

C 语言——经典 200 道实例【基础例题 100 道——进阶例题 100 道】

目录:

=====基础例题 100 道=====

C 语言——经典 200 道实例.....	1
1. 输出 "Hello, World!".....	1
2. 输出整数.....	1
3. 输出单个字符.....	2
4. 输出浮点数.....	2
5. 输出双精度 (double) 数.....	2
6. 两个整数相加.....	3
7. 两个浮点数相乘.....	4
8. 字符转 ASCII 码.....	4
(1) 一次读取长字符, 并一个一个显示 ASCII 码.....	5
(2) ASCII 转 字符.....	5
9. 两数相除.....	6
10. 数值比较.....	7
(1) 比较两个数.....	7
(2) 比较三个数.....	8
(3) 比较两数从键盘输入:	8
11. 计算字节大小.....	9
(1) 计算 int, float, double 和 char 字节大小.....	9
(2) 计算 long long, long double 字节大小.....	10
12. 交换两个数的值.....	10
(1) 使用临时变量.....	10
(2) 不使用临时变量.....	11
(3) 参考方法:	12
13. 判断奇数/偶数.....	12
参考方法:	13
14. 循环区间范围内的奇数/偶数.....	13
(1) 循环输出指定区间范围的偶数.....	14
(2) 循环输出指定区间范围的奇数.....	14
(3) 使用按位与运算符判断奇偶数:.....	15
(4) 改写为设定区间和除数, 返回可以整除的数.....	15
15. 判断元音/辅音.....	15
(1) 从键盘中输入的不是字母.....	16
(2) 参考方法:	18
(3) 多个字符输入错误的问题:.....	18
16. 判断三个数中的最大数.....	20
(1) 参考方法:	21
(2) 用三元表达式判断:.....	21

(3) 参考方法:	22
17. 一元二次方程.....	22
输入的三个系数 a、b、c 的判断解的情况:	23
18. 判断闰年.....	24
(1) 方法二.....	25
(2) 方法三.....	26
19. 判断正数/负数.....	27
方法二.....	27
20. 判断字母.....	28
方法二: 三元运算符.....	28
21. 计算自然数的和.....	29
(1) 使用 for.....	29
(2) 使用 while.....	29
(3) 使用递归.....	30
(4) 首尾相加.....	30
22. 输出九九乘法口诀表.....	31
方法二: 递归实现方式.....	31
23. 斐波那契数列.....	32
(1) 方法一: 输出指定数量的斐波那契数列.....	32
(2) 方法二: 输出指定数字前的斐波那契数列.....	33
24. 求两数的最大公约数.....	34
(1) 方法一: 使用 for 和 if.....	34
(2) 方法二: 使用 while 和 if.....	34
(3) 方法三: 适用正数和负数.....	35
(4) 方法四: 使用递归.....	36
(5) 方法五: 用辗转相除法.....	36
(6) 求两个数的最大公约数:.....	37
25. 求两数最小公倍数.....	37
(1) 方法一: 使用 while 和 if.....	37
(2) 方法二: 通过最大公约数计算.....	38
(3) 实例编写:.....	39
26. 阶乘.....	39
(1) 方法一:	39
(2) 方法二: 使用递归.....	40
27. 循环输出 26 个字母.....	41
(1) 循环输出 26 个字母.....	41
(2) 输出大写或小写字母.....	41
(3) 参考:	42
(4) 参考方法:	42
28. 判断数字为几位数.....	43
参考方法:	44
29. 计算一个数的 n 次方.....	44
(1) 方法一: 使用 while.....	44
(2) 方法二: 使用 pow() 函数.....	45

(3)方法三：递归.....	45
(4)参考方法：.....	46
30. 判断回文数.....	47
参考方法：.....	48
31. 判断素数.....	48
(1)	48
(2) 判断两个数之间的素数.....	49
(3) 使用函数判断两数间的素数.....	50
(4) 用 sqrt 降低时间复杂度.....	51
32. 判断 Armstrong 数(阿姆斯壮数).....	52
(1)	52
(2) 两数之间的 Armstrong 数.....	52
(3) 使用函数判断 Armstrong 数.....	54
(4) 参考方法：.....	55
(5) 两数之间的 Armstrong 数：.....	57
33. 求一个整数的所有因数.....	58
寻找两数的所有公因数：.....	59
34. 创建各类三角形图案.....	59
(1) 创建三角形图案.....	59
(2) 使用数字.....	60
(3) 使用字母.....	61
(4) 倒三角形.....	61
(5) 倒三角形数字.....	62
(6) 金字塔.....	63
(7) 数字金字塔.....	64
(8) 倒金字塔.....	65
(9) 杨辉三角.....	65
(10) 弗洛伊德三角形.....	66
(11) 数字金字塔 1：.....	67
(12) 数字金字塔 2：.....	67
35. 表格形式输出数据.....	68
(1) 将 1~100 的数据以 10x10 矩阵格式输出.....	69
(2) 等差数列输出 10x10 矩阵格式.....	69
(3) 乘法运算格式.....	70
(4) 将 1~100 的数据以 10x10 格式顺序输出.....	71
36. 实现简单的计算器.....	71
(1) 实现加减乘除计算.....	71
(2) 豪华版计算器.....	72
37. 计算一个数是否可为两个素数之和.....	75
38. 二进制与十进制相互转换.....	76
(1) 二进制转换为十进制.....	76
(2) 十进制转换为二进制.....	77
(3) 参考方法：.....	78
39. 八进制与十进制相互转换.....	79

(1) 十进制转换为八进制.....	79
(2) 八进制转换为十进制.....	80
(3) 任意进制转换.....	80
40. 八进制与二进制相互转换.....	81
(1) 二进制转换为八进制.....	81
(2) 八进制转换为二进制.....	82
(3) 分别定义函数实现八进制转二进制.....	83
41. 字符串翻转.....	84
(1) 字符串翻转.....	85
(2) 参考方法 1:	85
(3) 参考方法 2:	86
(4) 参考文档:	87
42. 计算数组元素平均值.....	87
(1) 实例 1.....	87
(2) 实例 2 用户自定义输入:	88
(3) 用 scanf 的特性来控制循环:	89
43. 输出数组.....	89
(1) 使用 for 循环输出数组:	89
(2) 使用 for 循环逆向输出数组:	90
44. 计算数组元素之和.....	90
45. 查找数组中最大的元素值.....	91
(1) 查找数组中最大的元素值.....	91
(2) 用户自定义输出:	91
(3) 三元运算符应用求最大:	92
46. 查找数组中最小的元素.....	92
47. 数组拆分与合并.....	93
(1) 将一个数组拆分为一个为奇数数组, 一个为偶数数组:	93
(2) 将奇数数组与偶数数组合并为一个数组:	94
48. 数组拷贝.....	95
(1) 将一个数组复制给另外一个数组:	95
(2) 值传递与地址传递的区别:	96
(3) 参考方法:	97
49. 计算标准偏差.....	98
50. 两个矩阵相加.....	99
(1) 使用多维数组将两个矩阵相加.....	99
(2) 矩阵相乘实例.....	101
51. 矩阵转换.....	103
52. 使用指针访问数组元素.....	104
53. 使用引用循环替换数值.....	105
54. 判断最大值.....	106
判断最大值.....	106
55. 删除字符串中的特殊字符.....	108
(1) 删除字符串中的除字母外的字符.....	108
(2) 参考方法:	109

56. 连接字符串.....	109
(1) 使用 strcat() 连接两个字符串.....	109
(2) 参考方法:	110
(3) 指针方法:	111
57. 计算字符串长度.....	111
(1) 使用 strlen().....	111
(2) 不使用 strlen().....	112
58. 查找字符在字符串中出现的次数.....	112
59. 字符串中各种字符计算.....	113
60. 字符串复制.....	114
(1) 使用 strcpy().....	114
(2) 不使用 strcpy().....	115
61. 字符串排序.....	116
62. 使用结构体 (struct)	117
(1) 使用结构体 (struct) 存储学生信息.....	117
(2) 两种不同对齐方式下计算 sizeof(struct).....	118
63. 复数相加.....	119
64. 计算两个时间段的差值.....	120
65. 将字符串写入文件.....	122
举例: 非常简单的字符串写入文件.....	122
66. 从文件中读取一行.....	124
(1) 从文件中读取一行.....	124
(2) 从文件中读取多行, 学会使用 fgets 函数 1.....	125
(3) 从文件中读取多行, 学会使用 fgets 函数 2.....	125
67. 输出当前文件执行代码.....	126
68. 约瑟夫生者死者小游戏.....	126
69. 五人分鱼.....	128
1. 有 1、2、3、4 个数字, 能组成多少个互不相同且无重复数字的三位数? 都是多少?	130
2. 企业发放的奖金根据利润提成.....	131
(1) 优化如下:.....	132
(2) 使用循环优化代码的适用性:	133
(3) 利用 switch 的击穿现象.....	134
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?	135
4. 输入某年某月某日, 判断这一天是这一年的第几天?	136
5. 输入三个整数 x, y, z, 请把这三个数由小到大输出.....	137
6. 用*号输出字母 C 的图案.....	138
7. 输出特殊图案, 请在 c 环境中运行, 看一看, Very Beautiful!.....	139
8. 输出 9*9 口诀.....	139
9. 要求输出国际象棋棋盘.....	140
10. 打印楼梯, 同时在楼梯上方打印两个笑脸.....	141
11. 古典问题 (兔子生崽)	142
12. 判断 101 到 200 之间的素数.....	143
13. 打印出所有的“水仙花数”.....	144

14. 将一个正整数分解质因数.....	145
15. 利用条件运算符的嵌套来完成此题：学习成绩 ≥ 90 分的同学用 A 表示，60-89 分之间的用 B 表示，60 分以下的用 C 表示.....	146
16. 最大公约数和最小公倍数.....	146
17. 输入一行字符，分别统计出其中英文字母、空格、数字和其它字符的个数.....	148
18.：求 $s=a+aa+aaa+aaaa+aa...a$ 的值.....	149
19. 一个数如果恰好等于它的因子之和，这个数就称为“完数”。例如 $6=1+2+3$ 。编程找出 1000 以内的所有完数.....	149
20. 小球自由下落.....	150
21. 猴子吃桃问题.....	151
22. 两个乒乓球队进行比赛，各出三人.....	152
23. 打印出如下图案（菱形）.....	153
24.：有一分数序列：2/1, 3/2, 5/3, 8/5, 13/8, 21/13... 求出这个数列的前 20 项之和.....	154
25. 求 $1+2!+3!+...+20!$ 的和.....	155
26. 利用递归方法求 5!.....	156
27. 利用递归函数调用方式，将所输入的 5 个字符，以相反顺序打印出来.....	156
28. 有 5 个人坐在一起，问第五个人多少岁？.....	157
29. 给一个不多于 5 位的正整数，要求：一、求它是几位数，二、逆序打印出各位数字.....	158
30. 回文数.....	159
31. 请输入星期几的第一个字母来判断一下是星期几，如果第一个字母一样，则继续判断第二个字母.....	160
32. 删除一个字符串中的指定字母，如：字符串“aca”，删除其中的 a 字母.....	161
33. 判断一个数字是否为质数.....	162
34. 练习函数调用.....	164
35. 字符串反转，如将字符串“www.runoob.com”反转为“moc.boonur.www”.....	165
36. 求 100 之内的素数.....	166
37. 对 10 个数进行排序.....	167
38. 求一个 3*3 矩阵对角线元素之和.....	168
39. 有一个已经排好序的数组。现输入一个数，要求按原来的规律将它插入数组中.....	169
40. 将一个数组逆序输出.....	170
41. 学习 static 定义静态变量的用法.....	171
42. 学习使用 auto 定义变量的用法.....	172
43. 学习使用 static 的另一用法.....	172
44. 学习使用 external 的用法.....	173
45. 学习使用 register 定义变量的方法.....	174
46. 宏#define 命令练习.....	174
47. 宏#define 命令练习 2.....	175
48. 宏#define 命令练习 3.....	176
49. #if #ifdef 和 #ifndef 的综合应用.....	176
50. #include 的应用练习.....	177
51. 学习使用按位与 &.....	178
52. 学习使用按位或 	179
53. 学习使用按位异或 ^.....	179

54. 取一个整数 a 从右端开始的 4~7 位	180
55. 学习使用按位取反~	181
56. 画图, 学用 circle 画圆形	181
57. 画图, 学用 line 画直线 (在 TC 中实现)	182
58. 学用 rectangle 画方形 (在 TC 中实现)	183
59. 画图, 综合例子 (在 TC 中实现)	183
60. 画图, 综合例子 2 (在 TC 中实现)	185
61. 杨辉三角形	186
62. 学习 putpixel 画点, (在 TC 中实现)	187
63. 画椭圆 ellipse (在 TC 中实现)	187
64. 利用 ellipse and rectangle 画图 (在 TC 中实现)	188
65. 一个最优美的图案 (在 TC 中实现)	189
66. 输入 3 个数 a, b, c, 按大小顺序输出	190
67. 输入数组, 最大的与第一个元素交换, 最小的与最后一个元素交换, 输出数组	191
68. 有 n 个整数, 使其前面各数顺序向后移 m 个位置, 最后 m 个数变成最前面的 m 个数	193
69. 有 n 个人围成一圈, 顺序排号。从第一个人开始报数 (从 1 到 3 报数), 凡报到 3 的人退出圈子, 问最后留下的是原来第几号的那位	194
70. 写一个函数, 求一个字符串的长度, 在 main 函数中输入字符串, 并输出其长度	196
71. 编写 input() 和 output() 函数输入, 输出 5 个学生的数据记录	196
72. 创建一个链表	198
73. 反向输出一个链表	199
74. 连接两个链表	200
75. 输入一个整数, 并将其反转后输出	202
76. 编写一个函数, 输入 n 为偶数时, 调用函数求 $1/2+1/4+\dots+1/n$, 当输入 n 为奇数时, 调用函数 $1/1+1/3+\dots+1/n$ (利用指针函数)	203
77. 填空练习 (指向指针的指针)	204
78. 找到年龄最大的人, 并输出。请找出程序中有什么问题	205
79. 字符串排序	206
80. 海滩上原来最少有多少个桃子?	207
81. $809*??=800*??+9*??$ 其中??代表的两位数, $809*??$ 为四位数, $8*??$ 的结果为两位数, $9*??$ 的结果为 3 位数。求??代表的两位数, 及 $809*??$ 后的结果	208
82. 八进制转换为十进制	209
83. 求 0—7 所能组成的奇数个数	209
84. 一个偶数总能表示为两个素数之和	211
85. 判断一个素数能被几个 9 整除	212
86. 两个字符串连接程序	212
87. 回答结果 (结构体变量传递)	213
88. 读取 7 个数 (1—50) 的整数值, 每读取一个值, 程序打印出该值个数的 *	214
89. 电话号码加密	215
90. 专升本一题, 读结果	216
91. 时间函数举例 1	216
92. 时间函数举例 2	217
93. 时间函数举例 3	218

94. 猜谜游戏.....	218
95. 简单的结构体应用实例.....	222
96. 计算字符串中子串出现的次数.....	222
97. 从键盘输入一些字符，逐个把它们送到磁盘上去，直到输入一个#为止.....	223
98. 从键盘输入一个字符串，将小写字母全部转换成大写字母，然后输出到一个磁盘文件“test”中保存。输入的字符串以！结束.....	224
99. 有两个磁盘文件 A 和 B, 各存放一行字母，要求把这两个文件中的信息合并（按字母顺序排列），输出到一个新文件 C 中.....	225
100. 计算平均成绩.....	227
101. 矩阵交换行.....	228
描述：	228
102. 猴子吃桃问题.....	230

1. 输出 "Hello, World!"

```
#include <stdio.h>
int main()
{
    // printf() 中字符串需要引号
    printf("Hello, World!");
    return 0;
}
```

输出结果:

Hello, World!

2. 输出整数

```
#include <stdio.h>
int main()
{
    int number;

    // printf() 输出字符串
    printf("输入一个整数: ");

    // scanf() 格式化输入
    scanf("%d", &number);

    // printf() 显示格式化输入
    printf("你输入的整数是: %d", number);
    return 0;
}
```

使用 printf() 与 %d 格式化输出整数

输出结果:

输入一个整数: 45
你输入的整数是: 45

3. 输出单个字符

```
#include <stdio.h>

int main() {
    char c;          // 声明 char 变量

    c = 'A';         // 定义 char 变量

    printf("c 的值为 %c", c);

    return 0;
}
```

使用 `printf()` 与 `%c` 格式化输出一个字符

输出结果：

c 的值为 A

4. 输出浮点数

```
#include <stdio.h>

int main() {
    float f;          // 声明浮点数变量

    f = 12.001234;     // 定义浮点数变量

    printf("f 的值为 %f", f);

    return 0;
}
```

使用 `printf()` 与 `%f` 输出浮点数

输出结果：

f 的值为 12.001234

5. 输出双精度（double）数

```
#include <stdio.h>

int main() {
    double d;           // 声明双精度变量

    d = 12.001234;      // 定义双精度变量

    printf("d 的值为 %le", d);

    return 0;
}
```

使用 `printf()` 与 `%e` 输出双精度数

输出结果:

d 的值为 1.200123e+01

6. 两个整数相加

```
#include <stdio.h>
int main()
{
    int firstNumber, secondNumber, sumOfTwoNumbers;

    printf("输入两个数(以空格分割): ");

    // 通过 scanf() 函数接收用户输入的两个整数
    scanf("%d %d", &firstNumber, &secondNumber);

    // 两个数字相加
    sumOfTwoNumbers = firstNumber + secondNumber;

    // 输出结果
    printf("%d + %d = %d", firstNumber, secondNumber, sumOfTwoNumbers);

    return 0;
}
```

使用 `scanf()` 来接收输入，`printf()` 与 `%d` 格式化输出整数

输出结果:

输入两个数(以空格分割): 1 2

1 + 2 = 3

7. 两个浮点数相乘

```
#include <stdio.h>
int main()
{
    double firstNumber, secondNumber, product;
    printf("输入两个浮点数: ");

    // 用户输入两个浮点数
    scanf("%lf %lf", &firstNumber, &secondNumber);

    // 两个浮点数相乘
    product = firstNumber * secondNumber;

    // 输出结果, %.2lf 保留两个小数点
    printf("结果 = %.2lf", product);

    return 0;
}
```

输入两个浮点数, 计算乘积

输出结果:

输入两个浮点数: 1.2 2.345
结果 = 2.81

8. 字符转 ASCII 码

<https://blog.csdn.net/liul7234050/article/details/104219239>

```
#include <stdio.h>
int main()
{
    char c;
    printf("输入一个字符: ");

    // 读取用户输入
    scanf("%c", &c);

    // %d 显示整数
```

```

    // %c 显示对应字符
    printf("%c 的 ASCII 为 %d", c, c);
    return 0;
}

```

输出结果：

输入一个字符：a
a 的 ASCII 为 97

(1) 一次读取长字符，并一个一个显示 ASCII 码

```

#include <stdio.h>
#define MAX_STRING_LENGTH 65535 // 最大字符串长度

int main() {
    char s[MAX_STRING_LENGTH];
    printf("请输入长度小于 %d 的任意字符：", MAX_STRING_LENGTH);
    scanf("%s", s); // 读取字符串。
    for(int i = 0; s[i]; i++) {
        printf("%c 的 ASCII:%d\t", s[i], s[i]);
    }
}

```

(2) ASCII 转 字符

```

#include <stdio.h>
#define MAX_ASCII 127

int main()
{
    char num, enter;
    int temp=1;
    for(;temp>0;)
    {
        printf("-----\n");
        printf("|**      开始          **|\n");
        printf("|**ASCII  转  字符  按:1 **|\n");
        printf("|**字符   转  ASCII 按:2 **|\n");
        printf("|**      结束          按:0 **|\n");
        printf("-----\n");
        scanf("%d",&temp);
        if(temp==1)

```

```

    {
        printf("请输入数值小于 %d 的任意字符: ", MAX_ASCII);
        scanf("%d", &num);
        printf("ASCII 为 %d , 对应的字符为 %c \n", num, num);
    }
    if(temp==2)
    {
        printf("输入一个字符: \n");
        scanf("%c", &enter);          //回车键也算字符, 所以这里使用其
他变量替之.
        scanf("%c", &num);
        printf("    %c 的 ASCII 为 %d    \n", num, num);
    }
}
return 0;
}

```

9. 两数相除

两数相除，如果有余数，输出余数

```

#include <stdio.h>

int main() {

    int dividend, divisor, quotient, remainder;

    printf("输入被除数: ");
    scanf("%d", &dividend);

    printf("输入除数: ");
    scanf("%d", &divisor);

    // 计算商
    quotient = dividend / divisor;

    // 计算余数
    remainder = dividend % divisor;

    printf("商 = %d\n", quotient);
    printf("余数 = %d", remainder);

    return 0;
}

```

输出结果:

输入被除数: 5
输入除数: 2
商 = 2
余数 = 1

10. 数值比较

(1) 比较两个数

以下实例中定义了两个整数变量，并使用 `if` 来比较两个数值，可以先看下逻辑图：

```
#include <stdio.h>

int main() {
    int a, b;

    a = 11;
    b = 99;

    // 也可以通过以下代码实现让用户在终端输入两个数
    // printf("输入第一个值:");
    // scanf("%d", &a);
    // printf("输入第二个值:");
    // scanf("%d", &b);

    if(a > b)
        printf("a 大于 b");
    else
        printf("a 小于等于 b");

    return 0;
}
```

输出结果:

a 小于等于 b

(2) 比较三个数

以下实例中定义了两个整数变量，并使用 if 来比较三个数值，可以先看下逻辑图：

```
#include <stdio.h>

int main() {
    int a, b, c;

    a = 11;
    b = 22;
    c = 33;

    if ( a > b && a > c )
        printf("%d 最大", a);
    else if ( b > a && b > c )
        printf("%d 最大", b);
    else if ( c > a && c > b )
        printf("%d 最大", c);
    else
        printf("有两个或三个数值相等");

    return 0;
}
```

