代码：

1.8 移位计算 2 的 n 次方的计算

题目描述：

不使用累计乘法的基础上，通过移位运算（<<）实现 2 的 n 次方的计算。

输入描述：

多组输入，每一行输入整数 n ($0 \leq n < 31$)。

输出描述：

针对每组输入输出对应的 2 的 n 次方的结果。

示例：

输入：

2

10

输出：

4

1024

运行截图：

代码：

```
#include <stdio.h>
int main()
{
    int a,n;
    while(scanf("%d",&n)!=EOF){
        a=1;
        a<=n;
        printf("%d\n",a);
    }
}
```

1.9 计算三角形的周长和面积

题目描述：

根据给出的三角形 3 条边 a, b, c ($0 < a, b, c < 100,000$)，计算三角形的周长和面积。

输入描述：

三角形 3 条边（能构成三角形），中间用一个空格隔开。

输出描述:

三角形周长和面积（保留两位小数），中间用一个空格隔开

示例:

输入:

3 3 3

输出:

circumference=9.00 area=3.90

运行截图:

代码:

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

int main(void)
{
    int a, b, c;
    float cir, area, tp;

    scanf("%d %d %d", &a, &b, &c);
    if (a + b < c && a - b > c)
    {
        printf("三条边无法构成三角形!\n");
    }
    else
    {
        cir = (float)(a + b + c);
        tp = 0.5 * cir;
        area = sqrt(tp * (tp - a) * (tp - b) * (tp - c));
        printf("circumference=%.2f area=%.2f\n", cir, area);
    }

    return 0;
}
```

//此题应该检测是否能构成三角形之后才能参与运算;

1.10 易飞云喝酸奶

题目描述:

老师买了一箱酸奶，里面有 n 盒未打开的酸奶，*Bingo* 喜欢喝酸奶，第一时间发现了酸奶。*Bingo* 每 h 分钟能喝光一盒酸奶，并且 *Bingo* 在喝光一盒酸奶之前不会喝另一个，那么经过 m 分钟后还有多少盒未打开的酸奶？

输入描述:

多组输入，每组输入仅一行，包括 n ， h 和 m （均为整数）。输入数据保证 $m \leq n * h$

输出描述:

针对每组输入，输出也仅一行，剩下的未打开的酸奶盒数

示例:

输入:

8 5 16

输出:

4

运行截图:

代码:

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

int main(void){
    int n,h,m;
    scanf("%d %d %d",&n,&h,&m);
    int rest = n-ceil((float)m/h);
    printf("%d",rest);
    return 0;
}
```

C 语言 `math.h` 中 `ceil(x)` 函数返回不小于 x 的最小正整数。刚刚好用到这里。

PS: 向 `ceil` 中输入时记得先转换成 `float`, 否则小数点后数据丢失就得不到正确结果了。

```
#include <iostream>
using namespace std;
int main(){
    int n,h,m;
    cin>>n>>h>>m;
    int s=(m%h==0)?(n-m/h):(n-m/h-1);
    cout<<s<<endl;
    return 0;
}
```

2. 选择结构和循环结构

2.1 完美成绩判断

题目描述:

Bingo 想知道他的考试成绩是否完美, 请帮他判断。从键盘输入一个整数表示的成绩, 编程判断成绩是否在 `90~100` 之间, 如果是则输出“Perfect”。

输入描述:

多组输入，每行输入包括一个整数表示的成绩（90~100）。

输出描述:

针对每行输入，输出“Perfect”。

示例:

输入:

98

输出:

Perfect

运行截图:

代码:

```
#include <stdio.h>

int main(void)
{
    int score;

    while (scanf("%d", &score) != EOF)
    {
        if (score >= 90 && score <= 100)
        {
            puts("Perfect");
        }
    }

    return 0;
}
```

2.2 最大公约数和最小公倍数

题目描述：

陈友铭最近在课上学习了如何求两个正整数的最大公约数与最小公倍数，但是他竟然不会求两个正整数的最大公约数与最小公倍数之和，请你帮助他解决这个问题。

输入描述：

每组输入包含两个正整数 n 和 m 。

输出描述：

对于每组输入，输出一个正整数，为 n 和 m 的最大公约数与最小公倍数之和。

示例：

输入：

20 30

输出：

10

运行截图：

代码：

辗转相除法：

1. $a \% b$ 得余数 c
2. 如果 $c = 0$, 则 b 为最大公约数
3. 如果 c 不等于 0 , 则 $a = b, b = c$ 继续执行步骤 1。

```
#include<stdio.h>int main() /* 辗转相除法求最大公约数 */
{
    int m, n, a, b, t, c;
    printf("Input two integer numbers:\n");
    scanf("%d%d", &a, &b);
    m=a;    n=b;
    while (b!=0) /* 余数不为0, 继续相除, 直到余数为0 */
```



```
{ c=a%b; a=b; b=c;}
printf("The largest common divisor:%d\n", a);
printf("The least common multiple:%d\n", m*n/a);
}
```

已知 1 是一个公约数，但是 1 不是最大公约数，所以可以检测 $K=2,3,4,\dots$ 是否为 n_1 和 n_2 的公约数，直到 k 大于 n_1 或者 n_2 。将公约数存储在 `gcd` 的变量中，`gcd` 初值设为 1

```
int gcd = 1; for(int k = 2; k <= n1 && k <= n2; k++) {
    if(n1 % k == 0 && n2 % k == 0)
        gcd = k;
}
```

2.3 竞选社长

题目描述：

假设你们社团要竞选社长，有两名候选人分别是 **A** 和 **B**，社团每名同学必须并且只能投一票，最终得票多的人为社长。

输入描述：

一行，字符序列，包含 A 或 B，输入以字符 0 结束。

输出描述：

一行，一个字符，A 或 B 或 E，输出 A 表示 A 得票数多，输出 B 表示 B 得票数多，输出 E 表示二人得票数相等。

示例：

输入：

ABBABBAAB0

输出:

B

运行截图:

代码:

```
#include <stdio.h>
int main(){
    char ch;
    int suma = 0, sumb = 0;
    while((ch = getchar()) != '0' && (ch != EOF)){
        if (ch == 'A'){
            suma = suma + 1;
        }
        if (ch == 'B'){
            sumb = sumb + 1;
        }
    }
    printf("%c", (suma > sumb)? 'A':((suma < sumb)?'B':'E'));
    return 0;
}
```

```
int main()
{
    char ch;

    int suma = 0, sumb = 0;
    while ((ch = getchar()) != '0' && (ch != EOF))
    {
        if (ch == 'A')
        {
            suma++;
        }

        if (ch == 'B')
        {
```

```

        sumb++;
    }

}

printf("%c", (suma > sumb) ? 'A' : ((suma < sumb) ? 'B' : 'E'));

return 0;
}

```

2.4 计算一元二次方程

题目描述:

从键盘输入 a, b, c 的值，编程计算并输出一元二次方程 $ax^2 + bx + c = 0$ 的根，当 $a = 0$ 时，输出“Not quadratic equation”，当 $a \neq 0$ 时，根据 $\Delta = b^2 - 4ac$ 的三种情况计算并输出方程的根。

输入描述:

多组输入，一行，包含三个浮点数 a, b, c ，以一个空格分隔，表示一元二次方程 $ax^2 + bx + c = 0$ 的系数

输出描述:

针对每组输入，输出一行，输出一元二次方程 $ax^2 + bx + c = 0$ 的根的情况。

如果 $a = 0$ ，输出“Not quadratic equation”；

如果 $a \neq 0$ ，分三种情况：

$\Delta = 0$ ，则两个实根相等，输出形式为： $x1=x2=...$ 。

$\Delta > 0$ ，则两个实根不等，输出形式为： $x1=...;x2=...$ ，其中 $x1 \leq x2$ 。

$\Delta < 0$ ，则有两个虚根，则输出： $x1=实部-虚部i;x2=实部+虚部i$ ，即 $x1$ 的虚部系数小于等于 $x2$ 的虚部系数，实部为 0 时不可省略。实部 = $-b / (2*a)$ ，虚部 = $\sqrt{-\Delta} / (2*a)$

所有实数部分要求精确到小数点后 2 位，数字、符号之间没有空格。

示例:

输入:

```
2.0 7.0 1.0
0.0 3.0 3.0
1 2 1
2 2 5
1 0 1
```

输出:

```
x1=-3.35;x2=-0.15
Not quadratic equation
x1=x2=-1.00
x1=-0.50-1.50i;x2=-0.50+1.50i
x1=0.00-1.00i;x2=0.00+1.00i
```

运行截图:

代码:

```
#include <stdio.h>
#include <math.h>
int main()
{
    double a,b,c;
    while(scanf("%lf%lf%lf",&a,&b,&c)!=EOF){
        if(a==0){
            printf("Not quadratic equation\n");
            continue;
        }
        double delta=b*b-4*a*c;
        if(delta==0){
            printf("x1=x2=%.2f\n",-b/(2*a));
        }else if(delta>0){
            printf("x1=%.2f;x2=%.2f\n",-(b+sqrt(delta))/(2*a),-(b-sqrt(delta))/(2*a));
        }else{
            if(b==0) printf("x1=%.2f-%.2fi;x2=%.2f+%.2fi\n",b,sqrt(-delta)/(2*a),b,sqrt(-delta)/(2*a));
            else
                printf("x1=%.2f-%.2fi;x2=%.2f+%.2fi\n",b/(-2*a),sqrt(-delta)/(2*a),b/(-2*a),sqrt(-
delta)/(2*a));
        }
    }
}
```

2.5 获取月份天数

题目描述：

Bingo 想获得某年某月有多少天，请帮他编程实现。输入年份和月份，计算这一年这个月有多少天。

输入描述：

多组输入，一行有两个整数，分别表示年份和月份，用空格分隔。

输出描述：

针对每组输入，输出为一行，一个整数，表示这一年这个月有多少天。

示例：

输入：

2008 2

输出：

29

运行截图：

代码：

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

```
int main(void){
    int y , m;
    while(scanf("%d %d", &y,&m)!= EOF){
        if(y%4==0 && m==2){
            printf("29\n");
        }else if(m==2){
            printf("28\n");
        }else if((m<8&& m%2==1)|| (m>=8 && m%2 == 0)){
            printf("31\n");
        }else{
            printf("30\n");
        }
    }
    return 0;
}
```

2.6 三角形判断

题目描述:

Bingo 想知道已经给出的三条边 a , b , c 能否构成三角形, 如果能构成三角形, 判断三角形的类型 (等边三角形、等腰三角形或普通三角形)。

输入描述:

题目有多组输入数据, 每一行输入三个 a , b , c ($0 < a, b, c < 1000$), 作为三角形的三个边, 用空格分隔。

输出描述:

针对每组输入数据, 输出占一行, 如果能构成三角形, 等边三角形则输出 "Equilateral triangle!", 等腰三角形则输出 "Isosceles triangle!", 其余的三角形则输出 "Ordinary triangle!", 反之输出 "Not a triangle!"。

示例:

输入:

2 3 2

3 3 3

输出:

Isosceles triangle!

Equilateral triangle!

运行截图:

代码:

```
#include<stdio.h>
#include<math.h>
int main()
{
    float a,b,c;
    while(scanf("%f %f %f",&a,&b,&c)!=EOF)
    {
        if(a+b>c&&a+c>b&&b+c>a)
        {
            if(a==b&&b==c)
                printf("Equilateral triangle!\n");
            else if(a==b||a==c||b==c)
                printf("Isosceles triangle!\n");
            else
                printf("Ordinary triangle!\n");
        }
        else
            printf("Not a triangle!\n");
    }
    return 0;
}
```

2.7 HTTP 状态码

题目描述:

Bingo 访问网站，得到 HTTP 状态码，但他不知道什么含义，焦老师告诉他常见 HTTP 状态码：200（OK，请求已成功），202（Accepted，服务器已接受请求，但尚未处理。）400（Bad Request，请求参数有误），403（Forbidden，被禁止），404（Not Found，请求失败），500（Internal Server Error，服务器内部错误），502（Bad Gateway，错误网关）。

输入描述:

多组输入，一行，一个整数（100~600），表示 HTTP 状态码

输出描述:

针对每组输入的 HTTP 状态，输出该状态码对应的含义，具体对应如下：

200-OK
202-Accepted
400-Bad Request
403-Forbidden
404-Not Found
500-Internal Server Error
502-Bad Gateway

示例:

输入:

200

输出:

OK

运行截图:

代码:

```
#include <stdio.h>
int main(){
    int a;
    while(~scanf("%d",&a)){
        if(a==200)printf("OK\n");
        if(a==202)printf("Accepted\n");
        if(a==400)printf("Bad Request\n");
        if(a==403)printf("Forbidden\n");
        if(a==404)printf("Not Found\n");
        if(a==500)printf("Internal Server Error\n");
```



```
if(a==502)printf("Bad Gateway\n");  
  
} return 0;  
}
```

2.8 统计成绩

题目描述：

输入 n 科成绩（浮点数表示），统计其中的最高分，最低分以及平均分。

输入描述：

两行，

第 1 行，正整数 n ($1 \leq n \leq 100$)

第 2 行， n 科成绩（范围 $0.0 \sim 100.0$ ），用空格分隔。

输出描述：

输出一行，三个浮点数，分别表示，最高分，最低分以及平均分（小数点后保留 2 位），用空格分隔。

示例：

输入：

5

```
99.5 100.0 22.0 60.0 88.5
```

输出:

```
100.00 22.00 74.00
```

运行截图:

代码:

```
#include <stdio.h>

int main()
{
    int n,i;
    scanf("%d",&n);
    float a[n];
    float max=0,min=100,avg=0;
    for(i=0;i<n;i++)
    {
        scanf("%f",&a[i]);
        if(a[i]>max) max = a[i];
        if(a[i]<min) min = a[i];
        avg += a[i];
    }
    avg /= i;
    printf("%.2f %.2f %.2f\n",max,min,avg);
    return 0;
}
```

2.9 直角三角形图案

题目描述：

陈友铭学习了循环，Bingo 老师给他出了一系列打印图案的练习，该任务是打印用“*”组成的直角三角形图案。

输入描述：

多组输入，一个整数（2~20），表示直角三角形直角边的长度，即“*”的数量，也表示输出行数。

输出描述：

针对每行输入，输出用“*”组成的对应长度的直角三角形，每个“*”后面有一个空格。

示例：

输入：

5

输出：

```
*
* *
* * *
* * * *
* * * * *
```

运行截图：

代码：

```
#include<stdio.h>
int main(){
    int a;
    while(scanf("%d",&a)!=EOF){
        for(int i=0;i<a;i++){
            for(int j=0;j<=i;j++){
                printf("* ");
            }
            printf("\n");
        }
    }
}
```