输出结果：

33 最大

(3) 比较两数从键盘输入：

```
#include <stdio.h>

int main()
{
    int a,b;
    printf("请输入两个整数以空格隔开: \n");
    scanf("%d %d",&a,&b);    // 从键盘输入两个数
    if(a>b){
```



```

        printf("%d>%d\n", a, b);
    }
    else
    {
        printf("%d<%d\n", a, b);
    }
}

```

11. 计算字节大小

(1) 计算 int, float, double 和 char 字节大小

使用 **sizeof 操作符** 计算 int, float, double 和 char 四种变量字节大小

sizeof 是 C 语言的一种**单目操作符**，如 C 语言的**其他操作符**++、--等，**它并不是函数**

sizeof 操作符以**字节形式**给出了其操作数的**存储大小**

```

#include <stdio.h>

int main()
{
    int integerType;
    float floatType;
    double doubleType;
    char charType;

    // sizeof 操作符用于计算变量的字节大小
    printf("Size of int: %ld bytes\n", sizeof(integerType));
    printf("Size of float: %ld bytes\n", sizeof(floatType));
    printf("Size of double: %ld bytes\n", sizeof(doubleType));
    printf("Size of char: %ld byte\n", sizeof(charType));

    return 0;
}

```

输出结果：

```

Size of int: 4 bytes
Size of float: 4 bytes
Size of double: 8 bytes
Size of char: 1 byte

```

(2) 计算 long long, long double 字节大小

```
#include <stdio.h>
int main()
{
    int a;
    long b;
    long long c;

    double e;
    long double f;

    printf("Size of int = %ld bytes \n", sizeof(a));
    printf("Size of long = %ld bytes\n", sizeof(b));
    printf("Size of long long = %ld bytes\n", sizeof(c));

    printf("Size of double = %ld bytes\n", sizeof(e));
    printf("Size of long double = %ld bytes\n", sizeof(f));

    return 0;
}
```

输出结果:

```
Size of int = 4 bytes
Size of long = 8 bytes
Size of long long = 8 bytes
Size of double = 8 bytes
Size of long double = 16 bytes
```

12. 交换两个数的值

(1) 使用临时变量

以下实例演示了交换两个浮点数的值

```
#include <stdio.h>

int main()
{
    double firstNumber, secondNumber, temporaryVariable;
```

```

printf("输入第一个数字: ");
scanf("%lf", &firstNumber);

printf("输入第二个数字: ");
scanf("%lf", &secondNumber);

// 将第一个数的值赋值给 temporaryVariable
temporaryVariable = firstNumber;

// 第二个数的值赋值给 firstNumber
firstNumber = secondNumber;

// 将 temporaryVariable 赋值给 secondNumber
secondNumber = temporaryVariable;

printf("\n 交换后, firstNumber = %.2lf\n", firstNumber);
printf("交换后, secondNumber = %.2lf", secondNumber);

return 0;
}

```

输出结果:

输入第一个数字: 1
输入第二个数字: 2

交换后, firstNumber = 2.00
交换后, secondNumber = 1.00

(2) 不使用临时变量

不使用临时变量交换两个整数的值

```

#include <stdio.h>

int main() {
    int a, b;

    a = 11;
    b = 99;

    printf("交换之前 - \n a = %d, b = %d \n\n", a, b);
}

```

```

a = a + b;  // ( 11 + 99 = 110) 此时 a 的变量为两数之和, b 未改变
b = a - b;  // ( 110 - 99 = 11)
a = a - b;  // ( 110 - 11 = 99)

printf("交换后 - \n a = %d, b = %d \n", a, b);
}

```

输出结果:

交换之前 -
a = 11, b = 99

交换后 -
a = 99, b = 11

(3) 参考方法:

```

#include<stdio.h>

void swap(int *a,int*b)
{
    int temp=*a;
    *a=*b;
    *b=temp;
}

int main()
{
    int a=1,b=2;
    swap(&a,&b);
    printf("a=%d\nb=%d\n",a,b);
    return 0;
}

```

13. 判断奇数/偶数

以下实例判断用户输入的整数是奇数还是偶数

```

#include <stdio.h>

int main()
{
    int number;

```

```

printf("请输入一个整数: ");
scanf("%d", &number);

// 判断这个数除以 2 的余数
if(number % 2 == 0)
    printf("%d 是偶数。", number);
else
    printf("%d 是奇数。", number);

return 0;
}

```

输出结果:

请输入一个整数: 5
5 是奇数。

参考方法:

奇偶数判断其实有个更简单高效的办法，我们的整数，在计算机中存储的**都是二进制**

奇数的**最后一位**必是**1**，所以我们可以这样写：

```

#include <stdio.h>

int main()
{
    int number;

    printf("请输入一个整数: ");
    scanf("%d", &number);

    // 判断这个数最后一位是 1 这为奇数
    if(number&1)
        printf("%d 是奇数。", number);
    else
        printf("%d 是偶数。", number);

    return 0;
}

```

14. 循环区间范围内的奇数/偶数

循环输出区间范围内的奇数/偶数可以通过除于 2 的余数来判断

(1) 循环输出指定区间范围的偶数

```
#include <stdio.h>

int main() {
    int i;

    for(i = 1; i <= 10; i++) {
        if(i%2 == 0)
            printf(" %2d\n", i);
    }
    return 0;
}
```

输出结果:

```
2
4
6
8
10
```

(2) 循环输出指定区间范围的奇数

```
#include <stdio.h>

int main() {
    int i;

    for(i = 1; i <= 10; i++) {
        if(i%2 != 0)
            printf("%d\n", i);
    }
    return 0;
}
```

输出结果:

```
1
3
5
```

7
9

(3) 使用按位与运算符判断奇偶数:

```
#include <stdio.h>
int main() {
    for (int i=1; i<10; i++)
        // 通过按位与运算符判断奇偶数
        i & 1 ? printf("奇数: %d\n", i) : printf("偶数: %d\n", i);
}
```

(4) 改写为设定区间和除数，返回可以整除的数

```
#include <stdio.h>
int main()
{
    // 定义自变量和起始数字及除数
    int i, start, end, divisor;
    printf("Please input a starting number:");
    scanf("%d", &start);
    printf("Please input a end number:");
    scanf("%d", &end);
    printf("Please input a integer as the divisor:");
    scanf("%d", &divisor);
    // 返回给定区间能被除数整除的数字
    for (i = start; i <= end; i++)
    {
        if (i % divisor == 0)
            printf("%d\n", i);
    }
    return 0;
}
```

15. 判断元音/辅音

判断输入的字母是元音，还是辅音

英语有 26 个字母，元音只包括 a、e、i、o、u 这五个字母，其余的都为辅音

y 是半元音、半辅音字母，但在英语中都把他当作辅音

```

#include <stdio.h>

int main()
{
    char c;
    int isLowercaseVowel, isUppercaseVowel;

    printf("输入一个字母: ");
    scanf("%c", &c);

    // 小写字母元音
    isLowercaseVowel = (c == 'a' || c == 'e' || c == 'i' || c == 'o' ||
c == 'u');

    // 大写字母元音
    isUppercaseVowel = (c == 'A' || c == 'E' || c == 'I' || c == 'O' ||
c == 'U');

    // if 语句判断
    if (isLowercaseVowel || isUppercaseVowel)
        printf("%c 是元音", c);
    else
        printf("%c 是辅音", c);
    return 0;
}

```

输出结果:

输入一个字母: G
G 是辅音

(1) 从键盘中输入的不是字母

如果从键盘中输入的不是字母，例如，数字，标点符号，都会输出是辅音。所以我对这个做了一个简单的判定

```

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

int main()
{
    char cLetter;

```



```

    int iInRange, iCheck, i, iLetter;
    i = 1;
    while (i)
    {
        printf("输入一个字母, 判断元音与辅音. 如果想退出, 请输入“0”。\n");
        scanf("%c", &cLetter);
        getchar();

        printf("-----\n");
        iLetter = (int)cLetter;
        iCheck = isalpha(iLetter);
        if (iCheck)
        {
            iInRange = (cLetter == 'a') || (cLetter == 'e') || (cLetter == 'i') || (cLetter == 'o') || (cLetter == 'u') || (cLetter == 'A') || (cLetter == 'E') || (cLetter == 'I') || (cLetter == 'O') || (cLetter == 'U');

            if (iInRange)
            {
                printf("字母 %c 是元音字母!\n", cLetter);

                printf("-----\n");
                continue;
            }
            else
            {
                printf("字母 %c 是辅音字母!\n", cLetter);

                printf("-----\n");
                continue;
            }
        }
        else if (iCheck == 0)
        {
            if (iLetter != 48)
            {
                printf("Error input!\n");

                printf("-----\n");
            }
        }
    }

```

```

        continue;
    }
    else if ((int)cLetter == 48)
    {
        printf("Bye bye~~~!\n");

printf("-----\n");
        i = (int)cLetter - 48;
    }
}
return 0;
}

```

(2) 参考方法:

```

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

int main() {
    char c;
    printf("请输入一个字母: ");
    scanf("%c", &c);
    if ((c>='A' &&c<='Z') || (c>='a' &&c<='z'))
        if (c=='A' || c=='E' || c=='I' || c
== 'O' || c=='U' || c=='a' || c=='e' || c=='i' || c == 'o' || c=='u')
            printf("%c 为元音", c);
        else
            printf("%c 为辅音", c);
    else
        printf("Error input!");
    return 0;
}

```

(3) 多个字符输入错误的问题:

```

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

int main()

```

```

{
    char cLetter, b, a;
    int iInRange, iCheck, i, iLetter;
    i = 1;
    while (i)
    {
        printf("输入一个字母, 判断元音与辅音. 如果想退出, 请输入“0”。\n");
        scanf("%c", &a);
        cLetter=a;
        scanf("%c", &a);

        while(a!=10 )
        {

            scanf("%c", &a);
        }
        //getchar();//getchar is must get a key input and CR

        printf("-----\n");
        iLetter = (int)cLetter;
        iCheck = isalpha(iLetter);
        if (iCheck)
        {
            iInRange = (cLetter == 'a') || (cLetter == 'e') || (cLetter == 'i') || (cLetter == 'o') || (cLetter == 'u') || (cLetter == 'A') || (cLetter == 'E') || (cLetter == 'I') || (cLetter == 'O') || (cLetter == 'U');

            if (iInRange)
            {
                printf("字母 %c 是元音字母!\n", cLetter);

                printf("-----\n");
                continue;
            }
            else
            {
                printf("字母 %c 是辅音字母!\n", cLetter);

                printf("-----\n");
                continue;
            }
        }
    }
}

```

```

    }
}
else if (iCheck == 0)
{
    if (iLetter != 48)
    {
        printf("Error input!\n");

printf("-----\n");
        continue;
    }
    else if ((int)cLetter == 48)
    {
        printf("Bye bye~~~!\n");

printf("-----\n");
        i = (int)cLetter - 48;
    }
}
}
return 0;
}

```

16. 判断三个数中的最大数

```

#include <stdio.h>

int main()
{
    double n1, n2, n3;

    printf("请输入三个数，以空格分隔：");
    scanf("%lf %lf %lf", &n1, &n2, &n3);

    if( n1>=n2 && n1>=n3 )
        printf("%.2f 是最大数。", n1);

    if( n2>=n1 && n2>=n3 )
        printf("%.2f 是最大数。", n2);

    if( n3>=n1 && n3>=n2 )

```

```

        printf("%.2f 是最大数。", n3);

    return 0;
}

```

输出结果:

请输入三个数，以空格分隔：1 2 3
3.00 是最大数。

(1) 参考方法:

```

#include<stdio.h>

int main()
{
    int a,b,c,max;
    printf("请输入三个数，用空格隔开:");
    scanf("%d %d %d",&a,&b,&c);
    if(a>b){
        max=a;
    } else {
        max=b;
    }
    if(max>c){
        printf("最大值是%d",max);
    } else{
        max=c;
        printf("最大值是%d",max);
    }
    return 0;
}

```

(2) 用三元表达式判断:

```

#include <stdio.h>

int main()
{
    int a, b, c, max;
    printf("请输入三个数，用空格分割: ");
    scanf("%d %d %d", &a, &b, &c);
    max = a > b ? ( a > c ? a : c ) : ( b > c ? b : c );
}

```

```

    printf("最大值是: %d", max);
    return 0;
}

```

(3) 参考方法:

```

#include <stdio.h>
/*
    判断三个数中的最大数
*/
int getMax(int a, int b, int c)
{
    int max=a;
    if(b>max)
    {
        max = b;
    }
    if(c>max)
    {
        max = c;
    }
    return max;
}

int main()
{
    int a, b, c;
    printf("请输入三个整数, 用空格分隔: ");
    scanf("%d %d %d",&a, &b, &c);
    printf("最大值是: %d",getMax(a, b, c));
    return 0;
}

```

17. 一元二次方程

求一元二次方程: $ax^2+bx+c=0$ 的根。

输入三个实数 a, b, c 的值, 且 a 不等于 0。

```

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

```

```

int main()
{
    float a, b, c, x1, x2, d;
    printf("输入方程的三个系数:");
    scanf("%f %f %f", &a, &b, &c);
    if(a!=0)
    {
        d=sqrt(b*b-4*a*c);
        x1=(-b+d)/(2*a);
        x2=(-b-d)/(2*a);
        if(x1<x2)
            printf("%.2f %.2f\n", x2, x1);
        else
            printf("%.2f %.2f\n", x1, x2);
    }
    return 0;
}

```

输出结果:

输入方程的三个系数:1 2 1
-1.00 -1.00

输入的三个系数 a、b、c 的判断解的情况:

一元二次方程可能有两个实数解，或者一个实数解，或者无实数解

```

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

```

```

int main(void)
{
    double a , b , c;
    double delat;
    double x1, x2;
    char ch;
    do
    {

```

```

        printf("请输入一元二次方程的三个系数: \n");

```

```

        printf("请输入系数 a = \t");

```

```

scanf("%lf",&a);

printf("请输入系数 b = \t");
scanf("%lf",&b);

printf("请输入系数 c = \t");
scanf("%lf",&c);

delat = b*b-4*a*c;

if(delat>0)
{
    x1= ( -b+sqrt(delat) )/2;
    x2= ( -b-sqrt(delat) )/2;
    printf("有 2 个实数解: x1 = %lf  x2 = %lf\n",x1,x2);
}
else if(delat==0)
{
    x1=( -b+sqrt(delat) )/2;
    printf("有 2 个相等实数解: x1=x2 = %lf\n",x1);
}
else
    printf("无实数解\n");

printf("是否继续吗: Y/N\n");
scanf(" %c",&ch);
}while(ch=='y' || ch=='Y');
return 0;
}

```

18. 判断闰年

用户输入年份，判断该年份是否为闰年

```

#include <stdio.h>

int main()
{
    int year;

    printf("输入年份: ");
    scanf("%d",&year);

    if(year%4 == 0)

```



```

{
    if( year%100 == 0)
    {
        // 这里如果被 400 整数是闰年
        if ( year%400 == 0)
            printf("%d 是闰年", year);
        else
            printf("%d 不是闰年", year);
    }
    else
        printf("%d 是闰年", year );
}
else
    printf("%d 不是闰年", year);

return 0;
}

```

输出结果:

输入年份: 1990
1990 不是闰

(1) 方法二

```

#include <stdio.h>

int main()
{
    int year;
    printf("输入年份: ");
    scanf("%d",&year);
    // year = 300;

    // 先处理百年, 余下的在处理非百年
    if(year % 100 == 0) // 先找出百年
    {
        if(year % 400 == 0) // 找出 400 年
        {
            printf("y\n");
        }
        else // 剩下就是是 100 年但不是 400 年
        {
            printf("n\n");
        }
    }
}

```

```

    }
}
else // 剩下的是非百年的年份按照正常走
{
    if(year % 4 == 0)
    {
        printf("y\n");
    }
    else
    {
        printf("n\n");
    }
}

return 0;
}

```

总体思路，先把特殊的百年年份拿出来，是 400 的倍数，是闰年，不是 400 的倍数，不是闰年。

然后，百年年份剩余的都为正常的非百年的年份直接按照 4 处理。

(2) 方法三

直接条件判断闰年

```

#include <stdio.h>

int main()
{
    int year;
    printf("输入年份: ");
    scanf("%d",&year);
    // year = 400;

    // (四年一闰，百年不闰) || 四百年在闰年
    if((year % 4 == 0 && year % 100 != 0) || year % 400 == 0)
    {
        printf("y\n");
    }
    else
    {
        printf("n\n");
    }
}

```

```
    return 0;
}
```

19. 判断正数/负数

用户输入一个数字，判断该数字是正数还是负数或是零

```
#include <stdio.h>
int main()
{
    double number;

    printf("输入一个数字: ");
    scanf("%lf", &number);

    if (number <= 0.0)
    {
        if (number == 0.0)
            printf("你输入的是 0。");
        else
            printf("你输入的是负数。");
    }
    else
        printf("你输入的是正数。");
    return 0;
}
```

输出结果:

输入一个数字: 9
你输入的是正数。

方法二

```
#include<stdio.h>

int main() {
    double number;
    printf("请输入一个数字: ");
    scanf("%lf", &number);
    if(number < 0.0)
        printf("你输入的是负数>_<\n");
}
```

```

    else if(number > 0.0)
        printf("你输入的是正数-.\n");
    else
        printf("你输入的是零 0.0\n");
}

```

20. 判断字母

用户输入一个字符，判断该字符是否为一个字母

```

#include <stdio.h>

int main()
{
    char c;
    printf("输入一个字符: ");
    scanf("%c",&c);

    if( (c>='a' && c<='z') || (c>='A' && c<='Z'))
        printf("%c 是字母",c);
    else
        printf("%c 不是字母",c);

    return 0;
}

```

输出结果:

输入一个字符: a
a 是字母

方法二：三元运算符

```

#include <stdio.h>

int main() {
    char c;
    printf("请输入一个字符: ");
    scanf("%c",&c);
    ((c>='a' && c<='z') || (c>='A' && c<='Z')) ? printf("%c 是字母",c) :
    printf("%c 不是字母",c);
    return 0;
}

```

21. 计算自然数的和

自然数是指表示物体个数的数，即由 0 开始，0, 1, 2, 3, 4, ……一个接一个，组成一个无穷的集体，即指非负整数

(1) 使用 for

```
#include <stdio.h>
int main()
{
    int n, i, sum = 0;

    printf("输入一个正整数: ");
    scanf("%d",&n);

    for(i=1; i <= n; ++i)
    {
        sum += i;    // sum = sum+i;
    }

    printf("Sum = %d",sum);

    return 0;
}
```

(2) 使用 while

```
#include <stdio.h>
int main()
{
    int n, i, sum = 0;

    printf("输入一个正整数: ");
    scanf("%d",&n);

    i = 1;
    while ( i <=n )
    {
        sum += i;
        ++i;
    }
}
```

```

        printf("Sum = %d", sum);

        return 0;
}

```

输出结果：

输入一个正整数：100
Sum = 5050

(3) 使用递归

```

#include <stdio.h>
int addNumbers(int n);

int main()
{
    int num;
    printf("输入一个整数: ");
    scanf("%d", &num);
    printf("Sum = %d", addNumbers(num));
    return 0;
}

int addNumbers(int n)
{
    if(n != 0)
        return n + addNumbers(n-1);
    else
        return n;
}

```

(4) 首尾相加

```

#include <stdio.h>
int main()
{
    int num;
    printf("请输入一个自然数: ");
    while (scanf("%d", &num) == 1)
    {
        printf("Sum = %d", (num + 1) * num / 2);
        break;
    }
}

```

```

    }
    return 0;
}

```

22. 输出九九乘法口诀表

使用嵌套 for 循环输出九九乘法口诀表

```

#include<stdio.h>

int main() {
    //外层循环变量, 控制行
    int i = 0;
    //内层循环变量, 控制列
    int j = 0;
    for(i=1; i<=9; i++) {
        for(j=1; j<=i; j++) {
            printf("%dx%d=%d\t", j, i, i*j);
        }
        //每行输出完后换行
        printf("\n");
    }
}

```

输出结果:

```

1x1=1
1x2=2    2x2=4
1x3=3    2x3=6    3x3=9
1x4=4    2x4=8    3x4=12    4x4=16
1x5=5    2x5=10    3x5=15    4x5=20    5x5=25
1x6=6    2x6=12    3x6=18    4x6=24    5x6=30    6x6=36
1x7=7    2x7=14    3x7=21    4x7=28    5x7=35    6x7=42    7x7=49
1x8=8    2x8=16    3x8=24    4x8=32    5x8=40    6x8=48    7x8=56
8x8=64
1x9=9    2x9=18    3x9=27    4x9=36    5x9=45    6x9=54    7x9=63
8x9=72    9x9=81

```

方法二：递归实现方式

```

#include <stdio.h>

void func(int i, int j)

```

```

{
    if(i>j)
        return;
    printf("%dx%d=%d\t", i, j, i*j);
    func(i+1, j);
}

void f(int n)
{
    if(n==1)
        printf("1x1=1\n");
    else
    {
        f(n-1);
        func(1, n);
        putchar('\n');
    }
}

int main()
{
    f(9);
    return 0;
}

```

23. 斐波那契数列

斐波那契数列指的是这样一个数列 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368.....

这个数列从第 3 项开始，每一项都等于前两项之和

(1) 方法一：输出指定数量的斐波那契数列

```

#include <stdio.h>

int main()
{
    int i, n, t1 = 0, t2 = 1, nextTerm;

    printf("输出几项: ");
    scanf("%d", &n);

```



```

printf("斐波那契数列: ");

for (i = 1; i <= n; ++i)
{
    printf("%d, ", t1);
    nextTerm = t1 + t2;
    t1 = t2;
    t2 = nextTerm;
}
return 0;
}

```

输出结果:

输出几项: 10

斐波那契数列: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34,

(2) 方法二: 输出指定数字前的斐波那契数列

```

#include <stdio.h>

int main()
{
    int t1 = 0, t2 = 1, nextTerm = 0, n;

    printf("输入一个正数: ");
    scanf("%d", &n);

    // 显示前两项
    printf("斐波那契数列: %d, %d, ", t1, t2);

    nextTerm = t1 + t2;

    while(nextTerm <= n)
    {
        printf("%d, ", nextTerm);
        t1 = t2;
        t2 = nextTerm;
        nextTerm = t1 + t2;
    }

    return 0;
}

```

输出结果:

输入一个正数: 100

斐波那契数列: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89,

24. 求两数的最大公约数

用户输入两个数，求这两个数的最大公约数

(1) 方法一：使用 for 和 if

```
#include <stdio.h>

int main()
{
    int n1, n2, i, gcd;

    printf("输入两个正整数，以空格分隔: ");
    scanf("%d %d", &n1, &n2);

    for(i=1; i <= n1 && i <= n2; ++i)
    {
        // 判断 i 是否为最大公约数
        if(n1%i==0 && n2%i==0)
            gcd = i;
    }

    printf("%d 和 %d 的最大公约数是 %d", n1, n2, gcd);

    return 0;
}
```

输出结果:

输入两个正整数，以空格分隔: 81 153

81 和 153 的最大公约数是 9

(2) 方法二：使用 while 和 if

```
#include <stdio.h>

int main()
{
```

```

int n1, n2;

printf("输入两个数，以空格分隔：");
scanf("%d %d", &n1, &n2);

while(n1!=n2)
{
    if(n1 > n2)
        n1 -= n2;
    else
        n2 -= n1;
}
printf("GCD = %d", n1);

return 0;
}

```

输出结果：

输入两个数，以空格分隔：81 153
GCD = 9

(3) 方法三：适用正数和负数

```

#include <stdio.h>

int main()
{
    int n1, n2;

    printf("输入两个数，以空格分隔：");
    scanf("%d %d", &n1, &n2);

    // 如果输入的是负数，将其转换为正数
    n1 = ( n1 > 0) ? n1 : -n1;
    n2 = ( n2 > 0) ? n2 : -n2;

    while(n1!=n2)
    {
        if(n1 > n2)
            n1 -= n2;
        else
            n2 -= n1;
    }
}

```

```

    printf("GCD = %d", n1);

    return 0;
}

```

输出结果:

输入两个数，以空格分隔：81 -153
GCD = 9

(4) 方法四：使用递归

```

#include <stdio.h>
int hcf(int n1, int n2);
int main()
{
    int n1, n2;
    printf("输入两个正整数: ");
    scanf("%d %d", &n1, &n2);

    printf("%d 和 %d 的最大公约数为 %d", n1, n2, hcf(n1, n2));
    return 0;
}

int hcf(int n1, int n2)
{
    if (n2 != 0)
        return hcf(n2, n1%n2);
    else
        return n1;
}

```

(5) 方法五：用辗转相除法

```

#include <stdio.h>

int main()
{
    int a, b;
    int t;
    scanf("%d %d", &a, &b);
    while (b != 0) {
        t = a%b;

```

```

        a = b;
        b = t;
        printf("a=%d b=%d t=%d\n", a, b, t);
    }
    printf("最大公约数是%d\n", a);
    return 0;
}

```

(6) 求两个数的最大公约数:

```

#include <stdio.h>

int main(void)
{
    int x, y, gcd;
    printf("请输入两个正整数: ");
    scanf("%d%d", &x, &y);
    if(x < y)
        gcd = x;
    else
        gcd = y;
    while(x % gcd || y % gcd)
        gcd--;
    printf("gcd = %d\n", gcd);

    return 0;
}

```

25. 求两数最小公倍数

用户输入两个数，求这两个数的最小公倍数

(1) 方法一：使用 while 和 if

```

#include <stdio.h>

int main()
{
    int n1, n2, minMultiple;
    printf("输入两个正整数: ");
    scanf("%d %d", &n1, &n2);

```

```

// 判断两数较大的值，并赋值给 minMultiple
minMultiple = (n1>n2) ? n1 : n2;

// 条件为 true
while(1)
{
    if( minMultiple%n1==0 && minMultiple%n2==0 )
    {
        printf("%d 和 %d 的最小公倍数为 %d", n1, n2,minMultiple);
        break;
    }
    ++minMultiple;
}
return 0;
}

```

输出结果：

输入两个正整数：72 120
72 和 120 的最小公倍数为 360

(2) 方法二：通过最大公约数计算

```

#include <stdio.h>

int main()
{
    int n1, n2, i, gcd, lcm;

    printf("输入两个正整数：");
    scanf("%d %d",&n1,&n2);

    for(i=1; i <= n1 && i <= n2; ++i)
    {
        // 判断最大公约数
        if(n1%i==0 && n2%i==0)
            gcd = i;
    }

    lcm = (n1*n2)/gcd;
    printf("%d 和 %d 的最小公倍数为 %d", n1, n2, lcm);

    return 0;
}

```

输出结果:

输入两个正整数: 72 120
72 和 120 的最小公倍数为 360

(3) 实例编写:

```
#include <stdio.h>

int gcd(int m, int n){    //定义函数
    if(m % n == 0)        return n;
    else return gcd(n, m%n);    //辗转相除法
}

void main() {
    int a,b, t,lcm;
    printf("请输入两个数: ");
    scanf("%d%d",&a,&b);
    if(a < b) {t=a, a=b, b=t;}    //确保 a>b

    t = gcd(a, b);    // 递归调用
    printf("%d 和%d 的最大公约数是: %d\n", a, b, t);

    lcm = (a * b) / t;    // 通过最大公约数 t 计算最小公倍数
    printf("%d 和%d 的最小公倍数是: %d", a, b, lcm);
}
```

输出结果:

请输入两个数: 12 18
18 和 12 的最大公约数是: 6

26. 阶乘

一个正整数的阶乘（英语：factorial）是所有小于及等于该数的正整数的积，并且 0 的阶乘为 1。自然数 n 的阶乘写作 $n!$ 。

$n!=1\times 2\times 3\times \dots \times n$ 。阶乘亦可以递归方式定义： $0!=1$ ， $n!=(n-1)!\times n$ 。

(1) 方法一:

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

```

int main()
{
    int n, i;
    unsigned long long factorial = 1;

    printf("输入一个整数: ");
    scanf("%d", &n);

    // 如果输入是负数, 显示错误
    if (n < 0)
        printf("Error! 负数没有阶乘 jiechen");

    else
    {
        for(i=1; i<=n; ++i)
        {
            factorial *= i;           // factorial = factorial*i;
        }
        printf("%d! = %llu", n, factorial);
    }

    return 0;
}

```

输出结果:

输入一个整数: 10
 10! = 3628800

(2) 方法二: 使用递归

```

#include <stdio.h>
long int multiplyNumbers(int n);

int main()
{
    int n;
    printf("输入一个整数: ");
    scanf("%d", &n);
    printf("%d! = %ld", n, multiplyNumbers(n));
    return 0;
}

long int multiplyNumbers(int n)
{

```



```

    if (n >= 1)
        return n*multiplyNumbers(n-1);
    else
        return 1;
}

```

27. 循环输出 26 个字母

(1) 循环输出 26 个字母

```

#include <stdio.h>

int main()
{
    char c;

    for(c = 'A'; c <= 'Z'; ++c)
        printf("%c ", c);

    return 0;
}

```

输出结果:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

(2) 输出大写或小写字母

```

#include <stdio.h>

int main()
{
    char c;

    printf("输入 u 显示大写字母, 输入 1 显示小写字母: ");
    scanf("%c", &c);

    if(c== 'U' || c== 'u')
    {
        for(c = 'A'; c <= 'Z'; ++c)
            printf("%c ", c);
    }
}

```

```

else if (c == 'L' || c == 'l')
{
    for(c = 'a'; c <= 'z'; ++c)
        printf("%c ", c);
}
else
    printf("Error! 输入非法字符。");
return 0;
}

```

输出结果:

输入 u 显示大写字母, 输入 l 显示小写字母: l
a b c d e f g h i j k l m n o p q r s t u v w x y z

(3) 参考:

```

#include<stdio.h>

int main() {
    int A=65;
    int i;
    for(i=1;i<=26;i++) {
        printf("%c\n",A);
        A++;
    }
    return 0;
}

```

(4) 参考方法:

```

#include "stdio.h"

int main() {
    int i,n;
    ant:printf("输入 0 或 1 输出小写或大写:");
    scanf("%d",&i);
    if(i!=0 && i!=1) {
        printf("error! 输入 0 或 1 ! \n");
        goto ant;
    } else {
        if(i==0)
        {

```

```

        printf("0 小写:");
        for(n=97;n<123;n++)
            printf("%c ",n);
        printf("\n");
    }

    if(i==1) {
        printf("1 大写:");
        for(n=65;n<91;n++)
            printf("%c ",n);
        printf("\n");
    }
}
}

```

28. 判断数字为几位数

用户输入数字，判断该数字是几位数

```

#include <stdio.h>
int main()
{
    long long n;
    int count = 0;

    printf("输入一个整数: ");
    scanf("%lld", &n);

    while(n != 0)
    {
        // n = n/10
        n /= 10;
        ++count;
    }

    printf("数字是 %d 位数。", count);
}

```

输出结果:

输入一个整数: 2345
数字是 4 位数。

参考方法:

```
#include<stdio.h>

int main()
{
    int a, b=10, c=1;
    printf("请输入一个人数字: ");
    scanf("%d",&a);
    while(a/b>0)
    {
        b=b*10;
        c++;
    };
    printf("%d 是一个%d 位数 \n", a, c);
    return 0;
}
```

29. 计算一个数的 n 次方

计算一个数的 n 次方，例如：23，其中 2 为基数，3 为指数

(1)方法一： 使用 while

```
#include <stdio.h>

int main()
{
    int base, exponent;

    long long result = 1;

    printf("基数: ");
    scanf("%d", &base);

    printf("指数: ");
    scanf("%d", &exponent);

    while (exponent != 0)
    {
        result *= base;
        --exponent;
    }
}
```

```

    }
    printf("结果: %lld", result);

    return 0;
}

```

输出结果:

基数: 2
指数: 3
结果: 8

(2) 方法二: 使用 pow() 函数

```

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

int main()
{
    double base, exponent, result;

    printf("基数: ");
    scanf("%lf", &base);

    printf("指数: ");
    scanf("%lf", &exponent);

    // 计算结果
    result = pow(base, exponent);

    printf("%.1lf^%.1lf = %.2lf", base, exponent, result);

    return 0;
}

```

输出结果:

基数: 2
指数: 3
 $2.0^3.0 = 8.00$

(3) 方法三: 递归

```

#include <stdio.h>

int power(int n1, int n2);

int main()
{
    int base, powerRaised, result;

    printf("基数: ");
    scanf("%d",&base);

    printf("指数(正整数): ");
    scanf("%d",&powerRaised);

    result = power(base, powerRaised);

    printf("%d^%d = %d", base, powerRaised, result);
    return 0;
}

int power(int base, int powerRaised)
{
    if (powerRaised != 0)
        return (base*power(base, powerRaised-1));
    else
        return 1;
}

```

(4) 参考方法:

```

#include<stdio.h>

int main()
{
    int i,a,b;
    unsigned long c=1;
    printf("请输入基数:");
    scanf("%d",&a);
    printf("请输入指数");
    scanf("%d",&b);
    for(i=1;i<=b;i++)
    {
        c *= a;
    }
}

```

```

    }
    printf("%lu", c);
    return 0;
}

```

30. 判断回文数

判断一个数是否为回文数。

设 n 是一任意自然数。若将 n 的各位数字反向排列所得自然数 n_1 与 n 相等，则称 n 为一回文数。例如，若 $n=1234321$ ，则称 n 为一回文数；但若 $n=1234567$ ，则 n 不是回文数

```

include <stdio.h>

int main()
{
    int n, reversedInteger = 0, remainder, originalInteger;

    printf("输入一个整数: ");
    scanf("%d", &n);

    originalInteger = n;

    // 翻转
    while( n!=0 )
    {
        remainder = n%10;
        reversedInteger = reversedInteger*10 + remainder;
        n /= 10;
    }

    // 判断
    if (originalInteger == reversedInteger)
        printf("%d 是回文数。", originalInteger);
    else
        printf("%d 不是回文数。", originalInteger);

    return 0;
}

```

输出结果：

输入一个整数：12321

12321 是回文数

参考方法:

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

int main()
{
    int a=12021;
    char s[10]={' \0' }, s1[10]={' \0' };
    sprintf(s, "%d", a); // 将整数转换为字符串
    int n=strlen(s);
    int j=0;
    for(int i=n-1;i>=0;i--)
    {
        s1[j++]=s[i];
    }
    //s[j]='\0';
    printf("%s %s\n", s, s1);
    if(!strcmp(s, s1))
        printf("整数%d 是回文串", a);
    else
        printf("整数%d 不是回文串", a);
    return 0;
}
```

31. 判断素数

质数 (prime number) 又称素数, 有无限个。质数定义为在大于 1 的自然数中, 除了 1 和它本身以外不再有其他因数, 这样的数称为质数

(1)

```
#include <stdio.h>

int main()
{
    int n, i, flag = 0;

    printf("输入一个正整数: ");
    scanf("%d", &n);
```



```

for(i=2; i<=n/2; ++i)
{
    // 符合该条件不是素数
    if(n%i==0)
    {
        flag=1;
        break;
    }
}

if (flag==0)
    printf("%d 是素数",n);
else
    printf("%d 不是素数",n);

return 0;
}

```

输出结果:

输入一个正整数: 29
29 是素数

(2) 判断两个数之间的素数

```

#include <stdio.h>

int main()
{
    int low, high, i, flag;
    printf("输入两个整数: ");
    scanf("%d %d", &low, &high);

    printf("%d 与 %d 之间的素数为: ", low, high);

    while (low < high)
    {
        flag = 0;

        for(i = 2; i <= low/2; ++i)
        {
            if(low % i == 0)
            {

```

```

        flag = 1;
        break;
    }
}

if (flag == 0)
    printf("%d ", low);

    ++low;
}

return 0;
}

```

输出结果：

输入两个整数：100 200
 100 与 200 之间的素数为：101 103 107 109 113 127 131 137 139 149 151 157
 163 167 173 179 181 191 193 197 199

(3) 使用函数判断两数间的素数

```

#include <stdio.h>

int checkPrimeNumber(int n);
int main()
{
    int n1, n2, i, flag;

    printf("输入两个正整数：");
    scanf("%d %d", &n1, &n2);
    printf("%d 和 %d 间的素数为：", n1, n2);

    for(i=n1+1; i<n2; ++i)
    {
        // 判断是否为素数
        flag = checkPrimeNumber(i);

        if(flag == 1)
            printf("%d ", i);
    }
    return 0;
}

```

```
// 函数定义
int checkPrimeNumber(int n)
{
    int j, flag = 1;

    for(j=2; j <= n/2; ++j)
    {
        if (n%j == 0)
        {
            flag =0;
            break;
        }
    }
    return flag;
}
```

输出结果：

输入两个正整数：10 30
10 和 30 间的素数为：11 13 17 19 23 29

(4) 用 sqrt 降低时间复杂度

```
#include<stdio.h>
#include<math.h>
int issushu(int k);
int main() {
    int n, count=0;
    scanf("%d",&n);
    for(int i=n;i>3;i--)
        if(issushu(i))
            if(issushu(i-2))
                count++;
    printf("%d",count);
    return 0;
}

int issushu(int k) {
    for(int i=2;i<sqrt(k)+1;i++) {
        if(k%i==0)
            return 0;
    }
    return 1;
}
```

32. 判断 Armstrong 数(阿姆斯壮数)

Armstrong 数，就是 n 位数的各位数的 n 次方之和等于该数，如：

(1)

```
153=1^3+5^3+3^3
1634=1^4+6^4+3^4+4^4
#include <stdio.h>
int main()
{
    int number, originalNumber, remainder, result = 0;

    printf("输入三位数: ");
    scanf("%d", &number);

    originalNumber = number;

    while (originalNumber != 0)
    {
        remainder = originalNumber%10;
        result += remainder*remainder*remainder;
        originalNumber /= 10;
    }

    if(result == number)
        printf("%d 是 Armstrong 数", number);
    else
        printf("%d 不是 Armstrong 数", number);

    return 0;
}
```

输出结果：

```
输入三位数: 371
371 是 Armstrong
```

(2) 两数之间的 Armstrong 数

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

```

int main()
{
    int low, high, i, temp1, temp2, remainder, n = 0, result = 0;

    printf("输入两个整数: ");
    scanf("%d %d", &low, &high);
    printf("%d 和 %d 之间的 Armstrong 数为: ", low, high);

    for(i = low + 1; i < high; ++i)
    {
        temp2 = i;
        temp1 = i;

        // 计算
        while (temp1 != 0)
        {
            temp1 /= 10;
            ++n;
        }

        while (temp2 != 0)
        {
            remainder = temp2 % 10;
            result += pow(remainder, n);
            temp2 /= 10;
        }

        if (result == i) {
            printf("%d ", i);
        }

        n = 0;
        result = 0;
    }
    return 0;
}

```

输出结果:

输入两个整数: 100 1000
 100 和 1000 之间的 Armstrong 数为: 153 370 371 407

(3) 使用函数判断 Armstrong 数

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

int checkPrimeNumber(int n);
int checkArmstrongNumber(int n);

int main()
{
    int n, flag;

    printf("输入正整数: ");
    scanf("%d", &n);

    // 检测素数
    flag = checkPrimeNumber(n);
    if (flag == 1)
        printf("%d 是素数。\\n", n);
    else
        printf("%d 不是素数\\n", n);

    // 检测 Armstrong 数
    flag = checkArmstrongNumber(n);
    if (flag == 1)
        printf("%d 是 Armstrong 数。", n);
    else
        printf("%d 不是 Armstrong 数。", n);
    return 0;
}

int checkPrimeNumber(int n)
{
    int i, flag = 1;

    for(i=2; i<=n/2; ++i)
    {

        // 非素数条件
        if(n%i == 0)
        {
            flag = 0;
            break;
        }
    }
}
```

```

        }
    }
    return flag;
}

int checkArmstrongNumber(int number)
{
    int originalNumber, remainder, result = 0, n = 0, flag;

    originalNumber = number;

    while (originalNumber != 0)
    {
        originalNumber /= 10;
        ++n;
    }

    originalNumber = number;

    while (originalNumber != 0)
    {
        remainder = originalNumber%10;
        result += pow(remainder, n);
        originalNumber /= 10;
    }

    // 判断条件
    if(result == number)
        flag = 1;
    else
        flag = 0;

    return flag;
}

```

输出结果：

输入正整数：371
 371 不是素数
 371 是 Armstrong 数。

(4) 参考方法：

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

```

long long power(long long x,int y)
{
    long long sum = 1;
    while(y-->0)
    {
        sum *= x;
    }
    return sum;
}

int check_arm(long long n)
{
    long long temp,sum = 0,remainder,count = 0;
    temp = n;
    while(temp%10 != 0)
    {
        temp /= 10;
        ++count;
    }
    temp = n;
    while(temp%10 != 0)
    {
        sum += power(temp%10,count);
        temp /= 10;
    }
    if(n == sum)
    {
        return 1;
    }
    else
    {
        return 0;
    }
}

int main()
{
    long long temp,n,result,a,b,i;
    int flag;
    printf("Please enter a number:\n");
    scanf("%lld",&n);
    result = check_arm(n);
    if(result == 1)
    {
        printf("Yes, it's an Armstrong number.\n\n");
    }
}

```



```

    }
    else
    {
        printf("No, it isn't an Armstrong number.\n\n");
    }
    printf("Please enter two numbers as an interregional, seperated by a
space.\n");
    scanf("%lld %lld",&a,&b);
    flag = 1;
    if(a == b)
    {
        printf("That is no an valid interregional.\n");
        flag = 0;
    }
    else if(a > b)
    {
        temp = a;
        a = b;
        b = temp;
    }
    if(flag == 1)
    {
        printf("Armstrong number(s) between %lld and %lld
is(are):\n",a,b);
        for(i = a;i<=b;i++)
        {
            if(check_arm(i) == 1)
            {
                printf("%lld ",i);
            }
        }
    }
    return 0;
}

```

(5) 两数之间的 Armstrong 数:

```

#include <stdio.h>
int main() {
    int low,high;
    scanf("%d %d",&low,&high);
    while(low<=high) {
        if(arm(low)==1)    printf("%d \t",low);
    }
}

```

```

        low++;
    }

}

int arm(int a){

    int b=0,c=0,e=0; //初始化
    c=a;
    while(c>0){
        if(c%10>=0) e++;
        c =c/10;
    }
    c=a;
    while(a>0){
        b=b+pow(a%10,e);
        a=a/10;
    }
    if(c==b) return 1;
    else return 0;
}

```

33. 求一个整数的所有因数

假如 $a*b=c$ (a 、 b 、 c 都是整数)，那么我们称 a 和 b 就是 c 的因数

```

#include <stdio.h>

int main()
{
    int number, i;

    printf("输入一个整数: ");
    scanf("%d",&number);

    printf("%d 的因数有: ", number);
    for(i=1; i <= number; ++i)
    {
        if (number%i == 0)
        {
            printf("%d ", i);
        }
    }
}

```

```

    return 0;
}

```

输出结果:

输入一个整数: 60

60 的因数有: 1 2 3 4 5 6 10 12 15 20 30 60

寻找两数的所有公因数:

```

#include <stdio.h>

int main()
{
    int num1, num2, i;
    printf("输入两个数, 使用空格分隔:");
    scanf("%d %d", &num1, &num2);
    for (i = 1; i <= (num1 > num2 ? num2 : num1) ; ++i)
    {
        if (num1%i == 0 && num2%i == 0)
        {
            printf("%d ", i);
        }
    }
    return 0;
}

```

34. 创建各类三角形图案

(1) 创建三角形图案

```

#include <stdio.h>

int main()
{
    int i, j, rows;

    printf("行数: ");
    scanf("%d", &rows);

    for(i=1; i<=rows; ++i)
    {

```

```

        for(j=1; j<=i; ++j)
        {
            printf("* ");
        }
        printf("\n");
    }
    return 0;
}