2.10 公务员面试

题目描述：

Bingo 去面试公务员公务员面试现场打分。有 7 位考官，从键盘输入若干组成绩，每组 7 个分数（百分制），去掉一个最高分和一个最低分，输出每组的平均成绩。

输入描述：

一行，输入 7 个整数（0~100），代表 7 个成绩，用空格分隔。

输出描述：

一行，输出去掉最高分和最低分的平均成绩，小数点后保留 2 位，每行输出后换行。

示例：

输入：

99 45 78 67 72 88 60

输出：

73.00

运行截图：

代码：

```
#include <stdio.h>
int main()
{
    int i,max=-1,min=101,a[7];
```

```
double sum=0;
for(i=0;i<7;i++){
    scanf("%d",&a[i]);
    if(max<a[i]) max=a[i];
    if(min>a[i]) min=a[i];
    sum+=a[i];
}
sum-=max;
sum-=min;
sum/=5;
printf("%.2f\n",sum);
}
```

2.11 X 型图案

题目描述:

陈友铭学习了循环，Bingo 老师给他出了一系列打印图案的练习，该任务是打印用“*”组成的 X 形图案。

输入描述:

多组输入，一个整数（2~20），表示输出的行数，也表示组成“x”的反斜线和正斜线的长度。

输出描述:

针对每行输入，输出用“*”组成的 x 形图案

示例:

输入:

5

输出:

* *

* *

*

* *

* *

运行截图:

代码:

```
#include<stdio.h>
int main(void)
{
    int a;
    int i,j;
    while(scanf("%d\n",&a)!=EOF)
    {
        for(i=1;i<=a;i++)
        {
            for(j=1;j<=a;j++)
            {
                if(i==j || i+j==a+1)
                    printf("*");
                else
                    printf(" ");
            }
            printf("\n");
        }
    }
    return 0;
}
```

2.12 空心正方形图案

题目描述：

陈友铭学习了循环，*Bingo* 老师给他出了一系列打印图案的练习，该任务是打印用“*”组成的“空心”正方形图案。

输入描述：

多组输入，一个整数（3~20），表示输出的行数，也表示组成正方形边的“*”的数量。

输出描述：

针对每行输入，输出用“*”组成的“空心”正方形，每个“*”后面有一个空格。

示例：

输入：

4

输出：

```
* * * *  
*      *  
*      *  
* * * *
```

运行截图：

代码：

```
#include<stdio.h>  
int main()  
{  
    int num,i,j;  
    while(scanf("%d",&num)!=EOF){  
        for(i=1;i<=num;i++){  
            for(j=1;j<=num;j++){  
                if(j==1 || j==num || i==1 || i==num)  
                {  
                    printf("* ");  
                }  
            }  
        }  
    }
```

```
    }  
    else printf(" ");  
    }  
    printf("\n");  
    }  
    }  
    return 0;  
}
```

2.13 奇偶数统计

题目描述:

任意输入一个正整数 N ，统计 $1 \sim N$ 之间奇数的个数和偶数的个数，并输出。

输入描述:

一行，一个正整数 N 。 ($1 \leq N \leq 100,000$)

输出描述:

一行， $1 \sim N$ 之间奇数的个数和偶数的个数，用空格分开。

示例:

输入:

5

输出:

3 2

运行截图：

代码：

```
#include <stdio.h>
int main(){
    int n;
    scanf("%d",&n);
    if(n%2==0)
        printf("%d %d",n/2,n/2);
    else
        printf("%d %d",n/2+1,n/2);
    return 0;
}
```

2.14 求质数个数

题目描述：

陈友铭知道了什么是质数（只能被 1 和他自身整除的数），他现在想知道所有三位整数中，有多少个质数。

输入描述：

无

输出描述：

一行，一个整数，表示所有三位整数中，有多少个质数。

运行截图:

代码:

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

int main(void)
{
    int count = 0;
    int i, j, isprime;

    for (i = 100; i < 1000; i++)
    {
        isprime = 1;
        for (j = 2; j <= sqrt(i); j++)
        {
            if (0 == i % j)
            {
                isprime = 0;
                break;
            }
        }
        if (isprime)
        {
            count++;
        }
    }
    printf("%d\n", count);

    return 0;
}
```

2.15 包含 9 的数字

题目描述:

2019 年已经过去了, 陈友铭想知道 1~2019 中有多少个包含数字 9 的数。包含数字的数是指有某一位是“9”的数, 例如“2019”、“299”等。

输入描述:

无

输出描述:

一行, 一个整数, 表示 1~2019 中共有多少个数包含数字 9。

运行截图:

代码:

```
#include <stdio.h>

int main()
{
    int i;
    int num=0;
    for(i=1;i<=2019;i++)
    {
        if(i%10==9)
        {
            num++;
            continue;
        }
        else if((i/10)%10==9)
        {
            num++;
            continue;
        }
        else if((i/100)%10==9)
        {
            num++;
            continue;
        }
    }
    printf("%d",num);

    return 0;
}
```

3. 指针、数组、字符串、函数

3.1 有序序列插入整数

题目描述：

有一个有序数字序列，从小到大排序，将一个新输入的数插入到序列中，保证插入新数后，序列仍然是升序。

输入描述：

第一行输入一个整数 N ($0 \leq N \leq 50$)。

第二行输入 N 个升序排列的整数，输入用空格分隔的 N 个整数。

第三行输入想要进行插入的一个整数。

输出描述：

输出为一行， $N+1$ 个有序排列的整数。

示例：

输入：

5

1 6 9 22 30

8

输出:

1 6 8 9 22 30

运行截图:

代码:

```
#include<stdio.h>
int main()
{
    int i,j,N,a[50],t,f;
    scanf("%d",&N);
    for(i=0;i<N;i++)
    {
        scanf("%d",&a[i]);
    }
    getchar();
    scanf("%d",&t);
    for(i=0;i<N;i++)
    {
        if(t<a[i])
        {
            for(j=N;j>i;j--)
            {
                a[j]=a[j-1];
            }
            break;
        }
    }
    a[i]=t;
    for(i=0;i<=N;i++)
        printf("%d ",a[i]);
    return 0;
}
```

3.2 杨辉三角

题目描述:

梦奇知道什么叫杨辉三角之后对杨辉三角产生了浓厚的兴趣，他想知道杨辉三角的前 n 行，请编程帮他解答。杨辉三角，本质上是二项式 $(a+b)$ 的 n 次方展开后各项的系数排成的三角形。其性质包括：每行的 endpoint 数为 1，一个数也为 1；每个数等于它左上方和上方的两数之和。

输入描述:

第一行包含一个整数 n 。 ($1 \leq n \leq 30$)

输出描述:

包含 n 行，为杨辉三角的前 n 行，每个数输出域宽为 5。

示例:

输入:

6

输出:

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
```

运行截图:

```
#include <stdio.h>
int main(){
    int n;
    int i, j;
    int a[30][30] = {0};
    scanf("%d", &n);
    for(i=0; i<n; i++){
        a[i][0] = 1;
    }
    for(i=1; i<n; i++){
        for(j=1; j<=i; j++){
            a[i][j] = a[i-1][j-1] + a[i-1][j];
        }
    }
    for(i=0; i<n; i++){
        for(j=0; j<=i; j++){
            printf("%5d", a[i][j]);
        }
        printf("\n");
    }
}
```

```
}  
    return 0;  
}
```