```

输出结果:

```

*
* *
* * *
* * * *
* * * * *

```

(2) 使用数字

```

#include <stdio.h>

int main()
{
    int i, j, rows;

    printf("行数: ");
    scanf("%d",&rows);

    for(i=1; i<=rows; ++i)
    {
        for(j=1; j<=i; ++j)
        {
            printf("%d ", j);
        }
        printf("\n");
    }
    return 0;
}

```

输出结果:

```

1
1 2
1 2 3

```

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

(3) 使用字母

```
#include <stdio.h>
int main()
{
    int i, j;
    char input, alphabet = 'A';

    printf("输入大写字母: ");
    scanf("%c",&input);

    for(i=1; i <= (input-'A'+1); ++i)
    {
        for(j=1;j<=i;++j)
        {
            printf("%c", alphabet);
        }
        ++alphabet;

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

输出结果:

```
A
B B
C C C
D D D D
E E E E E
```

(4) 倒三角形

```
#include <stdio.h>
int main()
{
    int i, j, rows;

    printf("行数: ");
```

```

scanf("%d",&rows);

for(i=rows; i>=1; --i)
{
    for(j=1; j<=i; ++j)
    {
        printf("* ");
    }
    printf("\n");
}

return 0;
}

```

输出结果:

```

* * * * *
* * * *
* * *
* *
*

```

(5) 倒三角形数字

```

#include <stdio.h>
int main()
{
    int i, j, rows;

    printf("行数: ");
    scanf("%d",&rows);

    for(i=rows; i>=1; --i)
    {
        for(j=1; j<=i; ++j)
        {
            printf("%d ", j);
        }
        printf("\n");
    }

    return 0;
}

```

输出结果:

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

(6) 金字塔

```
#include <stdio.h>
int main()
{
    int i, space, rows, k=0;

    printf("Enter number of rows: ");
    scanf("%d",&rows);

    for(i=1; i<=rows; ++i, k=0)
    {
        for(space=1; space<=rows-i; ++space)
        {
            printf("  ");
        }

        while(k != 2*i-1)
        {
            printf("* ");
            ++k;
        }

        printf("\n");
    }

    return 0;
}
```

输出结果:

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

(7) 数字金字塔

```
#include <stdio.h>
int main()
{
    int i, space, rows, k=0, count = 0, count1 = 0;

    printf("行数: ");
    scanf("%d",&rows);

    for(i=1; i<=rows; ++i)
    {
        for(space=1; space <= rows-i; ++space)
        {
            printf("  ");
            ++count;
        }

        while(k != 2*i-1)
        {
            if (count <= rows-1)
            {
                printf("%d ", i+k);
                ++count;
            }
            else
            {
                ++count1;
                printf("%d ", (i+k-2*count1));
            }
            ++k;
        }
        count1 = count = k = 0;

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

输出结果:

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



```
4 5 6 7 6 5 4
5 6 7 8 9 8 7 6 5
```

(8) 倒金字塔

```
#include<stdio.h>
int main()
{
    int rows, i, j, space;

    printf("行数: ");
    scanf("%d",&rows);

    for(i=rows; i>=1; --i)
    {
        for(space=0; space < rows-i; ++space)
            printf(" ");

        for(j=i; j <= 2*i-1; ++j)
            printf("* ");

        for(j=0; j < i-1; ++j)
            printf("* ");

        printf("\n");
    }

    return 0;
}
```

输出结果:

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

(9) 杨辉三角

```
#include <stdio.h>
int main()
{
```

```

int rows, coef = 1, space, i, j;

printf("行数: ");
scanf("%d",&rows);

for(i=0; i<rows; i++)
{
    for(space=1; space <= rows-i; space++)
        printf(" ");

    for(j=0; j <= i; j++)
    {
        if (j==0 || i==0)
            coef = 1;
        else
            coef = coef*(i-j+1)/j;

        printf("%4d", coef);

    }
    printf("\n");
}

return 0;
}

```

输出结果:

```

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

```

(10) 弗洛伊德三角形

```

#include <stdio.h>
#define N 10
int main()
{
    int i, j, l;
    for(i=1, j=1; i<=N; i++)
    {
        for(l=1; l<=i; l++, j++)

```

```

        printf("%5d", j);
        printf("\n");
    }
    return 0;
}

```

输出结果：

```

1
2   3
4   5   6
7   8   9   10
11  12  13  14  15
16  17  18  19  20  21
22  23  24  25  26  27  28
29  30  31  32  33  34  35  36
37  38  39  40  41  42  43  44  45
46  47  48  49  50  51  52  53  54  55

```

(11) 数字金字塔 1:

```

#include<stdio.h>
int main() {
    int i, j, k, n;
    printf("请输入数字金字塔基座宽度(正奇数):");
    scanf("%d", &n);
    for(k=1; k<=(n+1)/2; k++) {
        for(i=1; i<(n+1)/2+1-k; i++)    printf(" ");
        for(j=k; j<(2*k); j++) printf("%d", j);
        for(j=2; j>=k; j--) printf("%d", j);
        printf("\n");
    }
}

```

(12) 数字金字塔 2:

```

#include <stdio.h>
int main()
{
    int blank, number, column, n;
    printf("请输入数字金字塔层数（小于 50）:");
    scanf("%d", &n);
    //判断输出三角形中最大数的位数，即最后一行中间数的位数

```

```

int count = 0;
int x = 0;
x = 2 * n - 1; //最后一行中间数数值
while (x != 0)
{
    x /= 10;
    ++count;
}
//输出
for (column = 1; column <= n; column++)
{
    for (blank = 1; blank < (n - column)*(count+1)+count; blank++)
printf(" "); //输出每行前部空格
    for (number = column; number < (2 * column); number++) //输出从小到大; 例: 123
    {
        if (number < 10 && count != 1)
        {
            printf(" %d", number);
        }
        else
        {
            printf(" %d", number);
        }
    }
    for (number -= 2; number >= column; number--) //输出从大到小; 例:
21 {
        if (number < 10 && count != 1)
        {
            printf(" %d", number);
        }
        else
        {
            printf(" %d", number);
        }
    }
    printf("\n\n");
}
return 0;
}

```

35. 表格形式输出数据

(1) 将 1~100 的数据以 10x10 矩阵格式输出

```
#include <stdio.h>

int main() {
    int i, j, count;

    for(i = 1; i <= 10; i++) {
        for(j = i; j <=100; j += 10 )
            printf(" %3d", j);

        printf("\n");
    }

    return 0;
}
```

输出

```
1  11  21  31  41  51  61  71  81  91
2  12  22  32  42  52  62  72  82  92
3  13  23  33  43  53  63  73  83  93
4  14  24  34  44  54  64  74  84  94
5  15  25  35  45  55  65  75  85  95
6  16  26  36  46  56  66  76  86  96
7  17  27  37  47  57  67  77  87  97
8  18  28  38  48  58  68  78  88  98
9  19  29  39  49  59  69  79  89  99
10 20  30  40  50  60  70  80  90 100
```

(2) 等差数列输出 10x10 矩阵格式

```
#include <stdio.h>

int main() {
    int i, j, count;
    int start, end;

    start = 2, end = 10;

    for(i = start; i <= end; i++) {
        count = i;
```

```

        for(j = 1; j <= 10; j++) {
            printf(" %3d", count*j);
        }

        printf("\n");
    }

    return 0;
}

```

输出结果:

```

2   4   6   8  10  12  14  16  18  20
3   6   9  12  15  18  21  24  27  30
4   8  12  16  20  24  28  32  36  40
5  10  15  20  25  30  35  40  45  50
6  12  18  24  30  36  42  48  54  60
7  14  21  28  35  42  49  56  63  70
8  16  24  32  40  48  56  64  72  80
9  18  27  36  45  54  63  72  81  90
10 20 30 40 50 60 70 80 90 100

```

(3) 乘法运算格式

```

#include <stdio.h>

int main() {
    int i, j, n;

    n = 3;
    j = 1;

    for(i = n; i <= (n*10); i+=n) {
        printf("%3d x %2d = %3d\n", n, j, i);
        j++;
    }

    return 0;
}

```

输出结果:

```

3 x 1 = 3
3 x 2 = 6

```

```
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30
```

(4) 将 1~100 的数据以 10x10 格式顺序输出

```
#include<stdio.h>

int main()
{
    int i, j, count=0;
    for(i=1;i<=10;i++) {
        for(j=1;j<=10;j++) {
            count++;
            printf("%4d", count);
        }
        printf("\n");
    }
    return 0;
}
```

36. 实现简单的计算器

(1) 实现加减乘除计算

```
# include <stdio.h>

int main() {

    char operator;
    double firstNumber, secondNumber;

    printf("输入操作符 (+, -, *, /): ");
    scanf("%c", &operator);

    printf("输入两个数字: ");
    scanf("%lf %lf", &firstNumber, &secondNumber);
```

```

switch(operator)
{
    case '+':
        printf("%.1lf + %.1lf = %.1lf", firstNumber, secondNumber,
firstNumber + secondNumber);
        break;

    case '-':
        printf("%.1lf - %.1lf = %.1lf", firstNumber, secondNumber,
firstNumber - secondNumber);
        break;

    case '*':
        printf("%.1lf * %.1lf = %.1lf", firstNumber, secondNumber,
firstNumber * secondNumber);
        break;

    case '/':
        printf("%.1lf / %.1lf = %.1lf", firstNumber, secondNumber,
firstNumber / secondNumber);
        break;

    // operator doesn't match any case constant (+, -, *, /)
    default:
        printf("Error! operator is not correct");
}

return 0;
}

```

输出结果:

输入操作符 (+, -, *, /): *
 输入两个数字: 4 5
 4.0 * 5.0 = 20.0

(2) 豪华版计算器

```

#include <stdio.h>

int get_option();
void print_result(int num1, int num2, int result, int option);

```



```

int main(void)
{
    int done = 0;
    int option, num1, num2, result;

    while(!done)
    {
        option = get_option();
        if(option == 5)
        {
            done = 1;
        }
        else {
            do {
                printf("\n 请输入两个数: ");
                scanf("%d %d", &num1, &num2);
                if(option == 4 && num2 == 0)
                {
                    printf("\n 对不起, 除数不能为零");
                }
            }
            else {
                switch(option) {
                    case 1:
                        result = num1 + num2;
                        break;
                    case 2:
                        result = num1 - num2;
                        break;
                    case 3:
                        result = num1 * num2;
                        break;
                    case 4:
                        result = num1 / num2;
                }
                print_result(num1, num2, result, option);
            }
        }
        while(option == 4 && num2 == 0);
    }

    return 0;
}

int get_option()

```

```

{
    int option;
    do
    {
        printf("\n *****");
        printf("\n *    1. 加法    *");
        printf("\n *    2. 减法    *");
        printf("\n *    3. 乘法    *");
        printf("\n *    4. 除法    *");
        printf("\n *    0. 退出    *");
        printf("\n *****");

        printf("\n 请输入您需要的功能: ");
        scanf("%d",&option);

        if(option <1 || option > 5)
        {
            printf("对不起您输入的数字有误, 请重新输入.\n");
        }
    }while(option <1 || option > 5);

    return option;
}

void print_result(int num1,int num2,int result,int option){
    char operator;
    switch(option){
        case 1:
            operator = '+';
            break;
        case 2:
            operator = '-';
            break;
        case 3:
            operator = '*';
            break;
        case 4:
            operator = '/';
            break;
    }
    printf("\n** %d %c %d = %d **\n",num1,operator,num2,result);
}

```

37. 计算一个数是否可为两个素数之和

计算一个数为两个素数之和

```
#include <stdio.h>

int checkPrime(int n);
int main()
{
    int n, i, flag = 0;

    printf("输入正整数: ");
    scanf("%d", &n);

    for(i = 2; i <= n/2; ++i)
    {
        // 检测判断
        if (checkPrime(i) == 1)
        {
            if (checkPrime(n-i) == 1)
            {
                printf("%d = %d + %d\n", n, i, n - i);
                flag = 1;
            }
        }
    }

    if (flag == 0)
        printf("%d 不能分解为两个素数。", n);

    return 0;
}

// 判断素数
int checkPrime(int n)
{
    int i, isPrime = 1;

    for(i = 2; i <= n/2; ++i)
    {
        if(n % i == 0)
        {
```

```

        isPrime = 0;
        break;
    }
}

return isPrime;
}

```

输出结果:

输入正整数: 34

34 = 3 + 31

34 = 5 + 29

34 = 11 + 23

34 = 17 + 17

38. 二进制与十进制相互转换

(1) 二进制转换为十进制

```

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

int convertBinaryToDecimal(long long n);

int main()
{
    long long n;
    printf("输入一个二进制数: ");
    scanf("%lld", &n);
    printf("二进制数 %lld 转换为十进制为 %d", n,
convertBinaryToDecimal(n));
    return 0;
}

int convertBinaryToDecimal(long long n)
{
    int decimalNumber = 0, i = 0, remainder;
    while (n!=0)
    {
        remainder = n%10;
        n /= 10;
        decimalNumber += remainder*pow(2, i);
    }
}

```

```

        ++i;
    }
    return decimalNumber;
}

```

输出结果:

输入一个二进制数: 110110111
 二进制数 110110111 转换为十进制为 439

(2) 十进制转换为二进制

```

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

long long convertDecimalToBinary(int n);

int main()
{
    int n;
    printf("输入一个十进制数: ");
    scanf("%d", &n);
    printf("十进制数 %d 转换为二进制位 %lld", n,
convertDecimalToBinary(n));
    return 0;
}

long long convertDecimalToBinary(int n)
{
    long long binaryNumber = 0;
    int remainder, i = 1, step = 1;

    while (n!=0)
    {
        remainder = n%2;
        printf("Step %d: %d/2, 余数 = %d, 商 = %d\n", step++, n,
remainder, n/2);
        n /= 2;
        binaryNumber += remainder*i;
        i *= 10;
    }
    return binaryNumber;
}

```

输出结果:

输入一个十进制数: 100

Step 1: $100/2$, 余数 = 0, 商 = 50

Step 2: $50/2$, 余数 = 0, 商 = 25

Step 3: $25/2$, 余数 = 1, 商 = 12

Step 4: $12/2$, 余数 = 0, 商 = 6

Step 5: $6/2$, 余数 = 0, 商 = 3

Step 6: $3/2$, 余数 = 1, 商 = 1

Step 7: $1/2$, 余数 = 1, 商 = 0

十进制数 100 转换为二进制位 1100100

(3) 参考方法:

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

```
//求 a 的 b 次方函数
```

```
unsigned ppow(unsigned a, unsigned b) {  
    unsigned cc = 1;  
    while (b) {  
        cc*=a;  
        b--;  
    }  
    return cc;  
}
```

```
int main()  
{
```

```
    unsigned long long n = 0;  
    unsigned i, k=0, count = 0;  
    printf("请输入 2 进制数: ");
```

```
    scanf_s("%lld", &n);           //用 scanf () 在 Visual Studio 2019 会  
    警告返回值被忽略
```

```
    while (n) {  
        i = n % 10;                //这个说错了, 2 和 10 都可以, 用 10  
    会比较好
```

```
        if (i) {                   //判断 i 是否为 0  
            count += ppow(2, k);  
        }
```

```
        k++;                       //k 做为 2 的次方数, 每次 n /= 10 就+1  
        n /= 10;
```

```
    }  
    printf(" 转成 10 进制= %d", count);  
    return 0;
```

```
}
```

39. 八进制与十进制相互转换

(1) 十进制转换为八进制

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

int convertDecimalToOctal(int decimalNumber);
int main()
{
    int decimalNumber;

    printf("输入一个十进制数: ");
    scanf("%d", &decimalNumber);

    printf("十进制数 %d 转换为八进制为 %d", decimalNumber,
convertDecimalToOctal(decimalNumber));

    return 0;
}

int convertDecimalToOctal(int decimalNumber)
{
    int octalNumber = 0, i = 1;

    while (decimalNumber != 0)
    {
        octalNumber += (decimalNumber % 8) * i;
        decimalNumber /= 8;
        i *= 10;
    }

    return octalNumber;
}
```

输出结果:

输入一个十进制数: 78
十进制数 78 转换为八进制为 116

(2) 八进制转换为十进制

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

long long convertOctalToDecimal(int octalNumber);
int main()
{
    int octalNumber;

    printf("输入一个八进制数: ");
    scanf("%d", &octalNumber);

    printf("八进制数 %d 转换为十进制为 %lld", octalNumber,
convertOctalToDecimal(octalNumber));

    return 0;
}

long long convertOctalToDecimal(int octalNumber)
{
    int decimalNumber = 0, i = 0;

    while(octalNumber != 0)
    {
        decimalNumber += (octalNumber%10) * pow(8, i);
        ++i;
        octalNumber/=10;
    }

    i = 1;

    return decimalNumber;
}
```

输出结果:

输入一个八进制数: 116
八进制数 116 转换为十进制为 78

(3) 任意进制转换

```
/*
```


任意进制转换

srBaseNum: 原进制下的数

srBase: 原进制的基

destBase: 目标进制的基

destBaseNum: 转换后的数

*/

```
int numBaseConversion(int srBaseNum, int srBase, int destBase)
{
    int destBaseNum=0, i=0;
    while (srBaseNum != 0) {
        destBaseNum += (srBaseNum % destBase) * pow(srBase, i);
        srBaseNum /= destBase;
        i ++;
    }
    return destBaseNum;
}
```

40. 八进制与二进制相互转换

(1) 二进制转换为八进制

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

int convertBinarytoOctal(long long binaryNumber);
int main()
{
    long long binaryNumber;

    printf("输入一个二进制数: ");
    scanf("%lld", &binaryNumber);

    printf("二进制数 %lld 转换为八进制为 %d", binaryNumber,
convertBinarytoOctal(binaryNumber));

    return 0;
}

int convertBinarytoOctal(long long binaryNumber)
{
    int octalNumber = 0, decimalNumber = 0, i = 0;
```

```

while(binaryNumber != 0)
{
    decimalNumber += (binaryNumber%10) * pow(2,i);
    ++i;
    binaryNumber/=10;
}

i = 1;

while (decimalNumber != 0)
{
    octalNumber += (decimalNumber % 8) * i;
    decimalNumber /= 8;
    i *= 10;
}

return octalNumber;
}

```

输出结果:

输入一个二进制数: 101001
 二进制数 101001 转换为八进制为 51

(2) 八进制转换为二进制

```

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

long long convertOctalToBinary(int octalNumber);
int main()
{
    int octalNumber;

    printf("输入一个八进制数: ");
    scanf("%d", &octalNumber);

    printf("八进制数 %d 转二进制为 %lld", octalNumber,
    convertOctalToBinary(octalNumber));

    return 0;
}

long long convertOctalToBinary(int octalNumber)

```

```

{
    int decimalNumber = 0, i = 0;
    long long binaryNumber = 0;

    while(octalNumber != 0)
    {
        decimalNumber += (octalNumber%10) * pow(8, i);
        ++i;
        octalNumber/=10;
    }

    i = 1;

    while (decimalNumber != 0)
    {
        binaryNumber += (decimalNumber % 2) * i;
        decimalNumber /= 2;
        i *= 10;
    }

    return binaryNumber;
}

```

输出结果:

输入一个八进制数: 51
八进制数 51 转二进制为 101001

(3) 分别定义函数实现八进制转二进制

//分别定义函数实现八进制转二进制

```

#include <stdio.h>

int octTodeci(int num); //声明八进制转十进制函数
long long deciTobina(int num); //声明十进制转二进制函数
int powNum(int base, int exp); //声明求 base 的 exp 次方函数

void main()
{
    int octNum, deciNum;
    long long binaNum;
    printf("Please enter an octal number:");
    scanf("%d", &octNum);
}

```

```

        deciNum = octTodeci(octNum);
        binaNum = deciTobina(deciNum);
        printf("The conversion from octal number %d to binary number
is %lld.\n", octNum, binaNum);
    }

int octTodeci(int num)//定义八进制转十进制函数
{
    int n = 0, remainder, result = 0;
    while (num != 0)
    {
        remainder = num %10;
        result += remainder*powNum(8, n);
        num /= 10;
        ++n;
    }
    return result;
}

long long  deciTobina(int num)//定义十进制转二进制函数
{
    int remainder, n=0;
    long long result = 0;
    while (num != 0)
    {
        remainder = num % 2;
        result += remainder*powNum(10, n);
        num /= 2;
        ++n;
    }
    return result;
}

int powNum(int base, int exp)//定义求 base 的 exp 次方函数
{
    int result = 1, k;
    for (k = exp; k >= 1; --k)
        result *= base;
    return result;
}

```

41. 字符串翻转

(1) 字符串翻转

```
#include <stdio.h>
void reverseSentence();

int main()
{
    printf("输入一个字符串: ");
    reverseSentence();

    return 0;
}

void reverseSentence()
{
    char c;
    scanf("%c", &c);

    if( c != '\n')
    {
        reverseSentence();
        printf("%c", c);
    }
}
```

输出结果:

输入一个字符串: runoob
boonur

(2) 参考方法 1:

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

char* reverseStr(char* str);

int main()
{
    char str[30];
    printf("输入一个字符串: ");
    scanf("%s", str);
    printf("翻转之前的字符串为:%s\n", str);
```

```

    printf("翻转之后的字符串为:%s", reverseStr(str));
    return 0;
}

char* reverseStr(char* str)
{
    int i=0;
    int j=strlen(str)-1;
    char temp;
    while (i<j)
    {
        temp=*(str+i);
        *(str+i)=*(str+j);
        *(str+j)=temp;
        i++;
        j--;
    }
    return str;
}

```

(3) 参考方法 2:

```

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

void reverseStr(char* str);

int main()
{
    char str[30];
    printf("输入一个字符串: ");
    scanf("%s", str);
    printf("翻转之前的字符串为:");
    reverseStr(str);
    return 0;
}

void reverseStr(char* str)
{
    if(*str=='\0') return;
    reverseStr(str+1);
    printf("%c", *str);
}

```

(4) 参考文档:

```
#include <stdio.h>
#include <string.h>
int main()
{
    char c[40];
    int i=1;
    printf("请输入字符串: ");
    scanf("%s", c);
    int j=strlen(c);
    do{
        printf("%c", c[j-i]);
        ++i;
    } while(i<=j);
    return 0;
}
```

42. 计算数组元素平均值

使用 for 循环迭代出输出元素，并将各个元素相加算出总和，再除于元素个数：

(1) 实例 1

```
#include <stdio.h>

int main() {
    int array[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0};
    int sum, loop;
    float avg;

    sum = avg = 0;

    for(loop = 0; loop < 10; loop++) {
        sum = sum + array[loop];
    }

    avg = (float)sum / loop;

    printf("平均值为 %.2f", avg);

    return 0;
}
```

```
}
```

输出结果:

平均值为 4.50

(2) 实例 2 用户自定义输入:

```
#include <stdio.h>

int main()
{
    int n, i;
    float num[100], sum = 0.0, average;

    printf("输入元素个数: ");
    scanf("%d", &n);

    while (n > 100 || n <= 0)
    {
        printf("Error! 数字需要在 1 到 100 之间。\\n");
        printf("再次输入: ");
        scanf("%d", &n);
    }

    for(i = 0; i < n; ++i)
    {
        printf("%d. 输入数字: ", i+1);
        scanf("%f", &num[i]);
        sum += num[i];
    }

    average = sum / n;
    printf("平均值 = %.2f", average);

    return 0;
}
```

输出结果:

输入元素个数: 4

1. 输入数字: 1
2. 输入数字: 2
3. 输入数字: 4

4. 输入数字: 8
平均值 = 3.75

(3) 用 scanf 的特性来控制循环:

```
#include <stdio.h>

int main(void)
{
    float x;
    printf("请输入数字: (输入 q 退出) ");
    int i=0;
    float status;
    float sum=0;
    float avg=0;
    status=scanf("%f", &x);
    while ( status==1 ) {
        sum+=x;
        printf("请输入数字: (输入 q 退出) ");
        status=scanf("%f", &x);
        i++;
    }
    avg=sum/(i);
    printf("%.2f", avg);
    return 0;
}
```

43. 输出数组

(1) 使用 for 循环输出数组:

```
#include <stdio.h>

int main() {
    int array[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0};
    int loop;

    for(loop = 0; loop < 10; loop++)
        printf("%d ", array[loop]);

    return 0;
}
```

输出结果:

1 2 3 4 5 6 7 8 9 0

(2) 使用 for 循环逆向输出数组:

```
#include <stdio.h>

int main() {
    int array[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0};
    int loop;

    for(loop = 9; loop >= 0; loop--)
        printf("%d ", array[loop]);

    return 0;
}
```

输出结果:

0 9 8 7 6 5 4 3 2 1

44. 计算数组元素之和

使用 for 循环迭代出输出元素，并将各个元素相加:

```
#include <stdio.h>

int main() {
    int array[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0};
    int sum, loop;

    sum = 0;

    for(loop = 9; loop >= 0; loop--) {
        sum = sum + array[loop];
    }

    printf("元素和为: %d", sum);

    re
```

输出结果:

元素和为：45

45. 查找数组中最大的元素值

(1) 查找数组中最大的元素值

```
#include <stdio.h>

int main() {
    int array[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0};
    int loop, largest;

    largest = array[0];

    for(loop = 1; loop < 10; loop++) {
        if( largest < array[loop] )
            largest = array[loop];
    }

    printf("最大元素为 %d", largest);

    return 0;
}
```

输出结果：

最大元素为 9

(2) 用户自定义输出：

```
#include <stdio.h>

int main()
{
    int i, n;
    float arr[100];

    printf("输入元素个数 (0~100) : ");
    scanf("%d", &n);
    printf("\n");

    // 接收用户输入
```

```

for(i = 0; i < n; ++i)
{
    printf("输入数字 %d: ", i+1);
    scanf("%f", &arr[i]);
}

// 循环，并将最大元素存储到 arr[0]
for(i = 1; i < n; ++i)
{
    // 如果要查找最小值可以将 < 换成 >
    if(arr[0] < arr[i])
        arr[0] = arr[i];
}
printf("最大元素为 = %.2f", arr[0]);

return 0;
}

```

输出结果:

输入元素个数 (0~100) : 4

输入数字 1: 12

输入数字 2: 23

输入数字 3: 1

输入数字 4: 3

最大元素为 = 23.00

(3) 三元运算符应用求最大:

```

#include <stdio.h>
int main() {
    int n[]={3, 2, 1, 4, 7, 6, 5, 8, 0, 9};
    int largest, i, num;
    num= sizeof(n)/sizeof(n[0]);
    for(i=0; i<num; i++) largest=(n[i] > largest)? n[i] : largest;
    printf("Largest Number is:%d", largest);
}

```

46. 查找数组中最小的元素

使用 for 循环迭代出输出元素，并将各个元素相加算出总和，再除于元素个数:

```

#include <stdio.h>

int main() {
    int array[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0};
    int loop, smallest;

    smallest = array[0];

    for(loop = 1; loop < 10; loop++) {
        if( smallest > array[loop] )
            smallest = array[loop];
    }

    printf("最小元素为 %d", smallest);

    return 0;
}

```

输出结果:

最小元素为 0

47. 数组拆分与合并

(1) 将一个数组拆分为一个为奇数数组，一个为偶数数组:

```

#include <stdio.h>

int main() {
    int array[10] = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
    int even[10], odd[10];
    int loop, e, d;

    e = d = 0;

    for(loop = 0; loop < 10; loop++) {
        if(array[loop]%2 == 0) {
            even[e] = array[loop];
            e++;
        }else {
            odd[d] = array[loop];

```

```

        d++;
    }
}

printf(" 原始数组 -> ");

for(loop = 0; loop < 10; loop++)
    printf(" %d", array[loop]);

printf("\n 偶数 -> ");
for(loop = 0; loop < e; loop++)
    printf(" %d", even[loop]);

printf("\n 奇数 -> ");
for(loop = 0; loop < d; loop++)
    printf(" %d", odd[loop]);

return 0;
}

```

输出结果:

```

原始数组 ->  0 1 2 3 4 5 6 7 8 9
偶数 ->    0 2 4 6 8
奇数 ->    1 3 5 7 9

```

(2) 将奇数数组与偶数数组合并为一个数组:

```

#include <stdio.h>

int main() {
    int array[10];
    int even[5] = {0, 2, 4, 6, 8};
    int odd[5]  = {1, 3, 5, 7, 9};

    int loop, index, e_len, o_len;

    e_len = o_len = 5;

    index = 0;

    for(loop = 0; loop < e_len; loop++) {
        array[index] = even[loop];
        index++;
    }
}

```

```

    }

    for(loop = 0; loop < o_len; loop++) {
        array[index] = odd[loop];
        index++;
    }

    printf("\n 偶数 -> ");

    for(loop = 0; loop < e_len; loop++)
        printf(" %d", even[loop]);
    printf("\n 奇数  -> ");

    for(loop = 0; loop < o_len; loop++)
        printf(" %d", odd[loop]);

    printf("\n 合并后 -> ");

    for(loop = 0; loop < 10; loop++)
        printf(" %d", array[loop]);

    return 0;
}

```

输出结果:

```

偶数 ->  0 2 4 6 8
奇数  ->  1 3 5 7 9
合并后 -> 0 2 4 6 8 1 3 5 7 9

```

48. 数组拷贝

(1) 将一个数组复制给另外一个数组:

```

#include <stdio.h>

int main() {
    int original[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0};
    int copied[10];
    int loop;

    for(loop = 0; loop < 10; loop++) {
        copied[loop] = original[loop];
    }
}

```

```

    }
    printf("元素数组 -> 拷贝后的数组 \n");

    for(loop = 0; loop < 10; loop++) {
        printf("    %2d        %2d\n", original[loop], copied[loop]);
    }

    return 0;
}

```

输出结果:

元素数组 -> 拷贝后的数组

1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
0	0

(2) 值传递与地址传递的区别:

```

#include "stdio.h"

int ArrayCopy(char* ori, char* cop, char Length)
{
    char loop;
    for(loop = 0; loop < Length; loop++)
    {
        *cop++ = *ori++;
    }
    return 0;
}

int main()
{
    char original[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0};
    char *copiedOne=original;
    char copiedTwo[10];
    char loop;

```



```

char Length;

Length = sizeof(original);
printf("元素数组 -> 拷贝后的数组 \n");
for(loop = 0; loop < sizeof(original); loop++)
{
    printf("    %2d        %2d\n", original[loop], copiedOne[loop]);
}
ArrayCopy(original, copiedTwo, Length);

copiedTwo[9] = 20;
printf("值传递: \n");
printf("值传递更改数据后, 只有拷贝的数组中的元素被更改:  \n");
printf("元素数组 -> 拷贝后的数组 \n");
for(loop = 0; loop < sizeof(original); loop++)
{
    printf("    %2d        %2d\n", original[loop], copiedTwo[loop]);
}

copiedOne[9] = 50;
printf("地址传递: \n");
printf("地址传递更改数据后, 原数组元素与拷贝的数组中的元素同时被更改:  \n");
printf("元素数组 -> 拷贝后的数组 \n");
for(loop = 0; loop < sizeof(original); loop++)
{
    printf("    %2d        %2d\n", original[loop], copiedOne[loop]);
}

return 0;
}

```

(3) 参考方法:

```

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

int main()
{
    int array[] = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };
    int *arrayPointer = (int*)calloc(sizeof(array) / sizeof(array[0]),
sizeof(int)), i;//令指针指向一块新内存区, 该内存区大小刚好可以容纳旧数组所有元素

```

```

    for ( i = 0; i < sizeof(array)/sizeof(array[0]); i++)
        *arrayPointer++ = array[i];

    *arrayPointer = '\0'; //为新数组手动添上结束标记
    arrayPointer -= sizeof(array) / sizeof(array[0]); //指针归首位
    printf("元素数组 -> 拷贝后的数组 \n");
    for (i = 0; i < sizeof(array) / sizeof(array[0]); i++)
        printf("%5d%13d\n", array[i], arrayPointer[i]);
    return 0;
}

```

49. 计算标准偏差

```

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

float calculateSD(float data[]);

int main()
{
    int i;
    float data[10];

    printf("输入 10 个元素: ");
    for(i=0; i < 10; ++i)
        scanf("%f", &data[i]);

    printf("\n 标准偏差 = %.6f", calculateSD(data));

    return 0;
}

float calculateSD(float data[])
{
    float sum = 0.0, mean, standardDeviation = 0.0;

    int i;

    for(i=0; i<10; ++i)
    {
        sum += data[i];
    }
}

```

```

    mean = sum/10;

    for(i=0; i<10; ++i)
        standardDeviation += pow(data[i] - mean, 2);

    return sqrt(standardDeviation/10);
}

```

输出结果:

输入 10 个元素: 1

2
3
4
5
6
7
8
9
10

标准偏差 = 2.872281

50. 两个矩阵相加

(1) 使用多维数组将两个矩阵相加

```

#include <stdio.h>

int main() {
    int r, c, a[100][100], b[100][100], sum[100][100], i, j;

    printf("输入行数 ( 1 ~ 100): ");
    scanf("%d", &r);
    printf("输入列数 ( 1 ~ 100): ");
    scanf("%d", &c);

    printf("\n 输入第一维数组的元素:\n");

    for(i=0; i<r; ++i)
        for(j=0; j<c; ++j)
        {
            printf("输入元素 a%d%d: ", i+1, j+1);

```

```

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

    printf("输入二维数组的元素:\n");
    for(i=0; i<r; ++i)
        for(j=0; j<c; ++j)
        {
            printf("输入元素 a%d%d: ",i+1, j+1);
            scanf("%d", &b[i][j]);
        }

    // 相加

    for(i=0;i<r;++i)
        for(j=0;j<c;++j)
        {
            sum[i][j]=a[i][j]+b[i][j];
        }

    // 显示结果
    printf("\n 二维数组相加结果: \n\n");

    for(i=0;i<r;++i)
        for(j=0;j<c;++j)
        {

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

            if(j==c-1)
            {
                printf("\n\n");
            }
        }

    return 0;
}

```

输出结果:

输入行数 (1 ~ 100): 2
 输入列数 (1 ~ 100): 3

输入第一维数组的元素:
 输入元素 a11: 2

输入元素 a12: 3
输入元素 a13: 4
输入元素 a21: 5
输入元素 a22: 2
输入元素 a23: 3
输入二维数组的元素:
输入元素 a11: -4
输入元素 a12: 5
输入元素 a13: 3
输入元素 a21: 5
输入元素 a22: 6
输入元素 a23: 3

二维数组相加结果:

-2 8 7

10 8 6

(2) 矩阵相乘实例

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

```
// 求 m*k 型矩阵 A 左乘 k*n 型矩阵 B 后的 m*n 型矩阵 C
```

```
int main()
{
    struct Matrixs
    {
        int elements[100][100];
        int row;
        int col;
    }matrixA, matrixB, matrixC;

    int m, k, n;
    printf("请输入矩阵 A 行数和列数: ");
    scanf("%d %d", &matrixA.row, &matrixA.col);
    matrixB.row = matrixA.col; // 矩阵 A 的列数等于矩阵 B 的行数
    printf("请输入矩阵 B 列数: ");
    scanf("%d", &matrixB.col);
    printf("\n 输入矩阵 A:\n");
    for (m = 0; m < matrixA.row; ++m)
        for (k = 0; k < matrixA.col; ++k)
```

```

    {
        printf("请输入元素 A(%d,%d):", m+1, k+1);
        scanf("%d", &matrixA.elements[m][k]);
    }

printf("\n 输入矩阵 B:\n");
for (k = 0; k < matrixA.col; ++k)
    for (n = 0; n < matrixB.col; ++n)
    {
        printf("请输入元素 B(%d,%d):", k+1, n+1);
        scanf("%d", &matrixB.elements[k][n]);
    }
for (m = 0; m < matrixA.row; ++m)
    for (n = 0; n < matrixB.col; ++n)
        for (k = 0; k < matrixA.col; ++k)
        {
            if (k == 0) matrixC.elements[m][n] = 0; //为新矩阵 C 每个元
            matrixC.elements[m][n] += matrixA.elements[m][k] *
            matrixB.elements[k][n];
        }
        printf("\n\n 矩阵 A 为: \n\n");
for (m = 0; m < matrixA.row; ++m)
{
    printf("  |");
    for (k = 0; k < matrixA.col; ++k)
    {
        printf(" %-5d", matrixA.elements[m][k]);
    }
    printf("\b\b\b|\n");
}
printf("\n\n 矩阵 B 为: \n\n");
for (k = 0; k < matrixB.row; ++k)
{
    printf("  |");
    for (n = 0; n < matrixB.col; ++n)
    {
        printf(" %-6d", matrixB.elements[k][n]);
    }
    printf("\b\b\b|\n");
}

printf("\n 矩阵 A 左乘矩阵 B 为: \n\n");
for (m = 0; m < matrixA.row; ++m)

```

```

{
    printf("  |");
    for (n = 0; n < matrixB.col; ++n)
    {
        printf(" %-6d", matrixC.elements[m][n]);
    }
    printf("\b\b|\n");
}
printf("\n");
return 0;
}

```

51. 矩阵转换

```

#include <stdio.h>

int main()
{
    int a[10][10], transpose[10][10], r, c, i, j;
    printf("输入矩阵的行与列: ");
    scanf("%d %d", &r, &c);

    // 存储矩阵的元素
    printf("\n 输入矩阵元素:\n");
    for(i=0; i<r; ++i)
        for(j=0; j<c; ++j)
        {
            printf("输入元素 a%d%d: ", i+1, j+1);
            scanf("%d", &a[i][j]);
        }

    // 显示矩阵 a[][] */
    printf("\n 输入矩阵: \n");
    for(i=0; i<r; ++i)
        for(j=0; j<c; ++j)
        {
            printf("%d  ", a[i][j]);
            if (j == c-1)
                printf("\n\n");
        }

    // 转换
    for(i=0; i<r; ++i)

```

```

        for(j=0; j<c; ++j)
        {
            transpose[j][i] = a[i][j];
        }

    // 显示转换后的矩阵 a
    printf("\n 转换后矩阵:\n");
    for(i=0; i<c; ++i)
        for(j=0; j<r; ++j)
        {
            printf("%d  ", transpose[i][j]);
            if(j==r-1)
                printf("\n\n");
        }

    return 0;
}

```

输出结果:

输入矩阵的行与列: 2 3

输入矩阵元素:

输入元素 a11: 2

输入元素 a12: 3

输入元素 a13: 4

输入元素 a21: 5

输入元素 a22: 6

输入元素 a23: 4

输入矩阵:

2 3 4

5 6 4

转换后矩阵:

2 5

3 6

4 4

52. 使用指针访问数组元素


```

#include <stdio.h>

int main()
{
    int data[5], i;
    printf("输入元素: ");

    for(i = 0; i < 5; ++i)
        scanf("%d", data + i);

    printf("你输入的是: \n");
    for(i = 0; i < 5; ++i)
        printf("%d\n", *(data + i));

    return 0;
}

```

输出结果:

```

输入元素: 1
2
3
5
4
你输入的是:
1
2
3
5
4

```

53. 使用引用循环替换数值

a、b、c 三个变量，通过引用按顺序循环替换他们的值

```

#include<stdio.h>

void cyclicSwap(int *a,int *b,int *c);

int main()
{
    int a, b, c;

    printf("输入 a, b 和 c 的值: ");

```

```

scanf("%d %d %d",&a,&b,&c);

printf("交换前:\n");
printf("a = %d \nb = %d \nc = %d\n",a,b,c);

cyclicSwap(&a, &b, &c);

printf("交换后:\n");
printf("a = %d \nb = %d \nc = %d",a, b, c);

return 0;
}
void cyclicSwap(int *a,int *b,int *c)
{

    int temp;

    // 交换
    temp = *b;
    *b = *a;
    *a = *c;
    *c = temp;
}

```

输出结果:

输入 a, b 和 c 的值: 1 2 3
 交换前:
 a = 1
 b = 2
 c = 3
 交换后:
 a = 3
 b = 1
 c = 2

54. 判断最大值

通过用户输入指定的数值，来判断最大值

判断最大值

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

```

#include <stdlib.h>

int main()
{
    int i, num;
    float *data;

    printf("输入元素个数(1 ~ 100): ");
    scanf("%d", &num);

    // 为 'num' 元素分配内存
    data = (float*) calloc(num, sizeof(float));

    if(data == NULL)
    {
        printf("Error!!! 内存没有分配。");
        exit(0);
    }

    printf("\n");

    // 用户输入
    for(i = 0; i < num; ++i)
    {
        printf("输入数字 %d: ", i + 1);
        scanf("%f", data + i);
    }

    // 循环找出最大值
    for(i = 1; i < num; ++i)
    {
        // 如果需要找出最小值可以将 < 改为 >
        if(*data < *(data + i))
            *data = *(data + i);
    }

    printf("最大元素 = %.2f", *data);

    return 0;
}

```

输出结果:

输入元素个数(1 ~ 100): 5

输入数字 1: 12
输入数字 2: 32
输入数字 3: 6
输入数字 4: 56
输入数字 5: 21
最大元素 = 56.00

55. 删除字符串中的特殊字符

(1) 删除字符串中的除字母外的字符

```
#include<stdio.h>

int main()
{
    char line[150];
    int i, j;
    printf("输入一个字符串: ");
    fgets(line, (sizeof line / sizeof line[0]), stdin);

    for(i = 0; line[i] != '\0'; ++i)
    {
        while (!( (line[i] >= 'a' && line[i] <= 'z') || (line[i] >= 'A'
&& line[i] <= 'Z') || line[i] == '\0') )
        {
            for(j = i; line[j] != '\0'; ++j)
            {
                line[j] = line[j+1];
            }
            line[j] = '\0';
        }
    }
    printf("输出: ");
    puts(line);
    return 0;
}
```

输出结果:

输入一个字符串: run4#\$1oob
输出: runoob

(2) 参考方法:

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

int main()
{
    char line[100];
    int i, j, len;
    printf("输入一个字符串: ");
    scanf("%s", line);
    len = strlen(line);
    for(i=0; i<len+1; i++)
    {
        if((line[i]>='a' && line[i]<='z') || (line[i]>='A' && line[i]<='Z'))
            continue;
        for(j=i; j<len; j++)
        {
            line[j] = line[j+1];
        }
        len--;
        i--;
    }
    //line[len]='\0';
    printf("%s\n", line);
    return 0;
}
```

56. 连接字符串

(1) 使用 `strcat()` 连接两个字符串

```
#include <stdio.h>
int main()
{
    char s1[100], s2[100], i, j;

    printf("输入第一个字符串: ");
    scanf("%s", s1);

    printf("输入第二个字符串: ");
    scanf("%s", s2);
```

```

// 计算字符串 s1 长度
for(i = 0; s1[i] != '\0'; ++i);

for(j = 0; s2[j] != '\0'; ++j, ++i)
{
    s1[i] = s2[j];
}

s1[i] = '\0';
printf("连接后: %s", s1);

return 0;
}

```

输出结果:

输入第一个字符串: google
 输入第二个字符串: runoob
 连接后: googlerunoob

(2) 参考方法:

```

#include <stdio.h>
#include <string.h>
int main()
{
    char s1[100], s2[100];

    printf("输入第一个字符串: ");
    scanf("%s", s1);

    printf("输入第二个字符串: ");
    scanf("%s", s2);

    int len1 = strlen(s1);
    int len2 = strlen(s2);
    //printf("%d %d", len1, len2);
    for (int i = 0; i <= len2; i++)
    {
        s1[i + len1] = s2[i];
    }

    printf("%s\n", s1);
}

```

```

    // system("pause");
    return 0;
}

```

(3) 指针方法:

```

#include <stdio.h>

int main()
{
    char str1[100], str2[100], *p = str1, n = 0;
    printf("请输入第一个字符串:");
    scanf("%s", str1);
    printf("请输入第二个字符串:");
    scanf("%s", str2);
    while (*p++ != '\0'); /*移动指针到 str1 尾*/
    --p; //回退一个单元, 以便覆盖 str1 末的 '\0'; //回退一个单元, 以便覆盖
    str1 末的 '\0'
    while (str2[n] != '\0')
    {
        *p++ = str2[n]; //将 str2 接到 str1 末
        ++n;
    };
    *p = '\0'; //拼接完成, 手动为 str1 末添上结束标记
    printf("结果为:\n%s\n\n", str1);
    return 0;
}

```

57. 计算字符串长度

(1) 使用 strlen()

```

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

int main()
{
    char s[1000];
    int len;

    printf("输入字符串: ");
    scanf("%s", s);
}

```

```

    len = strlen(s);

    printf("字符串长度: %d", len);
    return 0;
}

```

输出结果:

输入字符串: runoob
字符串长度: 6

(2) 不使用 strlen()

```

#include <stdio.h>

int main()
{
    char s[1000], i;

    printf("输入字符串: ");
    scanf("%s", s);

    for(i = 0; s[i] != '\0'; ++i);

    printf("字符串长度: %d", i);
    return 0;
}

```

输出结果:

输入字符串: runoob
字符串长度: 6

58. 查找字符在字符串中出现的次数

查找字符在字符串中的起始位置(索引值从 0 开始)

```

#include <stdio.h>

int main()
{
    char str[1000], ch;
    int i, frequency = 0;

```



```

printf("输入字符串: ");
fgets(str, (sizeof str / sizeof str[0]), stdin);

printf("输入要查找的字符: ");
scanf("%c",&ch);

for(i = 0; str[i] != '\0'; ++i)
{
    if(ch == str[i])
        ++frequency;
}

printf("字符 %c 在字符串中出现的次数为 %d", ch, frequency);

return 0;
}

```

输出结果:

```

输入字符串: runoob
输入要查找的字符: o
字符 o 在字符串中出现的次数为 2

```

59. 字符串中各种字符计算

计算字符串中的元音、辅音、数字、空白符

```

#include <stdio.h>

int main()
{
    char line[150];
    int i, vowels, consonants, digits, spaces;

    vowels = consonants = digits = spaces = 0;

    printf("输入一个字符串: ");
    scanf("%[^\n]", line);

    for(i=0; line[i]!='\0'; ++i)
    {
        if(line[i]=='a' || line[i]=='e' || line[i]=='i' ||
           line[i]=='o' || line[i]=='u' || line[i]=='A' ||

```

```

        line[i]=='E' || line[i]=='I' || line[i]=='O' ||
        line[i]=='U')
    {
        ++vowels;
    }
    else if((line[i]>='a' && line[i]<='z') || (line[i]>='A' &&
line[i]<='Z'))
    {
        ++consonants;
    }
    else if(line[i]>='0' && line[i]<='9')
    {
        ++digits;
    }
    else if (line[i]==' ')
    {
        ++spaces;
    }
}

printf("元音: %d", vowels);
printf("\n 辅音: %d", consonants);
printf("\n 数字: %d", digits);
printf("\n 空白符: %d", spaces);

return 0;
}

```

输出结果:

输入一个字符串: runoob123 erkcomsww dfrunoob

元音: 8

辅音: 15

数字: 3

空白符: 2

60. 字符串复制

(1) 使用 strcpy()

```

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

```

```

int main()
{
    char src[40];
    char dest[100];

    memset(dest, '\0', sizeof(dest));
    strcpy(src, "This is runoob.com");
    strcpy(dest, src);

    printf("最终的目标字符串:  %s\n", dest);

    return(0);
}

```

输出结果:

最终的目标字符串: This is runoob.com

(2) 不使用 strcpy()

```

#include <stdio.h>

int main()
{
    char s1[100], s2[100], i;

    printf("字符串 s1: ");
    scanf("%s", s1);

    for(i = 0; s1[i] != '\0'; ++i)
    {
        s2[i] = s1[i];
    }

    s2[i] = '\0';
    printf("字符串 s2: %s", s2);

    return 0;
}

```

输出结果:

字符串 s1: runoob
字符串 s2: runoob

61. 字符串排序

按字典顺序排序

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

int main()
{
    int i, j;
    char str[10][50], temp[50];

    printf("输入 10 个单词:\n");

    for(i=0; i<10; ++i) {
        scanf("%s[^\n]", str[i]);
    }

    for(i=0; i<9; ++i) {
        for(j=i+1; j<10 ; ++j)
        {
            if(strcmp(str[i], str[j])>0)
            {
                strcpy(temp, str[i]);
                strcpy(str[i], str[j]);
                strcpy(str[j], temp);
            }
        }
    }

    printf("\n 排序后: \n");
    for(i=0; i<10; ++i)
    {
        puts(str[i]);
    }

    return 0;
}
```

输出结果:

输入 10 个单词:

C

C++
Java
PHP
Python
Perl
Ruby
R
JavaScript
PHP

排序后:

C
C++
Java
JavaScript
PHP
PHP
Perl
Python
R
Ruby

62. 使用结构体 (struct)

(1) 使用结构体 (struct) 存储学生信息

```
#include <stdio.h>
struct student
{
    char name[50];
    int roll;
    float marks;
} s;

int main()
{
    printf("输入信息:\n");

    printf("名字: ");
    scanf("%s", s.name);

    printf("编号: ");
    scanf("%d", &s.roll);
```

```

    printf("成绩: ");
    scanf("%f", &s.marks);

    printf("显示信息:\n");

    printf("名字: ");
    puts(s.name);

    printf("编号: %d\n", s.roll);

    printf("成绩: %.1f\n", s.marks);

    return 0;
}

```

输出结果:

```

输入信息:
名字: runoob
编号: 123
成绩: 89
显示信息:
名字: runoob
编号: 123
成绩: 89.0

```

(2) 两种不同对齐方式下计算 sizeof(struct)

```

#include "stdio.h"

struct
{
    int i;
    char j;
}s1;

#pragma pack(1)
struct
{
    int i;
    char j;
}s2;

```

```

int main(void)
{
    int size_s1;
    int size_s2;

    size_s1 = sizeof(s1);
    size_s2 = sizeof(s2);

    printf(" 默认对齐方式下 size_s1 = %d\n", size_s1);
    printf(" 按字节对齐方式下 size_s2 = %d\n", size_s2);
}

```

输出结果：

```

默认对齐方式下 size_s1 = 8
按字节对齐方式下 size_s2 = 5

```

63. 复数相加

使用结构体（struct）将两个复数相加。

我们把形如 $a+bi$ （ a, b 均为实数）的数称为复数，其中 a 称为实部， b 称为虚部， i 称为虚数单位。

```

#include <stdio.h>

typedef struct complex
{
    float real;
    float imag;
} complex;

complex add(complex n1, complex n2);

int main()
{
    complex n1, n2, temp;

    printf("第一个复数 \n");
    printf("输入实部和虚部:\n");
    scanf("%f %f", &n1.real, &n1.imag);

    printf("\n 第二个复数 \n");
    printf("输入实部和虚部:\n");
    scanf("%f %f", &n2.real, &n2.imag);
}

```

```

    temp = add(n1, n2);
    printf("Sum = %.1f + %.1fi", temp.real, temp.imag);

    return 0;
}

complex add(complex n1, complex n2)
{
    complex temp;

    temp.real = n1.real + n2.real;
    temp.imag = n1.imag + n2.imag;

    return(temp);
}

```

输出结果:

第一个复数
输入实部和虚部:
2.3 4.5

第二个复数
输入实部和虚部:
3.4 5
Sum = 5.7 + 9.5i

64. 计算两个时间段的差值

```

#include <stdio.h>

struct TIME
{
    int seconds;
    int minutes;
    int hours;
};

void differenceBetweenTimePeriod(struct TIME t1, struct TIME t2, struct
TIME *diff);

int main()
{
    struct TIME startTime, stopTime, diff;

```



```

    printf("输入开始时间: \n");
    printf("输入小时、分钟、秒: ");
    scanf("%d %d %d", &startTime.hours, &startTime.minutes,
&startTime.seconds);

    printf("输入停止时间: \n");
    printf("输入小时、分钟、秒: ");
    scanf("%d %d %d", &stopTime.hours, &stopTime.minutes,
&stopTime.seconds);

    // 计算差值
    differenceBetweenTimePeriod(startTime, stopTime, &diff);

    printf("\n 差值: %d:%d:%d - ", startTime.hours, startTime.minutes,
startTime.seconds);
    printf("%d:%d:%d ", stopTime.hours, stopTime.minutes,
stopTime.seconds);
    printf("= %d:%d:%d\n", diff.hours, diff.minutes, diff.seconds);

    return 0;
}

void differenceBetweenTimePeriod(struct TIME start, struct TIME stop,
struct TIME *diff)
{
    if(stop.seconds > start.seconds){
        --start.minutes;
        start.seconds += 60;
    }

    diff->seconds = start.seconds - stop.seconds;
    if(stop.minutes > start.minutes){
        --start.hours;
        start.minutes += 60;
    }

    diff->minutes = start.minutes - stop.minutes;
    diff->hours = start.hours - stop.hours;
}

```

输出结果:

输入开始时间:

输入小时、分钟、秒：12 34 55

输入停止时间：

输入小时、分钟、秒：8 12 5

差值：12:34:55 - 8:12:5 = 4:22:50

65. 将字符串写入文件

```
#include <stdio.h>
#include <stdlib.h> /* exit() 函数 */

int main()
{
    char sentence[1000];
    FILE *fptr;

    fptr = fopen("runoob.txt", "w");
    if(fptr == NULL)
    {
        printf("Error!");
        exit(1);
    }

    printf("输入字符串:\n");
    fgets(sentence, (sizeof sentence / sizeof sentence[0]), stdin);

    fprintf(fptr, "%s", sentence);
    fclose(fptr);

    return 0;
}
```

输出结果：

输入字符串：

runoob.com

打开文件 runoob.txt：

```
$ cat runoob.txt
```

runoob.com

举例：非常简单的字符串写入文件

```

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

int main()
{
    char msg[1000];
    FILE *stream;
    int i=0;
    if ((stream = fopen("G:/text/abc.txt", "w")) == NULL)
    {
        perror ("fail to write");
        exit (1);
    }
    scanf("%s", msg);    //控制台键入内容到文件中
    while(msg[i])
    {
        fputc(msg[i], stream);
        i++;
    }
    return 0;
}

```

C 语言实现读取一个 txt 文件里的数据，要按行读出来：

1、打开文件 fopen("需要打开的路径")

2、然后使用 fgets 函数读取行

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX_LINE 1024
int main()
{
    char buf[MAX_LINE]; /*缓冲区*/
    FILE *fp;           /*文件指针*/
    int len;            /*行字符个数*/
    if((fp = fopen("test.txt", "r")) == NULL)
    {
        perror("fail to read");
        exit (1) ;
    }
    while(fgets(buf, MAX_LINE, fp) != NULL)
    {
        len = strlen(buf);
    }
}

```

```

        buf[len-1] = '\0'; /*去掉换行符*/
        printf("%s %d \n", buf, len - 1);
    }
    return 0;
}

```

66. 从文件中读取一行

(1) 从文件中读取一行

文件 runoob.txt 内容:

```

$ cat runoob.txt
runoob.com
google.com

```

```

#include <stdio.h>
#include <stdlib.h> // exit() 函数
int main()
{
    char c[1000];
    FILE *fptr;

    if ((fptr = fopen("runoob.txt", "r")) == NULL)
    {
        printf("Error! opening file");
        // 文件指针返回 NULL 则退出
        exit(1);
    }

    // 读取文本，直到碰到新的一行开始
    fscanf(fptr, "%[^\n]", c);

    printf("读取内容:\n%s", c);
    fclose(fptr);

    return 0;
}

```

输出结果:

读取内容:
runoob.com

(2) 从文件中读取多行, 学会使用 fgets 函数 1

```
#include "stdlib.h"
#include "stdio.h"

int main(int argc, char *argv[])
{
    FILE *in= fopen("D:/in.java", "r");
    char buf[1024];

    while (fgets(buf, sizeof(buf), in) != NULL)
    {
        printf("%s", buf);
    }

    fclose(in);

    return 0;
}
```

同理, 如果要实现, 文件逐行写入到另一个文件, 可以使用 fputs 函数即可

(3) 从文件中读取多行, 学会使用 fgets 函数 2

```
#include "stdlib.h"
#include "stdio.h"

int main(int argc, char *argv[])
{
    FILE *in= fopen("D:/In.java", "r");
    FILE *out = fopen("D:/Out.java", "w");

    char buf[1024];

    while (fgets(buf, sizeof(buf), in) != NULL)
    {
        fputs(buf, out);
    }

    fclose(in);
}
```

```

    fclose(out);

    return 0;
}

```

67. 输出当前文件执行代码

输出当前文件执行代码，`__FILE__` 为当前执行的文件常量

```

#include <stdio.h>
int main() {
    FILE *fp;
    char c;
    fp = fopen(__FILE__, "r");
    do {
        c = getc(fp);
        putchar(c);
    }
    while(c != EOF);
    fclose(fp);
    return 0;
}

```

输出结果：

```

#include <stdio.h>
int main() {
    FILE *fp;
    char c;
    fp = fopen(__FILE__, "r");
    do {
        c = getc(fp);
        putchar(c);
    }
    while(c != EOF);
    fclose(fp);
    return 0;
}

```

68. 约瑟夫生者死者小游戏

30 个人在一条船上，超载，需要 15 人下船。

于是人们排成一队，排队的位置即为他们的编号。

报数，从 1 开始，数到 9 的人下船。

如此循环，直到船上仅剩 15 人为止，问都有哪些编号的人下船了呢？

```
#include<stdio.h>

int c = 0;
int i = 1;
int j = 0;
int a[30] = { 0 };
int b[30] = { 0 };

int main()
{
    while (i<=31)
    {
        if (i == 31)
        {
            i = 1;
        }
        else if (c == 15)
        {
            break;
        }
        else
        {
            if (b[i] != 0)
            {
                i++;
                continue;
            }
            else
            {
                j++;
                if (j != 9)
                {
                    i++;
                    continue;
                }
                else
                {
                    b[i] = 1;
                    a[i] = j;
                }
            }
        }
    }
}
```



```

int n, x, j, k, l, m;
for (n=5;;n++) {
    j=4*(n-1)/5;
    k=4*(j-1)/5;
    l=4*(k-1)/5;
    m=4*(l-1)/5;
    if (n%5==1&& j%5==1&& k%5==1&& l%5==1&& m%5==1) {
        printf("至少合伙捕鱼: %d 条\n", n);
        printf("分别见到鱼的条数: %d %d %d %d\n", j, k, l, m);
        break;
    }
}
return 0;
}

```

输出结果:

至少合伙捕鱼: 3121 条

分别见到鱼的条数: 2496 1996 1596 1276

=====进阶例题 100 道=====

1. 有 1、2、3、4 个数字，能组成多少个互不相同且无重复数字的三位数？都是多少？

程序分析：可填在百位、十位、个位的数字都是 1、2、3、4。组成所有的排列后再去 掉不满足条件的排列

```
#include<stdio.h>

int main()
{
    int i, j, k;
    printf("\n");
    for(i=1; i<5; i++) { // 以下为三重循环
        for(j=1; j<5; j++) {
            for (k=1; k<5; k++) { // 确保 i、j、k 三位互不相同
                if (i!=k&&i!=j&&j!=k) {
                    printf("%d, %d, %d\n", i, j, k);
                }
            }
        }
    }
}
```

输出结果：

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

3, 4, 1
3, 4, 2
4, 1, 2
4, 1, 3
4, 2, 1
4, 2, 3
4, 3, 1
4, 3, 2

2. 企业发放的奖金根据利润提成

- 利润(I)低于或等于 10 万元时，奖金可提 10%;
- 利润高于 10 万元，低于 20 万元时，低于 10 万元的部分按 10%提成，高于 10 万元的部分，可提成 7.5%;
- 20 万到 40 万之间时，高于 20 万元的部分，可提成 5%;
- 40 万到 60 万之间时高于 40 万元的部分，可提成 3%;
- 60 万到 100 万之间时，高于 60 万元的部分，可提成 1.5%;
- 高于 100 万元时，超过 100 万元的部分按 1%提成。

从键盘输入当月利润 I，求应发放奖金总数？

程序分析：请利用数轴来分界，定位。注意定义时需把奖金定义成长整型。

```
#include<stdio.h>
int main()
{
    double i;
    double bonus1, bonus2, bonus4, bonus6, bonus10, bonus;
    printf("你的净利润是: \n");
    scanf("%lf", &i);
    bonus1=100000*0.1;
    bonus2=bonus1+100000*0.075;
    bonus4=bonus2+200000*0.05;
    bonus6=bonus4+200000*0.03;
    bonus10=bonus6+400000*0.015;
    if(i<=100000) {
        bonus=i*0.1;
    } else if(i<=200000) {
        bonus=bonus1+(i-100000)*0.075;
    } else if(i<=400000) {
        bonus=bonus2+(i-200000)*0.05;
    } else if(i<=600000) {
        bonus=bonus4+(i-400000)*0.03;
    } else if(i<=1000000) {
```

```

        bonus=bonus6+(i-600000)*0.015;
    } else if(i>1000000) {
        bonus=bonus10+(i-1000000)*0.01;
    }
    printf("提成为: bonus=%lf", bonus);

    printf("\n");
}

```

输出结果:

你的净利润是:
 120000
 提成为: bonus=11500.000000

(1) 优化如下:

```

#include <stdio.h>

#define WAN 10000

int main()
{
    double I = 0; // 利润
    double B = 0; // 奖金

    scanf("%lf", &I);
    I /= WAN;

    if (I > 100 * WAN)
    {
        B += ((I - 100) * 0.01);
        I = 100;
    }

    if (I > 60)
    {
        B += ((I - 60) * 0.015);
        I = 60;
    }

    if (I > 40)
    {
        B += ((I - 40) * 0.03);
        I = 40;
    }
}

```

```

    }

    if (I > 20)
    {
        B += ((I - 20) * 0.05);
        I = 20;
    }

    if (I > 10)
    {
        B += ((I - 10) * 0.075);
        I = 10;
    }

    B += (I * 0.1);

    printf("%lf", B);
}

```

(2) 使用循环优化代码的适用性:

```

#include<stdio.h>
int main()
{
    int i;
    double lirun;
    double jiangjin = 0;
    float fanwei[] = {100000, 200000, 400000, 600000, 1000000};
    float ticheng[] = {0.1, 0.075, 0.05, 0.03, 0.015, 0.01};
    printf("您好, 请问您的净利润是多少? \n");
    scanf("%lf", &lirun);
    for (i=0;i<5;i++)
    {
        if (lirun < fanwei[i])
        {
            jiangjin += lirun * ticheng[i];
            break;
        }
        else
        {
            jiangjin += fanwei[i] * ticheng[i];
            lirun -= fanwei[i];
        }
    }
}

```

```

    }
    printf("奖金是%.2lf\n", jiangjin);

    return 0;
}

```

(3) 利用 switch 的击穿现象

```

#include <stdio.h>

int main() {
    double d;
    int money = 100000;
    float res=0.0;
    int flag;
    scanf("%lf",&d);
    flag = (int)(d/money);
    flag = flag >10?10:flag;
    switch(flag) {
        case 10:
            res += (d-10*money)*0.01;
            d = 10*money;
        case 9:
        case 8:
        case 7:
        case 6:
            res += (d-6*money)*0.015;
            d = 6*money;
        case 5:
        case 4:
            res+= (d-4*money)*0.03;
            d = 4*money;
        case 3:
        case 2:
            res += (d-2*money)*0.05;
            d = 2*money;
        case 1:
            res += (d-money)*0.075;
            d = money;
        case 0:
            res += d *0.1;
    }
}

```

```

printf("%.2f\n",res);
return 0;
}

```

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？

程序分析：

假设该数为 x 。

- 1、则： $x + 100 = n^2$, $x + 100 + 168 = m^2$
- 2、计算等式： $m^2 - n^2 = (m + n)(m - n) = 168$
- 3、设置： $m + n = i$, $m - n = j$, $i * j = 168$, i 和 j 至少一个是偶数
- 4、可得： $m = (i + j) / 2$, $n = (i - j) / 2$, i 和 j 要么都是偶数，要么都是奇数。
- 5、从 3 和 4 推导可知道， i 与 j 均是大于等于 2 的偶数。
- 6、由于 $i * j = 168$, $j \geq 2$, 则 $1 < i < 168 / 2 + 1$ 。
- 7、接下来将 i 的所有数字循环计算即可。

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

```

int main (void)
{
    int i, j, m, n, x;
    for (i = 1; i < 168 / 2 + 1; i++)
    {
        if (168 % i == 0)
        {
            j = 168 / i;
            if (i > j && (i + j) % 2 == 0 && (i - j) % 2 == 0)
            {
                m = (i + j) / 2;
                n = (i - j) / 2;
                x = n * n - 100;
            }
        }
    }
}

```

```

        printf ("%d + 100 = %d * %d\n", x, n, n);
        printf ("%d + 268 = %d * %d\n", x, m, m);
    }
}
}
return 0;
}

```

输出结果：

```

-99 + 100 = 1 * 1
-99 + 268 = 13 * 13
21 + 100 = 11 * 11
21 + 268 = 17 * 17
261 + 100 = 19 * 19
261 + 268 = 23 * 23
1581 + 100 = 41 * 41
1581 + 268 = 43 * 43

```

4. 输入某年某月某日，判断这一天是这一年的第几天？

程序分析：以3月5日为例，应该先把前两个月的加起来，然后再加上5天即本年的第几天，特殊情况，闰年且输入月份大于3时需考虑多加一天

```

#include <stdio.h>
int main()
{
    int day, month, year, sum, leap;
    printf("\n 请输入年、月、日，格式为： 年, 月, 日 (2015, 12, 10) \n");
    scanf("%d, %d, %d", &year, &month, &day); // 格式为： 2015, 12, 10
    switch(month) // 先计算某月以前月份的总天数
    {
        case 1: sum=0; break;
        case 2: sum=31; break;
        case 3: sum=59; break;
        case 4: sum=90; break;
        case 5: sum=120; break;
        case 6: sum=151; break;
        case 7: sum=181; break;
        case 8: sum=212; break;
        case 9: sum=243; break;
        case 10: sum=273; break;
    }
}

```



```

        case 11:sum=304;break;
        case 12:sum=334;break;
        default:printf("data error");break;
    }
    sum=sum+day; // 再加上某天的天数
    if(year%400==0||(year%4==0&&year%100!=0)) { // 判断是不是闰年
        leap=1;
    } else {
        leap=0;
    }
    if(leap==1&&month>2) { // *如果是闰年且月份大于 2, 总天数应该加一天
        sum++;
    }
    printf("这是这一年的第 %d 天。", sum);
    printf("\n");
}

```

输出结果:

请输入年、月、日，格式为：年, 月, 日 (2015, 12, 10)
 2015, 10, 1
 这是这一年的第 274 天。

5. 输入三个整数 x, y, z，请把这三个数由小到大输出

程序分析：我们想办法把最小的数放到 x 上，先将 x 与 y 进行比较，如果 x>y 则将 x 与 y 的值进行交换，然后再用 x 与 z 进行比较，如果 x>z 则将 x 与 z 的值进行交换，这样能使 x 最小

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include <stdio.h>

```

```

int main()
{
    int x, y, z, t;
    printf("\n 请输入三个数字:\n");
    scanf("%d%d%d", &x, &y, &z);
    if (x>y) { /*交换 x, y 的值*/
        t=x;x=y;y=t;
    }
}

```

```

    }
    if(x>z) { /*交换 x, z 的值*/
        t=z;z=x;x=t;
    }
    if(y>z) { /*交换 z, y 的值*/
        t=y;y=z;z=t;
    }
    printf("从小到大排序: %d %d %d\n", x, y, z);
}

```

输出结果:

请输入三个数字:

1

3

2

从小到大排序: 1 2 3

6. 用*号输出字母 C 的图案

程序分析: 可先用'/'号在纸上写出字母 C, 再分行输出

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include "stdio.h"
int main()
{
    printf("用 * 号输出字母 C!\n");
    printf(" ****\n");
    printf(" * \n");
    printf(" * \n");
    printf(" ****\n");
}

```

输出结果:

用 * 号输出字母 C!