3.3 序列中整数去重

题目描述：

输入 n 个整数的序列，要求对这个序列进行去重操作。所谓去重，是指对这个序列中每个重复出现的整数，只保留该数第一次出现的位置，删除其余位置。

输入描述：

输入包含两行，第一行包含一个正整数 n ($1 \leq n \leq 1000$)，表示第二行序列中数字的个数；第二行包含 n 个整数（范围 $1 \sim 5000$ ），用空格分隔。

输出描述：

输出为一行，按照输入的顺序输出去重之后的数字，用空格分隔。

示例：

输入：

```
5  
10 12 93 12 75
```

输出：

```
10 12 93 75
```

运行截图：

代码：

```
#include <stdio.h>  
int main()  
{  
    int n,i,j,k;  
    scanf("%d",&n);  
    int a[n];  
    for(i=0;i<n;i++)  
    {  
        scanf("%d",&a[i]);  
    }  
}
```

```
for(i=0;i<n;i++)
{
    for(j=i+1;j<n;j++)
    {
        if(a[i]==a[j])
        {
            for(k=j;k<n;k++)
            {
                a[k]=a[k+1];
            }
            n--;
            j--;
        }
    }
}
for(i=0;i<n;i++)
    printf("%d ",a[i]);
}
```

3.4 最高分与最低分之差

题目描述:

输入 n 个成绩，换行输出 n 个成绩中最高分数和最低分数的差。

输入描述:

两行，第一行为 n ，表示 n 个成绩，不会大于 10000。

第二行为 n 个成绩（整数表示，范围 0~100），以空格隔开。

输出描述:

一行，输出 n 个成绩中最高分数和最低分数的差。

示例:

输入:

10

98 100 99 97 95 99 98 97 96 100

输出:

5

运行截图:

代码:

```
#include<stdio.h>

int main(void)
{
    int n;
    scanf("%d",&n);

    int i;
    int chengji[n];
    for(i=0;i<n;i++)
    {
        scanf("%d",chengji+i);
    }

    int max,min;
    max = min =chengji[0];

    for(i=0;i<n;i++)
    {
        if(chengji[i] >= max)
        {
            max = chengji[i];
        }
        if(chengji[i] <= min)
        {
            min = chengji[i];
        }
    }

    printf("%d",max-min);
}
```

3.5 陈友铭查找数字

题目描述:

给定 n 个整数和一个整数 x ，小乐乐想从 n 个整数中判断 x 出现几次，请编程帮他实现。

输入描述:

共 3 行
第一行，一个整数，表示 n ($1 \leq n \leq 100$)。
第二行，共输入 n 个整数，两个整数之间用空格分隔。
第三行，输入一个整数，表示要查找的整数 x 。

输出描述:

一行，表示整数 x 出现的次数。

示例:

输入:

5
1 1 2 1 3
1

输出:

3

运行截图:

代码:

```
#include<stdio.h>
int main()
{
    int n,x,count=0;
    scanf("%d",&n);
    int a[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    scanf("%d",&x);
```

```
for(int i=0;i<n;i++)
{
    if(a[i]==x)
        count++;
}
printf("%d ",count);
return 0;
}
```

3.6 矩阵元素定位

题目描述:

陈友铭得到了一个 n 行 m 列的矩阵，现在他想知道第 x 行第 y 列的值是多少，请你帮助他完成这个任务。

输入描述:

第一行包含两个数 n 和 m ，表示这个矩阵包含 n 行 m 列。从 2 到 $n+1$ 行，每行输入 m 个整数，用空格分隔，共输入 $n*m$ 个数，表示矩阵中的元素。接下来一行输入 x 和 y ，用空格分隔，表示 Kiki 想得到的元素的位置。($1 \leq x \leq n \leq 10$, $1 \leq y \leq m \leq 10$)

输出描述:

一行，输出一个整数值，为陈友铭想知道的值。

示例:

输入:

```
2 3
1 2 3
4 5 6
1 2
```

输出:

```
2
```

运行截图:

代码:

```
#include <stdio.h>
int main()
{
    int a[15][15];
    int i,j;
    int n,m;
    scanf("%d%d",&n,&m);
    for(i=0;i<n;i++){
        for(j=0;j<m;j++){
            scanf("%d",&a[i][j]);
        }
    }
    scanf("%d%d",&n,&m);
    printf("%d\n",a[n-1][m-1]);
}
```

3.7 最高身高

题目描述:

陈友铭想从 n 行 m 列的方阵队列中找到身高最高的人的位置, 请你帮助他完成这个任务。

输入描述:

第一行包含两个整数 n 和 m , 表示这个方阵队列包含 n 行 m 列。从 2 到 $n+1$ 行, 每行输入 m 个整数, 用空格分隔, 共输入 $n*m$ 个数, 表示方阵中的所有人的身高 (保证输入身高都不相同)。($1 \leq x \leq n \leq 10$, $1 \leq y \leq m \leq 10$)

输出描述:

一行, 输出两个整数, 用空格分隔, 表示方阵中身高最高的人所在的行号和列号。

示例:

输入:

```
2 2
175 180
176 185
```

输出:

```
2 2
```

运行截图:

代码:

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

int main()
{
    int m, n;
    int i, j;
    int maxm=0, maxn=0;
    scanf("%d %d", &m, &n);
    int a[m][n];
    for(i=0;i<m;i++)
    for(j=0;j<n;j++)
    scanf("%d", &a[i][j]);

    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            if(a[i][j]>a[maxm][maxn])
            {
                maxm=i;
                maxn=j;
            }
        }
    }

    printf("%d %d", maxm+1, maxn+1);
    return 0;
}
```

3.8 矩阵相等

题目描述:

陈友铭得到了两个 n 行 m 列的矩阵, 他想知道两个矩阵是否相等, 请你回答他。(当两个矩阵对应数组元素都相等时两个矩阵相等)。

输入描述:

第一行包含两个整数 n 和 m , 表示两个矩阵包含 n 行 m 列, 用空格分隔。

从 2 到 $n+1$ 行, 每行输入 m 个整数, 用空格分隔, 共输入 $n*m$ 个数, 表示第一个矩阵中的元素。

从 $n+2$ 行到 $2n+1$, 每行输入 m 个整数, 用空格分隔, 共输入 $n*m$ 个数, 表示第二个矩阵中的元素。 $1 < n, m < 10$

输出描述:

一行, 如果两个矩阵相等输出 "Yes" 并换行, 否则输出 "No" 并换行。

示例:

输入:

```
2 2
1 2
3 4
1 2
3 4
```

输出:

Yes

运行截图:

```
#include<stdio.h>
int main(){
    int n,m,i,j;
    int num=0;
    scanf("%d %d",&n,&m);
    int a[10][10],b[10][10];
    for(i=0;i<n;i++){
        for(j=0;j<m;j++){
            scanf("%d",&a[i][j]);
        }
    }
    for(i=0;i<n;i++){
        for(j=0;j<m;j++){
            scanf("%d",&b[i][j]);
        }
    }
    for(i=0;i<n;i++){
        for(j=0;j<m;j++){
            if(a[i][j]==b[i][j])
                num++;
        }
    }
    if(num==n*m)
        printf("Yes\n");
    else
        printf("No\n");
}
```

3.9 统计正负数的个数

题目描述:

输入 10 个整数，分别统计输出正数、负数的个数。

输入描述:

输入 10 个整数，用空格分隔。

输出描述:

两行，第一行正数个数，第二行负数个数。

示例:

输入:

-1 2 3 -6 7 8 -1 6 8 10

输出:

positive:7

negative:3

运行截图:

```
#include <stdio.h>
int main()
{
    int a[10];
    for (int i=0;i<10;i++)
    {
        scanf("%d",&a[i]);
    }
    int x=0;
    int y=0;
    for (int i=0;i<10;i++)
    {
        if (a[i]>=0)
            x++;
        else
            y++;
    }
    printf("positive:%d\n",x);
    printf("negative:%d\n",y);
    return 0;
}
```

3.10 矩阵交换

题目描述:

陈友铭有一个矩阵，他想知道经过 k 次行变换或列变换后得到的矩阵。请编程帮他解答。

输入描述:

第一行包含两个整数 n 和 m ，表示一个矩阵包含 n 行 m 列，用空格分隔。 ($1 \leq n \leq 10, 1 \leq m \leq 10$)

从 2 到 $n+1$ 行，每行输入 m 个整数，用空格分隔，共输入 $n*m$ 个数，表示第一个矩阵中的元素。

接下来一行输入 k ，表示要执行 k 次操作 ($1 \leq k \leq 5$)。接下来有 k 行，每行包括一个字符 t 和两个数 a 和 b ，中间用空格分隔， t 代表需要执行的操作，当 t 为字符 'r' 时代表进行行变换，当 t 为字符 'c' 时代表进行列变换， a 和 b 为需要互换的行或列 ($1 \leq a \leq b \leq n \leq 10, 1 \leq a \leq b \leq m \leq 10$)。

输出描述:

输出 n 行 m 列，为矩阵交换后的结果。每个数后面有一个空格。

示例:

输入:

2 2

1 2

3 4

1

r 1 2

输出:

3 4

1 2

运行截图:

代码:

```
#include<stdio.h>
int main(){
    int n,m,i,j,k,a,b,c,temp;
    char op;
    scanf("%d %d",&n,&m);
    int A[n][m];
    for(i=0;i<n;i++){
        for(j=0;j<m;j++){
            scanf("%d",&A[i][j]);
        }
    }
    scanf("%d",&c);
    for(i=0;i<c;i++){
        getchar();
        scanf("%c %d %d",&op,&a,&b);
        if(op=='r'){
            for(j=0;j<m;j++){
                temp=A[a-1][j];
                A[a-1][j]=A[b-1][j];
                A[b-1][j]=temp;
            }
        }
        else if(op=='c'){
            for(k=0;k<n;k++){
                temp=A[k][a-1];
                A[k][a-1]=A[k][b-1];
                A[k][b-1]=temp;
            }
        }
    }
    for(i=0;i<n;i++){
        for(j=0;j<m;j++){
            printf("%d ",A[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

3.11 找出并输出其中的最大数和最小数

题目描述:

任意输入 $a[N][N]$ 个数，找出并输出其中的最大数和最小数，以及它们的下标。

运行截图:

```
main()

{

    int i,j,min_i,min_j,max_i,max_j;

    int a[3][3],max,min;

    for(i=0;i<3;i++)

        for(j=0;j<3;j++)

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

    min=max=a[i][j];

    min_i=min_j=max_i=max_j=0;

    for(i=0;i<3;i++)

        for(j=0;j<3;j++)

        {

            if(a[i][j]>max)

                {max=a[i][j];max_i=i;max_j=j;}

            if(a[i][j]<min)

                {min=a[i][j];min_i=i;min_j=j;}

        }

    printf("最大元素为 %d ,下标为 (%d, %d) \n",max,max_i,max_j);

    printf("最小元素为 %d ,下标为 (%d, %d) \n",min,min_i,min_j);

}
```

3.12 马克思手稿中的趣味数学题

题目描述:

马克思手稿中有一道趣味数学题：有 30 个人，其中有男人、女人和小孩，在一家饭馆里吃饭共花了 50 先令，每个男人各花 3 先令，每个女人各花 2 先令，每个小孩各花 1 先令，问男人、女人和小孩各有几人？

运行截图:

代码:

```
#include <stdio.h>

void main()
{
    int man, woman;
    for (man=1; man<=17; man++)
        for (woman=1; woman<=25; woman++) {
            if (50 == man*3 + woman*2 + (30-man-woman) )
                printf("man:%d, woman:%d, child:%d\n", man, woman, 30-man-woman);
        }
}
```

```
}
```

3.13 将字符串中的前导*号全部删除，中间和后面的*号不删除

题目描述：

假定输入的字符串中只包含字母和*号。请编写函数 fun()，它的功能是：将字符串中的前导*号全部删除，中间和后面的*号不删除。例如，若字符串中的内容为****A*BC*DEF*G*****，删除后，字符串中的内容则应当是A*BC*DEF*G*****。

运行截图：

```
main(){
```

```
char s[81],*t,*f;
int m=0,tn=0,fn=0;
printf("Enter a string:\n");gets(s);
t=f=s;
while(*t){
    t++,m++;/*m 为字符串的长度*/
}
t--;/*指针 t 指向字符串尾部*/
while(*t=='*'){
    t--;tn++;/*指针 t 指向最后一个字母，tn 统计尾部*数量。*/
}
while(*f=='*'){
    f++,fn++;
}
fun(s,m,fn,tn);
printf("The string after deleted:\n");
puts(s);
}
```

3.6 同时能被 3 与 7 整除的所有自然数之和的平方根 s

题目描述:

编写函数 fun(), 它的功能是求 n 以内 (不包括 n) 同时能被 3 与 7 整除的所有自然数之和的平方根 s, 并做为函数值返回。例如: n 为 1000 时, 函数值应为 s = 153.909064

```
#include <stdio.h>
#include <math.h>
double fun(int n){
}
main(){
    printf("%lf",fun(1000));
}
```

运行截图:

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <math.h>

double fun(int n) {
    double s = 0.0;int i;
    for (i = 0; i < n; i++) /* 从 0~n 中找到既能被 3 整除又能被 7 整除的数, 并求和 */
        if(i%3==0&& i%7==0)
            s = s + i;
```

```
    s = sqrt(s); /* 对 s 求平方根 */  
  
    return s;  
}
```

```
int main()  
{  
  
    printf("%lf\n", fun(1000));  
  
    system("pause");  
  
    return 0;  
}
```

3.7 从字符中删除指定的字符，同一字母的大、小写按不同字符处理

题目描述：

编写函数 fun, 该函数的功能是：从字符中删除指定的字符，同一字母的大、小写按不同字符处理。

运行截图:

个人资料请勿外传

3.10 求出小于或等于 lim 的所有素数并放在 aa 数组

题目描述:

编写函数 `int fun(int lim,int aa[max])`, 该函数的功能是求出小于或等于 `lim` 的所有素数并放在 `aa` 数组中, 该函数返回所求的素数的个数

输入描述:

多组输入, 每行输入包括一个整数表示的成绩 (90~100)。

输出描述:

针对每行输入, 输出 "Perfect"。

示例:

输入:

98

输出:

Perfect

运行截图:

3.11 判断字符串是否为回文

题目描述:

请编写函数 fun，该函数的功能是：判断字符串是否为回文?若是则函数返回 1，主函数中输出 yes，否则返回 0，主函数中输出 no。回文是指顺读和倒读都是一样的字符串

运行截图：

3.12 将 ss 所指字符串中所有下标为奇数位置上的字母转换为大写

题目描述：

请编写一个函数 fun，它的功能是：将 ss 所指字符串中所有下标为奇数位置上的字母转换为大写(若该位置上不是字母，则不转换)

运行截图：

代码：

3.13 将 s 所指字符串中除了下标为奇数

题目描述：

将 s 所指字符串中除了下标为奇数、同时 ascii 值也为奇数的字符之外，其余的所有字符都删除，串中剩余字符所形成的一个新串放在 t 所指的一个数组中

运行截图：

代码：

3.14 使字符串中尾部的*号不得多于 n 个

题目描述：

题目描述：

假定输入的字符串中只包含字母和*号。请编写函数 fun，它的功能是：使字符串中尾部的*号不得多于 n 个；若多于 n 个，则删除多于的*号；若少于或等于 n 个，则什么也不做，字符串中间和前面的*号不删除。

运行截图：

代码：

3.15 删除字符串中的所有空格

题目描述：

请编写一个函数，用来删除字符串中的所有空格

运行截图：

代码：

3.16 将字符串中的前导*号全部移到字符串的尾部

题目描述：

假定输入的字符串中只包含字母和*号。请编写函数 fun，它的功能是：将字符串中的前导*号全部移到字符串的尾部

运行截图：

代码：

3.17 求出该学生的平均分放在记录的 ave 成员中

题目描述：

某学生的记录由学号、8 门课程成绩和平均分组成，学号和 8 门课程的成绩已在主函数中给出。请编写函数 fun，它的功能是：求出该学生的平均分放在记录的 ave 成员中。请自己定义正确的形参。

运行截图：

代码：

3.18 求出 ss 所指字符串中指定字符的个数

题目描述:

题目描述:

请编写函数 fun，它的功能是：求出 ss 所指字符串中指定字符的个数，并返回此值。

运行截图:

代码:

3.19 移动一维数组中的内容

题目描述:

请编写函数 fun，该函数的功能是：移动一维数组中的内容，若数组中有 n 个整数，要求把下标从 0 到 p (p 小于等于 n-1) 的数组元素平移到数组的最后

运行截图:

代码:

3.20 该函数的功能是移动字符串中内容

题目描述:

请编写函数 fun，该函数的功能是移动字符串中内容，移动的规则如下：把第 1 到第 m 个字符，平移到字符串的最后，把第 m+1 到最后的字符移到字符串的前部。

运行截图:

代码:

3.21 按列的顺序依次放到一个字符串中

题目描述:

请编写函数 fun，该函数的功能是：将 m 行 n 列的二维数组中的字符数据，按列的顺序依次放到一个字符串中。

运行截图:

代码:

3.22 将数组右上半三角元素中的值乘以 m

题目描述:

下列程序定义了 $n \times n$ 的二维数组, 并在主函数中自动赋值。请编写函数 `fun(int a[][n], int n)`, 该函数的功能是: 将数组右上半三角元素中的值乘以 m

运行截图:

代码:

3.23 找出一个最长的一个字符串

题目描述:

编写一个函数, 从传入的 num 个字符串中找出一个最长的一个字符串, 并通过形参指针 max 传回该串地址 (用 **** 作为结束输入的标志)

运行截图:

代码:

3.24 统计一个长度为 2 的字符串在另一个字符串中出现的次数

题目描述:

编写一个函数, 该函数可以统计一个长度为 2 的字符串在另一个字符串中出现的次数。

运行截图:

代码:

3.25 实现两个字符串的连接

题目描述:

编写一个函数 fun, 它的功能是: 实现两个字符串的连接 (不使用库函数 `strcat`), 即把 p2 所指的字符串连接到 p1 所指的字符串后。

运行截图:

代码:

3.26 二维数组中的数据, 按行的顺序依次放到一维数组中

题目描述:

请编写函数 fun, 该函数的功能是: 将 m 行 n 列的二维数组中的数据, 按行的顺序依次放到一维数组中, 一维数组中数据的个数存放在形参 n 所指的储存单元中。

运行截图:

代码:

3.27 除了尾部的*号之外, 将字母串中其他*号全部删除

题目描述:

假定输入的字符串中只包含字母和*号。请编写函数 fun, 它的功能是: 除了尾部的*号之外, 将字母串中其他*号全部删除。形参 p 已指向字符串中最后的一个字母。不使用 c 的字符串函数。

运行截图:

代码:

3.28 删去一维数组中所有相同的数

题目描述:

请编写函数 fun, 该函数的功能是: 删去一维数组中所有相同的数, 使之只剩一个。数组中的数已按由小到大的顺序排列, 函数返回删除后数组中数据的个数

运行截图:

代码:

3.29 计算并输出下列多项式的值

题目描述:

请编写函数 fun, 其功能时: 计算并输出下列多项式的值:

$$sn=1+1/1!+1/2!+1/3!+1/4!+\cdots+1/n!$$

运行截图:

代码:

3.30 将两个两位数的正整数 ab 合并形成一个整数放在 c 中

题目描述:

编写一个函数 fun，其功能是：将两个两位数的正整数 ab 合并形成一个整数放在 c 中。合并方式是：将 a 数的十位和个位数依次放在 c 数的个位和百位上，b 数的十位和个位数依次放在 c 数的十位和千位上。

运行截图:

代码:

3.31 使数组左下半三角元素中的值乘以 n

题目描述:

程序定义了 $n \times n$ 的二维数组，并在主函数中自动赋值。请编写函数 fun，该函数的功能是：使数组左下半三角元素中的值乘以 n

运行截图:

代码:

3.32 将字符串尾部的*号全部删除，前面和中间的*号不删除

题目描述:

假定输入的字符串中只包含字母和*号。请编写函数 fun，它的功能使：将字符串尾部的*号全部删除，前面和中间的*号不删除

运行截图:

代码:

3.33 计算并输出下列多项式值

题目描述:

请编写函数 fun，其功能是：计算并输出下列多项式值： $sn=(1-1/2)+(1/3-1/4)+\cdots+(1/(2n-1)-1/2n)$.

运行截图:

代码:

个人资料请勿外传

3.34 求出 1 到 100 之内能被 7 或者 11 整除

题目描述:

请编写函数 fun，它的功能是：求出 1 到 100 之内能被 7 或者 11 整除，但不能同时被 7 和 11 整除的所有整数，并将他们放在 a 所指的数组中，通过 n 返回这些数的个数

运行截图:

代码:

3.35 编写一个函数 fun

题目描述:

编写一个函数 fun，它的功能是：根据以下公式求 p 的值，结果由函数值带回。M 与 n 为两个正整数，且要求 $m > n$ 。 $p = m! / n! (m-n)!$

运行截图:

代码:

3.36 实现矩阵(3 行 3 列)的转置

题目描述:

编写程序，实现矩阵(3 行 3 列)的转置(即行列互换)

运行截图:

代码:

3.37 计算并输出 3 到 n 之间所有素数的平方根之和

题目描述:

请编写函数 fun，其功能是：计算并输出 3 到 n 之间所有素数的平方根之和

运行截图:

代码:

3.38 计算并输出 n 被 5 或 9 整除的所有自然数的倒数之和

题目描述:

请编写函数 fun，它的功能是：计算并输出 n (包括 n) 以内能被 5 或 9 整除的所有自然数的倒数之和

运行截图:

代码:

3.39 计算并输出给定 10 个数的方差

题目描述:

请编写函数 fun，计算并输出给定 10 个数的方差

运行截图:

代码:

3.40 计算 n 门课程的平均分

题目描述:

请编写一个函数 fun，它的功能使：计算 n 门课程的平均分，计算结果作为函数值返回。

运行截图:

代码:

4. 文件

4.1 C 语言文件操作

题目描述:

C 语言实现文件的创建、读、写、关闭

运行截图:

代码:

4.2 win32 API 文件操作

题目描述:

Win32 API 实现文件的创建、读、写、关闭

运行截图:

代码:

4.3 C++文件操作

题目描述:

C++实现文件的创建、读、写、关闭

输入描述:

第一行输入一个整数 N ($0 \leq N \leq 50$)。

第二行输入 N 个升序排列的整数，输入用空格分隔的 N 个整数。

第三行输入想要进行插入的一个整数。

输出描述:

输出为一行， $N+1$ 个有序排列的整数。

示例:

输入:

```
5
1 6 9 22 30
8
```

输出:

```
1 6 8 9 22 30
```

运行截图:

代码:

4.4 MFC 文件操作

题目描述:

MFC 语言实现文件的创建、读、写、关闭

输入描述:

```
第一行输入一个整数 N (0≤N≤50) 。
第二行输入 N 个升序排列的整数，输入用空格分隔的 N 个整数。
第三行输入想要进行插入的一个整数。
```

输出描述:

```
输出为一行，N+1 个有序排列的整数。
```

示例:

输入:

```
5
1 6 9 22 30
8
```

输出:

1 6 8 9 22 30

运行截图:

代码:

4.5 配置文件操作

题目描述:

实现配置文件的创建、读、写、关闭，以及按键名搜索

运行截图:

代码:

5. 结构体、枚举

5.1 把分数最高的学生数据放在 h 所指的数组

题目描述:

学生的记录由学号和成绩组成，n 名学生的数据已在主函数中放入结构体数组 s 中，请编写函数 fun，它的功能使：把分数最高的学生数据放在 h 所指的数组中，注意：分数最高的学生可能不止一个，函数返回分数最高的学生的人数

输入描述:

运行截图:

代码:

5.2 找出成绩最低的学生记录，通过形参返回主函数

题目描述:

已知学生的记录由学号和学习成绩构成， n 名学生的数据已存入 a 结构体数组中。请编写函数 fun ，该函数的功能是：找出成绩最低的学生记录，通过形参返回主函数(规定只有一个最低分)。

运行截图:

代码:

5.3 把高于等于平均分的学生数据放在 b 所指的数组中，高于等于平均分的学生人数通过形参 n 传回

题目描述:

学生的记录是由学号和成绩组成， n 名学生的数据已在主函数中放入结构体数组 s 中，请编写函数 fun ，它的功能是：把高于等于平均分的学生数据放在 b 所指的数组中，高于等于平均分的学生人数通过形参 n 传回，平均分通过函数值返回

运行截图:

代码:

5.4 输出 5 个学生的数据记录

题目描述:

编写 `input()` 和 `output()` 函数输入，输出 5 个学生的数据记录。（用结构体设计，学生记录中包括学号、姓名、四门课程成绩）

运行截图:

代码:

5.5 结构体内容拷贝

题目描述:

结构体内容拷贝

运行截图:

代码:

百题魔鬼训练营详解（进阶篇）

6. 链表

6.1 找出学生的最高分，由函数值返回

题目描述：

n 名学生的成绩已在主函数中放入一个带头节点的链表结构中，h 指向链表的头节点。请编写函数 fun，它的功能是：找出学生的最高分，由函数值返回

运行截图：

代码：

6.2 有逆置/反转单链表

题目描述：

遍历链表，同时将数据头插入链表，新的链表为反转链表

运行截图：

代码：

6.3 单链表冒泡排序

题目描述:

单链表冒泡排序

运行截图:

代码:

6.4 构造一个环形链表

题目描述:

构造一个环形链表

运行截图:

代码:

6.5 链表实现最后退出的人原来的序号

题目描述:

13 个人转成一圈，从第一个人开始报数，报到 3 的退出，问最后退出的人原来的序号。（用链表实现）

运行截图:

代码:

6.6 反向输出一个链表

题目描述:

反向输出一个链表

运行截图:

代码:

6.7 链表合并

题目描述:

已有 a,b 两个链表，每个链表中的的结点包括学号、成绩。要求把两个链表合并，按学号升序排列

运行截图:

代码:

6.8 链表判断相同节点

题目描述:

有两个链表 a 和 b，设结点中包含学号、姓名。从 a 链表中删去与 b 链表中有相同学号的那些结点。

运行截图:

代码:

6.9 链表按指定方法删除节点

题目描述:

建立一个链表，每个结点包括：学号、姓名、性别、年龄。输入一个年龄，如果链表中的结点所包含的年龄等于此年龄，则将此结点删去。

运行截图:

代码:

6.10 环形链表的操作

题目描述:

操作环形链表

运行截图:

代码:

7. 栈、队列、堆

7.1 Bingo 想要梦奇的微信号

题目描述:

Bingo 刚认识了梦奇老师，他想要梦奇老师的微信号，梦奇老师不想直接告诉他，所以给了 Bingo 一串加密了的数字，并且把解密规则告诉了 Bingo。

解密规则是：首先删除第 1 个数，接着把第 2 个数放在这串数的最后面，再删除第 3 个数，并把第 4 个数放在这串数的最后面.....直至只剩最后一个数，把最后一个数也删除。

按照删除的顺序，把这些数字连在一起就是梦奇的微信号。请你按照解密规则帮 Bingo 得到梦奇老师的微信号。

输入描述:

第一行包括一个正整数 n ($1 < n < 500$)，表示这串微信号的长度；

第二行包括 n 个数字，即加密的小红的微信号。

输出描述:

输出解密后的微信号，相邻数字之间有空格。

示例:

输入:

```
9
1 2 3 4 5 6 7 8 9
```

输出:

```
1 3 5 7 9 4 8 6 2
```

运行截图:

代码:

7.2 有谁比我大

题目描述:

给定一个含有 n 个整数的数列 a_1, a_2, \dots, a_n 。定义函数 $f(a_i)$ 表示数列中第 i 个元素 a_i 之后第一个大于 a_i 的元素的下标，若这样的元素不存在，则 $f(a_i)=0$ 。

输入描述:

第一行包含一个正整数 n ($n \leq 1e6$)；

第二行包含 n 个正整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 1e9$)

输出描述:

输出仅一行包含 n 个整数, 分别代表 $f(a_i)$ 的值。

示例:

输入:

```
5
1 4 2 3 5
```

输出:

```
2 5 4 5 0
```

运行截图:

代码:

7.3 有序序列插入整数

题目描述:

有一个有序数字序列, 从小到大排序, 将一个新输入的数插入到序列中, 保证插入新数后, 序列仍然是升序。

输入描述:

第一行输入一个整数 N ($0 \leq N \leq 50$)。

第二行输入 N 个升序排列的整数, 输入用空格分隔的 N 个整数。

第三行输入想要进行插入的一个整数。

输出描述:

输出为一行, $N+1$ 个有序排列的整数。

示例:

输入:

```
5
1 6 9 22 30
8
```

输出:

```
1 6 8 9 22 30
```

运行截图:

代码:

8. 树、图

8.1 二叉树

题目描述:

有一个有序数字序列，从小到大排序，将一个新输入的数插入到序列中，保证插入新数后，序列仍然是升序。

输入描述:

第一行输入一个整数 N ($0 \leq N \leq 50$)。

第二行输入 N 个升序排列的整数，输入用空格分隔的 N 个整数。

第三行输入想要进行插入的一个整数。

输出描述:

输出为一行， $N+1$ 个有序排列的整数。

示例:

输入:

```
5
```

```
1 6 9 22 30
```

```
8
```

输出:

```
1 6 8 9 22 30
```

运行截图：

代码：

8.2 图

题目描述：

有一个有序数字序列，从小到大排序，将一个新输入的数插入到序列中，保证插入新数后，序列仍然是升序。

输入描述：

第一行输入一个整数 N ($0 \leq N \leq 50$)。

第二行输入 N 个升序排列的整数，输入用空格分隔的 N 个整数。

第三行输入想要进行插入的一个整数。

输出描述：

输出为一行， $N+1$ 个有序排列的整数。

示例：

输入：

5

1 6 9 22 30

8

输出：

1 6 8 9 22 30

运行截图：

代码：

9. 查找算法和排序算法

9.1 二分查找

题目描述：

有一个有序数字序列，从小到大排序，将一个新输入的数插入到序列中，保证插入新数后，序列仍然是升序。

输入描述：

第一行输入一个整数 N ($0 \leq N \leq 50$)。

第二行输入 N 个升序排列的整数，输入用空格分隔的 N 个整数。

第三行输入想要进行插入的一个整数。

输出描述：

输出为一行， $N+1$ 个有序排列的整数。

示例：

输入：

5

1 6 9 22 30

8

输出：

1 6 8 9 22 30

运行截图：

代码：

9.2 顺序查找

题目描述：

有一个有序数字序列，从小到大排序，将一个新输入的数插入到序列中，保证插入新数后，序列仍然是升序。

输入描述：

第一行输入一个整数 N ($0 \leq N \leq 50$)。

第二行输入 N 个升序排列的整数，输入用空格分隔的 N 个整数。

第三行输入想要进行插入的一个整数。

输出描述：

输出为一行， $N+1$ 个有序排列的整数。

示例：

输入：

5

1 6 9 22 30

8

输出：

1 6 8 9 22 30

运行截图：

代码：

9.3 哈希查找

题目描述：

有一个有序数字序列，从小到大排序，将一个新输入的数插入到序列中，保证插入新数后，序列仍然是升序。

输入描述：

第一行输入一个整数 N ($0 \leq N \leq 50$)。

第二行输入 N 个升序排列的整数，输入用空格分隔的 N 个整数。

第三行输入想要进行插入的一个整数。

输出描述：

输出为一行， $N+1$ 个有序排列的整数。

示例：

输入：

5

1 6 9 22 30

8

输出：

1 6 8 9 22 30

运行截图：

代码：

9.4 分块查找

题目描述：

有一个有序数字序列，从小到大排序，将一个新输入的数插入到序列中，保证插入新数后，序列仍然是升序。

输入描述：

第一行输入一个整数 N ($0 \leq N \leq 50$)。

第二行输入 N 个升序排列的整数，输入用空格分隔的 N 个整数。

第三行输入想要进行插入的一个整数。

输出描述：

输出为一行， $N+1$ 个有序排列的整数。

示例：

输入：

5

1 6 9 22 30

8

输出:

```
1 6 8 9 22 30
```

运行截图:

代码:

9.5 冒泡排序

题目描述:

有一个有序数字序列，从小到大排序，将一个新输入的数插入到序列中，保证插入新数后，序列仍然是升序。

输入描述:

第一行输入一个整数 N ($0 \leq N \leq 50$)。

第二行输入 N 个升序排列的整数，输入用空格分隔的 N 个整数。

第三行输入想要进行插入的一个整数。

输出描述:

输出为一行， $N+1$ 个有序排列的整数。

示例:

输入:

```
5
```

```
1 6 9 22 30
```

```
8
```

输出:

```
1 6 8 9 22 30
```

运行截图:

代码:

9.6 选择排序

题目描述:

有一个有序数字序列，从小到大排序，将一个新输入的数插入到序列中，保证插入新数后，序列仍然是升序。

输入描述:

第一行输入一个整数 N ($0 \leq N \leq 50$)。

第二行输入 N 个升序排列的整数，输入用空格分隔的 N 个整数。

第三行输入想要进行插入的一个整数。

输出描述:

输出为一行， $N+1$ 个有序排列的整数。

示例:

输入:

5

1 6 9 22 30

8

输出:

1 6 8 9 22 30

运行截图:

代码:

9.7 插入排序

题目描述:

有一个有序数字序列，从小到大排序，将一个新输入的数插入到序列中，保证插入新数后，序列仍然是升序。

输入描述：

第一行输入一个整数 N ($0 \leq N \leq 50$)。

第二行输入 N 个升序排列的整数，输入用空格分隔的 N 个整数。

第三行输入想要进行插入的一个整数。

输出描述：

输出为一行， $N+1$ 个有序排列的整数。

示例：

输入：

5

1 6 9 22 30

8

输出：

1 6 8 9 22 30

运行截图：

代码：

9.8 希尔排序

题目描述：

有一个有序数字序列，从小到大排序，将一个新输入的数插入到序列中，保证插入新数后，序列仍然是升序。

输入描述：

第一行输入一个整数 N ($0 \leq N \leq 50$)。

第二行输入 N 个升序排列的整数，输入用空格分隔的 N 个整数。

第三行输入想要进行插入的一个整数。

输出描述:

输出为一行, $N+1$ 个有序排列的整数。

示例:

输入:

```
5
1 6 9 22 30
8
```

输出:

```
1 6 8 9 22 30
```

运行截图:

代码:

9.9 归并排序

题目描述:

有一个有序数字序列, 从小到大排序, 将一个新输入的数插入到序列中, 保证插入新数后, 序列仍然是升序。

输入描述:

```
第一行输入一个整数  $N$  ( $0 \leq N \leq 50$ )。
第二行输入  $N$  个升序排列的整数, 输入用空格分隔的  $N$  个整数。
第三行输入想要进行插入的一个整数。
```

输出描述:

输出为一行, $N+1$ 个有序排列的整数。

示例:

输入:

```
5
```

1 6 9 22 30

8

输出:

1 6 8 9 22 30

运行截图:

代码:

9.10 快速排序

题目描述:

有一个有序数字序列，从小到大排序，将一个新输入的数插入到序列中，保证插入新数后，序列仍然是升序。

输入描述:

第一行输入一个整数 N ($0 \leq N \leq 50$)。

第二行输入 N 个升序排列的整数，输入用空格分隔的 N 个整数。

第三行输入想要进行插入的一个整数。

输出描述:

输出为一行， $N+1$ 个有序排列的整数。

示例:

输入:

5

1 6 9 22 30

8

输出:

1 6 8 9 22 30

运行截图:

代码:

个人资料请勿外传