```

****
*
*
****

```

7. 输出特殊图案，请在 c 环境中运行，看一看，Very Beautiful!

程序分析：字符共有 256 个。不同字符，图形不一样。

VC6.0 下出现中文乱码(原因+解决方法)：

176 的 16 进制是 B0，219 的 16 进制是 DB，0xB0DB 是“佰”字的内码，所以输出的就是“佰”了。

主要原因是文件信息的代码页不同，我们所使用的操作系统中文状态下的代码页，要显示扩展的 ASCII 码需要在 437 OEM-美国这个下面显示，这样就可以显示出你所希望的。具体修改控制台的默认代码页步骤如下：

- 1. 点击运行界面左上角标题栏图标【c:\】，选择默认值一项
- 2. 修改默认代码页，936（ANSI/OEM-简体中文 GBK）为 437 OEM-美国
- 3、关闭后重新运行一下即可

```
// Created by www.runoob.com on 15/11/9.  
// Copyright © 2015 年 菜鸟教程. All rights reserved.  
//
```

```
#include<stdio.h>  
int main()  
{  
    char a=176,b=219;  
    printf("%c%c%c%c%c\n", b, a, a, a, b);  
    printf("%c%c%c%c%c\n", a, b, a, b, a);  
    printf("%c%c%c%c%c\n", a, a, b, a, a);  
    printf("%c%c%c%c%c\n", a, b, a, b, a);  
    printf("%c%c%c%c%c\n", b, a, a, a, b);  
    return 0;  
}
```

输出结果：

8. 输出 9*9 口诀

程序分析：分行与列考虑，共 9 行 9 列，i 控制行，j 控制列

```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

#include<stdio.h>
int main()
{
    int i,j,result;
    printf("\n");
    for (i=1;i<10;i++)
    {
        for(j=1;j<=i;j++)
        {
            result=i*j;
            printf("%d*%d=%-3d", i, j, result); /*-3d 表示左对齐，占 3 位
*/
        }
        printf("\n"); /*每一行后换行*/
    }
}
```

输出结果：

```
1*1=1
2*1=2  2*2=4
3*1=3  3*2=6  3*3=9
4*1=4  4*2=8  4*3=12 4*4=16
5*1=5  5*2=10 5*3=15 5*4=20 5*5=25
6*1=6  6*2=12 6*3=18 6*4=24 6*5=30 6*6=36
7*1=7  7*2=14 7*3=21 7*4=28 7*5=35 7*6=42 7*7=49
8*1=8  8*2=16 8*3=24 8*4=32 8*5=40 8*6=48 8*7=56 8*8=64
9*1=9  9*2=18 9*3=27 9*4=36 9*5=45 9*6=54 9*7=63 9*8=72 9*9=81
```

9. 要求输出国际象棋棋盘

程序分析：国际象棋棋盘由 64 个黑白相间的格子组成，分为 8 行*8 列。用 i 控制行，j 来控制列，根据 i+j 的和的变化来控制输出黑方格，还是白方格

```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

#include<stdio.h>
```

```

int main()
{
    int i, j;
    for(i=0; i<8; i++)
    {
        for(j=0; j<8; j++)
            if((i+j)%2==0)
                printf("%c%c", 219, 219);
            else printf("  ");
        printf("\n");
    }
    return 0;
}

```

输出结果：

10. 打印楼梯，同时在楼梯上方打印两个笑脸

程序分析：用 ASCII 1 来输出笑脸；用 i 控制行，j 来控制列，j 根据 i 的变化来控制输出黑方格的个数

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include<stdio.h>

```

```

int main()
{
    int i, j;
    printf("\1\1\n"); /*输出两个笑脸*/
    for(i=1; i<11; i++)
    {

```

```

        for(j=1;j<=i;j++)
            printf("%c%c", 219, 219);
        printf("\n");
    }
    return 0;
}

```

输出结果：

11. 古典问题（兔子生崽）

题目：古典问题（兔子生崽）：有一对兔子，从出生后第 3 个月起每个月都生一对兔子，小兔子长到第三个月后每个月又生一对兔子，假如兔子都不死，问每个月的兔子总数为多少？（输出前 40 个月即可）

程序分析：兔子的规律为数列 1, 1, 2, 3, 5, 8, 13, 21....，即下个月是上两个月之和（从第三个月开始）。

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include<stdio.h>

int main()
{
    int f1=1, f2=1, i;
    for(i=1; i<=20; i++)
    {
        printf("%12d%12d", f1, f2);
        if(i%2==0) printf("\n");
        f1=f1+f2;
        f2=f1+f2;
    }
}

```

```

    return 0;
}

```

输出结果:

1	1	2	3
5	8	13	21
34	55	89	144
233	377	610	987
1597	2584	4181	6765
10946	17711	28657	46368
75025	121393	196418	317811
514229	832040	1346269	2178309
3524578	5702887	9227465	14930352
24157817	39088169	63245986	102334155

12. 判断 101 到 200 之间的素数

程序分析：判断素数的方法：用一个数分别去除 2 到 sqrt(这个数)，如果能被整除， 则表明此数不是素数，反之是素数。

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include <stdio.h>

```

```

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

    for (i=101; i<=200; i++)
    {
        for (j=2; j<i; j++)
        {
            // 如果 j 能被 i 整出在跳出循环
            if (i%j==0)
                break;
        }
        // 判断循环是否提前跳出，如果 j<i 说明在 2~j 之间, i 有可整出的数
        if (j>=i)
        {

```

```

        count++;
        printf("%d ", i);
    // 换行，用 count 计数，每五个数换行
        if (count % 5 == 0)
            printf("\n");
    }
}
return 0;
}

```

输出结果：

```

101 103 107 109 113
127 131 137 139 149
151 157 163 167 173
179 181 191 193 197
199

```

C 语言经典 100 例 C 语言经典 100 例

13. 打印出所有的“水仙花数”

题目：打印出所有的“水仙花数”，所谓“水仙花数”是指一个三位数，其各位数字立方和等于该数 本身。例如：153 是一个“水仙花数”，因为 $153=1$ 的三次方+5 的三次方+3 的三次方。

程序分析：利用 for 循环控制 100-999 个数，每个数分解出个位，十位，百位。

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include<stdio.h>

int main()
{
    int i, x, y, z;
    for(i=100;i<1000;i++)
    {
        x=i%10;
        y=i/10%10;
        z=i/100%10;

        if(i==(x*x*x+y*y*y+z*z*z))
            printf("%d\n", i);
    }
}

```



```

    }
    return 0;
}

```

输出结果:

```

153
370
371
407

```

14. 将一个正整数分解质因数

题目：将一个正整数分解质因数。例如：输入 90, 打印出 $90=2*3*3*5$ 。

程序分析：对 n 进行分解质因数，应先找到一个最小的质数 k ，然后按下述步骤完成：

- (1) 如果这个质数恰等于（小于的时候，继续执行循环） n ，则说明分解质因数的过程已经结束，另外 打印出即可。
- (2) 但 n 能被 k 整除，则应打印出 k 的值，并用 n 除以 k 的商, 作为新的正整数 n . 重复执行第二步。
- (3) 如果 n 不能被 k 整除，则用 $k+1$ 作为 k 的值, 重复执行第一步。

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include<stdio.h>
int main()
{
    int n,i;
    printf("请输入整数: ");
    scanf("%d",&n);
    printf("%d=",n);
    for(i=2;i<=n;i++)
    {
        while(n%i==0)
        {
            printf("%d",i);
            n/=i;
            if(n!=1) printf("*");
        }
    }
}

```

```

    }

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

```

输出结果：

请输入整数： 90
90=2*3*3*5

15. 利用条件运算符的嵌套来完成此题：学习成绩 ≥ 90 分的同学用 A 表示，60-89 分之间的用 B 表示，60 分以下的用 C 表示

程序分析：(a>b)?a:b 这是条件运算符的基本例子。

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include<stdio.h>
int main()
{
    int score;
    char grade;
    printf("请输入分数： ");
    scanf("%d",&score);
    grade=(score>=90)?'A':((score>=60)?'B':'C');
    printf("%c\n",grade);
    return 0;
}

```

输出结果：

请输入分数： 87
B

16. 最大公约数和最小公倍数

题目：输入两个正整数 m 和 n ，求其最大公约数和最小公倍数。

程序分析：

(1) 最小公倍数=输入的两个数之积除于它们的最大公约数，关键是求出最大公约数；

(2) 求最大公约数用辗转相除法（又名欧几里德算法）

1) 证明：设 c 是 a 和 b 的最大公约数，记为 $c=\gcd(a, b)$, $a \geq b$,

令 $r=a \bmod b$

设 $a=kc$, $b=jc$, 则 k, j 互素，否则 c 不是最大公约数

据上, $r=a-mb=kc-mjc=(k-mj)c$

可知 r 也是 c 的倍数，且 $k-mj$ 与 j 互素，否则与前述 k, j 互素矛盾，

由此可知， b 与 r 的最大公约数也是 c ，即 $\gcd(a, b)=\gcd(b, a \bmod b)$ ，得证。

2) 算法描述：

第一步： $a \div b$ ，令 r 为所得余数 ($0 \leq r$) 第二步：互换：置 $a \leftarrow b$, $b \leftarrow r$ ，并返回第一步。

```
// Created by www.runoob.com on 15/11/9.  
// Copyright © 2015 年 菜鸟教程. All rights reserved.  
//
```

```
#include<stdio.h>  
int main()  
{  
    int a,b,t,r,n;  
    printf("请输入两个数字: \n");  
    scanf("%d %d",&a,&b);  
    if(a<b)  
        {t=b;b=a;a=t;}  
    r=a%b;  
    n=a*b;  
    while(r!=0)  
    {  
        a=b;  
        b=r;  
        r=a%b;  
    }  
    printf("这两个数的最大公约数是%d, 最小公倍数是%d\n",b,n/b);  
  
    return 0;  
}
```

输出结果:

请输入两个数字:

12 26

这两个数的最大公约数是 2，最小公倍数是 156

17. 输入一行字符，分别统计出其中英文字母、空格、数字和其它字符的个数

程序分析：利用 while 语句, 条件为输入的字符不为 '\n'。

```
// Created by www.runoob.com on 15/11/9.  
// Copyright © 2015 年 菜鸟教程. All rights reserved.  
//
```

```
#include<stdio.h>  
int main()  
{  
    char c;  
    int letters=0, spaces=0, digits=0, others=0;  
    printf("请输入一些字母: \n");  
    while((c=getchar())!='\n')  
    {  
        if((c>='a' && c<='z') || (c>='A' && c<='Z'))  
            letters++;  
        else if(c>='0' && c<='9')  
            digits++;  
        else if(c==' ')  
            spaces++;  
        else  
            others++;  
    }  
    printf("字母=%d, 数字=%d, 空格=%d, 其他  
=%d\n", letters, digits, spaces, others);  
    return 0;  
}
```

输出结果:

请输入一些字母:

www.runoob.com 123

字母=12, 数字=3, 空格=1, 其他=2

18.：求 $s=a+aa+aaa+aaaa+aa...a$ 的值

题目：求 $s=a+aa+aaa+aaaa+aa...a$ 的值，其中 a 是一个数字。例如 $2+22+222+2222+22222$ (此时共有 5 个数相加)，几个数相加有键盘控制。

程序分析：关键是计算出每一项的值。

```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

#include<stdio.h>
int main()
{
    int s=0,a,n,t;
    printf("请输入 a 和 n: \n");
    scanf("%d%d",&a,&n);
    t=a;
    while(n>0)
    {
        s+=t;
        a=a*10;
        t+=a;
        n--;
    }
    printf("a+aa+...=%d\n",s);
    return 0;
}
```

输出结果：

```
请输入 a 和 n:
2 5
a+aa+...=24690
```

19. 一个数如果恰好等于它的因子之和，这个数就称为“完数”。例如 $6=1+2+3$ 。编程找出 1000 以内的所有完数

```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
```

```
//

#include<stdio.h>
#define N 1000
int main()
{
    int i, j, k, n, sum;
    int a[256];
    for(i=2; i<=N; i++)
    {
        sum=a[0]=1;
        k=0;
        for(j=2; j<=(i/2); j++)
        {
            if(i%j==0)
            {
                sum+=j;
                a[++k]=j;
            }
        }
        if(i==sum)
        {
            printf("%d=%d", i, a[0]);
            for(n=1; n<=k; n++)
                printf("+%d", a[n]);
            printf("\n");
        }
    }
    return 0;
}
```

输出结果：

```
6=1+2+3
28=1+2+4+7+14
496=1+2+4+8+16+31+62+124+248
```

20. 小球自由下落

题目：一球从 100 米高度自由落下，每次落地后反跳回原高度的一半；再落下，求它在第 10 次落地时，共经过多少米？第 10 次反弹多高？

程序分析：见下面注释。

```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

#include<stdio.h>
int main()
{
    float h,s;
    h=s=100;
    h=h/2; //第一次反弹高度
    for(int i=2;i<=10;i++)
    {
        s=s+2*h;
        h=h/2;
    }
    printf("第 10 次落地时，共经过%f 米，第 10 次反弹高%f 米\n",s,h);
    return 0;
}
```

输出结果：

第 10 次落地时，共经过 299.609375 米，第 10 次反弹高 0.097656 米

21. 猴子吃桃问题

题目：猴子吃桃问题：猴子第一天摘下若干个桃子，当即吃了一半，还不瘾，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 的一半零一个。到第 10 天早上想再吃时，见只剩下一个桃子了。求第一天共摘了多少。

程序分析：采取逆向思维的方法，从后往前推断。

1) 设 x_1 为前一天桃子数，设 x_2 为第二天桃子数， 则：

$$x_2 = x_1 / 2 - 1, \quad x_1 = (x_2 + 1) * 2$$

$$x_3 = x_2 / 2 - 1, \quad x_2 = (x_3 + 1) * 2$$

以此类推： $x_{前} = (x_{后} + 1) * 2$

2) 从第 10 天可以类推到第 1 天，是一个循环过程。

```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

#include <stdio.h>
#include <stdlib.h>
int main() {
    int day, x1 = 0, x2;
    day=9;
    x2=1;
    while(day>0) {
        x1=(x2+1)*2; // 第一天的桃子数是第 2 天桃子数加 1 后的 2 倍
        x2=x1;
        day--;
    }
    printf("总数为 %d\n", x1);

    return 0;
}
```

输出结果：

总数为 1534

22. 两个乒乓球队进行比赛，各出三人

题目：两个乒乓球队进行比赛，各出三人。甲队为 a, b, c 三人，乙队为 x, y, z 三人。已抽签决定比赛名单。有人向队员打听比赛的名单。a 说他不和 x 比，c 说他不和 x, z 比，请编程序找出三队赛手的名单。

```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

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

int main()
{
    char i, j, k;
    for(i='x'; i<='z'; i++) {
        for(j='x'; j<='z'; j++) {
            if(i!=j) {
                for(k='x'; k<='z'; k++) {
```



```

        if(i!=k&&j!=k) {
            if(i!='x' &&k!='x' &&k!='z') {
                printf("顺序为:
a--%c\tb--%c\tc--%c\n", i, j, k);
            }
        }
    }
}

```

输出结果:

顺序为: a--z b--x c--y

23. 打印出如下图案（菱形）

程序分析:

先把图形分成两部分来看待，前四行一个规律，后三行一个规律，利用双重 for 循环，第一层控制行，第二层控制列。

```

    *
   ***
  *****
 *****
  *****
   ***
    *

```

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include <stdio.h>
int main()
{
    int i, j, k;
    for(i=0; i<=3; i++) {
        for(j=0; j<=2-i; j++) {
            printf(" ");

```

```

    }
    for(k=0;k<=2*i;k++) {
        printf("*");
    }
    printf("\n");
}
for(i=0;i<=2;i++) {
    for(j=0;j<=i;j++) {
        printf(" ");
    }
    for(k=0;k<=4-2*i;k++) {
        printf("*");
    }
    printf("\n");
}
}

```

输出结果：

```

*
***
*****
*****
*****
***
*

```

24.： 有一分数序列： 2/1， 3/2， 5/3， 8/5， 13/8， 21/13... 求出这个数列的前 20 项之和

程序分析： 请抓住分子与分母的变化规律。

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include <stdio.h>

```

```

int main()
{

```

```

int i, t;
float sum=0;
float a=2, b=1;
for(i=1; i<=20; i++)
{
    sum=sum+a/b;
    t=a;
    a=a+b;
    b=t;
}
printf("%9.6f\n", sum);
}

```

输出结果:

32.660263

25. 求 $1+2!+3!+\dots+20!$ 的和

程序分析：此程序只是把累加变成了累乘。

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include <stdio.h>

int main()
{
    int i;
    long double sum, mix;
    sum=0, mix=1;
    for(i=1; i<=20; i++)
    {
        mix=mix*i;
        sum=sum+mix;
    }
    printf("%Lf\n", sum);
}

```

输出结果:

2561327494111820313.000000

26. 利用递归方法求 5!

程序分析：递归公式： $fn=fn_1*4!$

```
// Created by www.runoob.com on 15/11/9.  
// Copyright © 2015 年 菜鸟教程. All rights reserved.  
//
```

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

```
int main()  
{  
    int i;  
    int fact(int);  
    for(i=0;i<6;i++){  
        printf("%d!=%d\n", i, fact(i));  
    }  
}  
int fact(int j)  
{  
    int sum;  
    if(j==0){  
        sum=1;  
    } else {  
        sum=j*fact(j-1);  
    }  
    return sum;  
}
```

输出结果：

```
0!=1  
1!=1  
2!=2  
3!=6  
4!=24  
5!=120
```

27. 利用递归函数调用方式，将所输入的 5 个字符，以相反顺序打印出来

```
// Created by www.runoob.com on 15/11/9.
```

```
// Copyright © 2015 年 菜鸟教程. All rights reserved.  
//
```

```
#include <stdio.h>  
  
int main()  
{  
    int i=5;  
    void palin(int n);  
    printf("请输入 5 个字符\40:\40");  
    palin(i);  
    printf("\n");  
}  
void palin(n)  
int n;  
{  
    char next;  
    if(n<=1) {  
        next=getchar();  
        printf("相反顺序输出结果\40:\40");  
        putchar(next);  
    } else {  
        next=getchar();  
        palin(n-1);  
        putchar(next);  
    }  
}
```

输出结果：

请输入 5 个字符 : abcde
相反顺序输出结果 : edcba

28. 有 5 个人坐在一起，问第五个人多少岁？

题目：有 5 个人坐在一起，问第五个人多少岁？他说比第 4 个人大 2 岁。问第 4 个人岁数，他说比第 3 个人大 2 岁。问第三个人，又说比第 2 人大两岁。问第 2 个人，说比第一个人大两岁。最后问第一个人， he 说是 10 岁。请问第五个人多大？

程序分析：利用递归的方法，递归分为回推和递推两个阶段。要想知道第五个人岁数，需知道第四人的岁数，依次类推，推到第一人（10 岁），再往回推。

```
// Created by www.runoob.com on 15/11/9.  
// Copyright © 2015 年 菜鸟教程. All rights reserved.  
//
```

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

```
int age(n)  
int n;  
{  
    int c;  
    if(n==1) c=10;  
    else c=age(n-1)+2;  
    return(c);  
}  
int main()  
{  
    printf("%d\n", age(5));  
}
```

输出结果:

18

29. 给一个不多于 5 位的正整数，要求： 一、求它是几位数，二、逆序打印出各位数字

程序分析：学会分解出每一位数，如下解释。

```
// Created by www.runoob.com on 15/11/9.  
// Copyright © 2015 年 菜鸟教程. All rights reserved.  
//
```

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

```
int main( )  
{  
    long a, b, c, d, e, x;  
    printf("请输入 5 位数字: ");  
    scanf("%ld", &x);  
    a=x/10000;      /*分解出万位*/  
    b=x%10000/1000; /*分解出千位*/
```

```

c=x%1000/100;    /*分解出百位*/
d=x%100/10;      /*分解出十位*/
e=x%10;          /*分解出个位*/
if (a!=0) {
    printf("为 5 位数, 逆序为:  %d %d %d %d %d\n", e, d, c, b, a);
} else if(b!=0) {
    printf("为 4 位数, 逆序为:  %d %d %d %d\n", e, d, c, b);
} else if(c!=0) {
    printf("为 3 位数, 逆序为: %d %d %d\n", e, d, c);
} else if(d!=0) {
    printf("为 2 位数, 逆序为:  %d %d\n", e, d);
} else if(e!=0) {
    printf("为 1 位数, 逆序为: %d\n", e);
}
}

```

输出结果:

请输入 5 位数字: 12345
 为 5 位数, 逆序为: 5 4 3 2 1

30. 回文数

题目: 一个 5 位数, 判断它是不是回文数。即 12321 是回文数, 个位与万位相同, 十位与千位相同。

程序分析: 学会分解出每一位数。

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

#include <stdio.h>

int main( )
{
    long ge, shi, qian, wan, x;
    printf("请输入 5 位数字: ");
    scanf("%ld", &x);
    wan=x/10000;    /*分解出万位*/
    qian=x%10000/1000; /*分解出千位*/
    shi=x%100/10;    /*分解出十位*/
    ge=x%10;         /*分解出个位*/
    if (ge==wan&&shi==qian) { /*个位等于万位并且十位等于千位*/

```

```

        printf("这是回文数\n");
    } else {
        printf("这不是回文数\n");
    }
}

```

输出结果：

请输入 5 位数字：12321
这是回文数

请输入 5 位数字：12345
这不是回文数

31. 请输入星期几的第一个字母来判断一下是星期几，如果第一个字母一样，则继续判断第二个字母

程序分析：用情况语句比较好，如果第一个字母一样，则判断用情况语句或 if 语句判断第二个字母。

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

#include<stdio.h>

int main()
{
    char i, j;
    printf("请输入第一个字母:\n");
    scanf("%c", &i);
    getchar(); //scanf("%c", &j); 的问题，第二次是读入的一个换行符，而不是输入的字符，因此需要加一个 getchar() 吃掉换行符
    switch(i)
    {
        case 'm':
            printf("monday\n");
            break;
        case 'w':
            printf("wednesday\n");
            break;
    }
}

```



```

        case 'f':
            printf("friday\n");
            break;
        case 't':
            printf("请输入下一个字母\n");
            scanf("%c",&j);
            if (j=='u') {printf("tuesday\n");break;}
            if (j=='h') {printf("thursday\n");break;}
        case 's':
            printf("请输入下一个字母\n");
            scanf("%c",&j);
            if (j=='a') {printf("saturday\n");break;}
            if (j=='u') {printf("sunday\n"); break;}
        default :
            printf("error\n"); break;
    }
    return 0;
}

```

输出结果:

请输入第一个字母:

s

请输入下一个字母

a

saturday

32. 删除一个字符串中的指定字母，如： 字符串 "aca"，删除其中的 a 字母

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

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

```

// 删除字符串中指定字母函数

```

char* deleteCharacters(char * str, char * charSet)
{
    int hash [256];
    if(NULL == charSet)

```

```

        return str;
    for(int i = 0; i < 256; i++)
        hash[i] = 0;
    for(int i = 0; i < strlen(charSet); i++)
        hash[charSet[i]] = 1;
    int currentIndex = 0;
    for(int i = 0; i < strlen(str); i++)
    {
        if(!hash[str[i]])
            str[currentIndex++] = str[i];
    }
    str[currentIndex] = '\0';
    return str;
}

int main()
{
    char s[2] = "a";    // 要删除的字母
    char s2[5] = "aca"; // 目标字符串
    printf("%s\n", deleteCharacters(s2, s));
    return 0;
}

```

输出结果：

c

33. 判断一个数字是否为质数

程序分析：质数（prime number）又称素数，有无限个。一个大于 1 的自然数，除了 1 和它本身外，不能被其他自然数整除。

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include<stdio.h>
#include<math.h>
#define MAX 1000

```

```

int prime[MAX];

```

```

int isPrimeNaive(int n)

```

```

{
    if(n <= 1)
        return 0;
    for(int i = 2; i < n; i++)
        if(n % i == 0)
            return 0;
    return 1;
}

int isPrime(int n)
{
    if(n<= 1)
        return 0;
    if(n == 2)
        return 1;
    if(n%2 == 0)
        return 0;
    int limit = (int)sqrt((double)n);
    for(int i = 3; i <= limit; i=i+2)
    {
        if(n % i == 0)
            return 0;
    }
    return 1;
}

void sieve()
{
    prime[0] = 0;
    prime[1] = 0;
    for(int i = 2; i < MAX; i++)
        prime[i] = 1;
    int limit = (int)sqrt((double)MAX);
    for(int i = 2; i <= limit; i++)
    {
        if(prime[i])
            for(int j = i*i; j <= MAX; j+=i)
                prime[j] = 0;
    }
}

int isPrimeSieve(int n)
{
    if(prime[n])

```

```

        return 1;
    else
        return 0;
}

int main()
{
    sieve();
    printf("N=%d %d\n", 1, isPrime(1));
    printf("N=%d %d\n", 2, isPrime(2));
    printf("N=%d %d\n", 3, isPrime(3));
    printf("N=%d %d\n", 4, isPrime(4));
    printf("N=%d %d\n", 7, isPrime(7));
    printf("N=%d %d\n", 9, isPrime(9));
    printf("N=%d %d\n", 13, isPrime(13));
    printf("N=%d %d\n", 17, isPrime(17));
    printf("N=%d %d\n", 100, isPrime(100));
    printf("N=%d %d\n", 23, isPrime(23));
    printf("N=%d %d\n", 1, isPrime(1));
    return 0;
}

```

输出结果：

```

N=1 0
N=2 1
N=3 1
N=4 0
N=7 1
N=9 0
N=13 1
N=17 1
N=100 0
N=23 1
N=1 0

```

34. 练习函数调用

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include <stdio.h>
void hello_world(void)

```

```

{
    printf("Hello, world!\n");
}
void three_hellos(void)
{
    int counter;
    for (counter = 1; counter <= 3; counter++)
        hello_world();/*调用此函数*/
}
int main(void)
{
    three_hellos();/*调用此函数*/
}

```

输出结果:

```

Hello, world!
Hello, world!
Hello, world!

```

35. 字符串反转，如将字符串 “www.runoob.com” 反转为 “moc.boonur.www”

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include <stdio.h>

```

```

void reverse(char* s)
{
    // 获取字符串长度
    int len = 0;
    char* p = s;
    while (*p != 0)
    {
        len++;
        p++;
    }
}

```

```

// 交换 ...
int i = 0;
char c;
while (i <= len / 2 - 1)
{
    c = *(s + i);
    *(s + i) = *(s + len - 1 - i);
    *(s + len - 1 - i) = c;
    i++;
}

int main()
{
    char s[] = "www.runoob.com";
    printf("' %s' =>\n", s);
    reverse(s);          // 反转字符串
    printf("' %s' \n", s);
    return 0;
}

```

输出结果：

```

' www.runoob.com' =>
' moc.boonur.www'

```

36. 求 100 之内的素数

题目：求 100 之内的素数。

程序分析：质数（prime number）又称素数，有无限个。一个大于 1 的自然数，除了 1 和它本身外，不能被其他自然数整除。

输出结果：

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

#include<stdio.h>
#include<math.h>
int main()
{
    int i, j, k, n=0;

```

```

for(i=2;i<=100;i++)
{
    k=(int)sqrt(i);
    for(j=2;j<=k;j++)
        if(i%j==0) break;
    if(j>k)
    {
        printf("%d ",i);
        n++;
        if(n%5==0)
            printf("\n");
    }
}
return 0;
}

```

输出结果：

```

2 3 5 7 11
13 17 19 23 29
31 37 41 43 47
53 59 61 67 71
73 79 83 89 97

```

37. 对 10 个数进行排序

程序分析：可以利用选择法，即从后 9 个比较过程中，选择一个最小的与第一个元素交换，下次类推，即用第二个元素与后 8 个进行比较，并进行交换

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include<stdio.h>
#define N 10
int main()
{
    int i, j, a[N], temp;
    printf("请输入 10 个数字: \n");
    for(i=0; i<N; i++)
        scanf("%d", &a[i]);
    for(i=0; i<N-1; i++)
    {
        int min=i;

```

```

        for(j=i+1;j<N;j++)
            if(a[min]>a[j]) min=j;
        if(min!=i)
        {
            temp=a[min];
            a[min]=a[i];
            a[i]=temp;
        }
    }
    printf("排序结果是:\n");
    for(i=0;i<N;i++)
        printf("%d ",a[i]);
    printf("\n");
    return 0;
}

```

输出结果：

请输入 10 个数字：

23 2 27 98 234 1 4 90 88 34

排序结果是：

1 2 4 23 27 34 88 90 98 234

38. 求一个 3*3 矩阵对角线元素之和

程序分析：利用双重 for 循环控制输入二维数组，再将 a[i][i] 累加后输出

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include<stdio.h>
#define N 3
int main()
{
    int i, j, a[N][N], sum=0;
    printf("请输入矩阵(3*3): \n");
    for(i=0;i<N;i++)
        for(j=0;j<N;j++)
            scanf("%d",&a[i][j]);
    for(i=0;i<N;i++)
        sum+=a[i][i];
    printf("对角线之和为: %d\n", sum);
    return 0;
}

```



```
}
```

输出结果:

请输入矩阵(3*3):

1 2 3

4 5 6

7 8 9

对角线之和为: 15

39. 有一个已经排好序的数组。现输入一个数，要求按原来的规律将它插入数组中

程序分析: 首先判断此数是否大于最后一个数, 然后再考虑插入中间的数的情况, 插入后此元素之后的数, 依次后移一个位置。

```
// Created by www.runoob.com on 15/11/9.  
// Copyright © 2015 年 菜鸟教程. All rights reserved.  
//
```

```
#include<stdio.h>  
int main()  
{  
    int a[11]={1, 4, 6, 9, 13, 16, 19, 28, 40, 100};  
    int temp1, temp2, number, end, i, j;  
    printf("原始数组是:\n");  
    for(i=0; i<10; i++)  
        printf("%4d", a[i]);  
    printf("\n 插入一个新的数字: ");  
    scanf("%d", &number);  
    end=a[9];  
    if(number>end)  
        a[10]=number;  
    else  
    {  
        for(i=0; i<10; i++)  
        {  
            if(a[i]>number)  
            {  
                temp1=a[i];  
                a[i]=number;
```

```

        for(j=i+1;j<11;j++)
        {
            temp2=a[j];
            a[j]=temp1;
            temp1=temp2;
        }
        break;
    }
}
for(i=0;i<11;i++)
    printf("%4d",a[i]);
printf("\n");
return 0;
}

```

输出结果：

原始数组是：

1 4 6 9 13 16 19 28 40 100

插入一个新的数字：10

1 4 6 9 10 13 16 19 28 40 100

40. 将一个数组逆序输出

程序分析：用第一个与最后一个交换。

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include<stdio.h>
#define N 10
int main()
{
    int a[N]={0,1,2,3,4,5,6,7,8,9};
    int i,t;
    printf("原始数组是:\n");
    for(i=0;i<N;i++)
        printf("%d ",a[i]);
    for(i=0;i<N/2;i++)
    {
        t=a[i];
        a[i]=a[N-1-i];

```

```

        a[N-1-i]=t;
    }
    printf("\n 排序后的数组:\n");
    for(i=0;i<N;i++)
        printf("%d ",a[i]);
    printf("\n");
    return 0;
}

```

输出结果:

原始数组是:

0 1 2 3 4 5 6 7 8 9

排序后的数组:

9 8 7 6 5 4 3 2 1 0

41. 学习 static 定义静态变量的用法

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include<stdio.h>
int main()
{
    void fun();
    for(int i=0;i<3;i++)
        fun();
    return 0;
}
void fun()
{
    int i=0;
    static int static_i=0;
    printf("i=%d\n",i);
    printf("static_i=%d\n",static_i);
    i++;
    static_i++;
}

```

输出结果:

```

i=0
static_i=0

```

```
i=0
static_i=1
i=0
static_i=2
```

42. 学习使用 auto 定义变量的用法

```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//
```

```
#include <stdio.h>
int main()
{
    int i, num;
    num=2;
    for(i=0; i<3; i++)
    {
        printf("num 变量为 %d \n", num);
        num++;
        {
            auto int num=1;
            printf("内置模块 num 变量为 %d \n", num);
            num++;
        }
    }
    return 0;
}
```

输出结果:

```
num 变量为 2
内置模块 num 变量为 1
num 变量为 3
内置模块 num 变量为 1
num 变量为 4
内置模块 num 变量为 1
```

43. 学习使用 static 的另一用法

```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//
```

```

#include <stdio.h>
int main()
{
    int i,num;
    num=2;
    for(i=0;i<3;i++)
    {
        printf("num 变量为 %d \n",num);
        num++;
        {
            static int num=1;
            printf("内置模块 num 变量为 %d\n",num);
            num++;
        }
    }
    return 0;
}

```

输出结果:

```

num 变量为 2
内置模块 num 变量为 1
num 变量为 3
内置模块 num 变量为 2
num 变量为 4
内置模块 num 变量为 3

```

44. 学习使用 external 的用法

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include <stdio.h>
int a,b,c;
void add()
{
    int a;
    a=3;
    c=a+b;
}
int main()
{

```

```

    a=b+4;
    add();
    printf("c 的值为 %d\n",c);
    return 0;
}

```

输出结果:

c 的值为 7

45. 学习使用 register 定义变量的方法

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include <stdio.h>
int main()
{
    register int i;
    int tmp=0;
    for(i=1;i<=100;i++)
        tmp+=i;
    printf("总和为 %d\n",tmp);
    return 0;
}

```

输出结果:

总和为 5050

46. 宏#define 命令练习

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include<stdio.h>
#define TRUE 1
#define FALSE 0
#define SQ(x) (x)*(x)
int main()
{

```

```

int num;
int again=1;
printf("如果值小于 50 程序将终止。 \n");
while(again)
{
    printf("\n 请输入数字: ");
    scanf("%d", &num);
    printf("该数字的平方为 %d \n", SQ(num));
    if(num>=50)
        again=TRUE;
    else
        again=FALSE;
}
return 0;
}

```

输出结果:

如果值小于 50 程序将终止。

请输入数字: 100
该数字的平方为 10000

请输入数字: 5
该数字的平方为 25

47. 宏#define 命令练习 2

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

#include<stdio.h>
#define exchange(a,b) { int t;t=a;a=b;b=t;} //注意放在一行里
int main()
{
    int x=10;
    int y=20;
    printf("x=%d; y=%d\n", x, y);
    exchange(x, y);
    printf("x=%d; y=%d\n", x, y);
    return 0;
}

```

输出结果:

```
x=10; y=20
x=20; y=10
```

48. 宏#define 命令练习 3

```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//
```

```
#define LAG >
#define SMA <
#define EQ ==
#include <stdio.h>
int main()
{
    int i, j;
    printf("请输入两个数字: \n");
    scanf("%d %d", &i, &j);
    if(i LAG j)
        printf("%d 大于 %d \n", i, j);
    else if(i EQ j)
        printf("%d 等于 %d \n", i, j);
    else if(i SMA j)
        printf("%d 小于 %d \n", i, j);
    else
        printf("没有值. \n");
    return 0;
}
```

输出结果:

```
请输入两个数字:
1 2
1 小于 2
```

49. #if #ifdef 和#ifndef 的综合应用

```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//
```



```

#include<stdio.h>
#define MAX
#define MAXIMUM(x, y) (x>y)?x:y
#define MINIMUM(x, y) (x>y)?y:x
int main()
{
    int a=10, b=20;
#ifdef MAX
    printf("更大的数字是 %d\n", MAXIMUM(a, b));
#else
    printf("更小的数字是 %d\n", MINIMUM(a, b));
#endif
#ifdef MIN
    printf("更小的数字是 %d\n", MINIMUM(a, b));
#else
    printf("更大的数字是 %d\n", MAXIMUM(a, b));
#endif
#undef MAX
#ifdef MAX
    printf("更大的数字是 %d\n", MAXIMUM(a, b));
#else
    printf("更小的数字是 %d\n", MINIMUM(a, b));
#endif
#define MIN
#ifdef MIN
    printf("更小的数字是 %d\n", MINIMUM(a, b));
#else
    printf("更大的数字是 %d\n", MAXIMUM(a, b));
#endif
    return 0;
}

```

输出结果：

更大的数字是 20
 更小的数字是 10
 更小的数字是 10
 更大的数字是 20

50. #include 的应用练习

test.h 文件代码如下：

```
#define LAG >
```

```

#define SMA <
#define EQ ==

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

#include "test.h"
#include <stdio.h>

int main()
{
    int i=10;
    int j=20;
    if(i LAG j)
        printf("%d 大于 %d \n", i, j);
    else if(i EQ j)
        printf("%d 等于 %d \n", i, j);
    else if(i SMA j)
        printf("%d 小于 %d \n", i, j);
    else
        printf("没有值。 \n");
    return 0;
}

```

输出结果：

10 小于 20

51. 学习使用按位与 &

程序分析： 0&0=0; 0&1=0; 1&0=0; 1&1=1 。

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

#include <stdio.h>
int main()
{
    int a,b;
    a=077;

```

```

    b=a&3;
    printf("a & b(decimal) 为 %d \n",b);
    b&=7;
    printf("a & b(decimal) 为 %d \n",b);
    return 0;
}

```

输出结果：

```

a & b(decimal) 为 3
a & b(decimal) 为 3

```

52. 学习使用按位或 |

程序分析： $0|0=0$; $0|1=1$; $1|0=1$; $1|1=1$ 。

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include<stdio.h>
int main()
{
    int a,b;
    a=077;
    b=a|3;
    printf("b 的值为 %d \n",b);
    b|=7;
    printf("b 的值为 %d \n",b);
    return 0;
}

```

输出结果：

```

b 的值为 63
b 的值为 63

```

53. 学习使用按位异或 ^

程序分析： $0^0=0$; $0^1=1$; $1^0=1$; $1^1=0$ 。

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.

```

```
//

#include <stdio.h>
int main()
{
    int a,b;
    a=077;
    b=a^3;
    printf("b 的值为 %d \n",b);
    b^=7;
    printf("b 的值为 %d \n",b);
    return 0;
}
```

输出结果：

b 的值为 60
b 的值为 59

54. 取一个整数 a 从右端开始的 4~7 位

程序分析：可以这样考虑：

- (1)先使 a 右移 4 位。
- (2)设置一个低 4 位全为 1，其余全为 0 的数，可用 $\sim(\sim 0 \ll 4)$
- (3)将上面二者进行 & 运算。

```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//
```

```
#include <stdio.h>
int main()
{
    unsigned a,b,c,d;
    printf("请输入整数: \n");
    scanf("%u",&a);
    b=a>>4;
    c=~(\sim 0<<4);
    d=b&c;
```

```

    printf("%o\n%o\n", a, d);
    return 0;
}

```

输出结果:

请输入整数:

36

36

1

55. 学习使用按位取反~

程序分析: $\sim 0=1$; $\sim 1=0$;

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include <stdio.h>
int main()
{
    int a,b;
    a=234;
    b=~a;
    printf("a 的按位取反值为（十进制） %d \n",b);
    a=~a;
    printf("a 的按位取反值为（十六进制） %x \n",a);
    return 0;
}

```

输出结果:

请输入整数:

a 的按位取反值为（十进制） -235

a 的按位取反值为（十六进制） ffffffff15

56. 画图，学用 circle 画圆形

```

#include <graphics.h> //VC6.0 中是不能运行的，要在 Turbo2.0/3.0 中
int main()
{
    int driver,mode,i;

```

```

float j=1,k=1;
driver=VGA;
mode=VGAHI;
initgraph(&driver,&mode,"");
setbkcolor(YELLOW);
for(i=0;i<=25;i++)
{
    setcolor(8);
    circle(310,250,k);
    k=k+j;
    j=j+0.3;
}
return 0;
}

```

57. 画图，学用 line 画直线（在 TC 中实现）

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include "graphics.h"
int main()
{
    int driver,mode,i;
    float x0,y0,y1,x1;
    float j=12,k;
    driver=VGA;mode=VGAHI;
    initgraph(&driver,&mode,"");
    setbkcolor(GREEN);
    x0=263;y0=263;y1=275;x1=275;
    for(i=0;i<=18;i++)
    {
        setcolor(5);
        line(x0,y0,x0,y1);
        x0=x0-5;
        y0=y0-5;
        x1=x1+5;
        y1=y1+5;
        j=j+10;
    }
}

```

```
}
```

58. 学用 rectangle 画方形（在 TC 中实现）

```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

#include "graphics.h"
int main()
{
    int x0,y0,y1,x1,driver,mode,i;
    driver=VGA;mode=VGAHI;
    initgraph(&driver,&mode,"");
    setbkcolor(YELLOW);
    x0=263;y0=263;y1=275;x1=275;
    for(i=0;i<=18;i++)
    {
        setcolor(1);
        rectangle(x0,y0,x1,y1);
        x0=x0-5;
        y0=y0-5;
        x1=x1+5;
        y1=y1+5;
    }
    settextstyle(DEFAULT_FONT,HORIZ_DIR,2);
    outtextxy(150,40,"How beautiful it is!");
    line(130,60,480,60);
    setcolor(2);
    circle(269,269,137);
}
```

59. 画图，综合例子（在 TC 中实现）

```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

# define PAI 3.1415926
# define B 0.809
# include "graphics.h"
```

```

#include "math.h"
int main()
{
    int i, j, k, x0, y0, x, y, driver, mode;
    float a;
    driver=CGA;mode=CGAC0;
    initgraph(&driver, &mode, "");
    setcolor(3);
    setbkcolor(GREEN);
    x0=150;y0=100;
    circle(x0, y0, 10);
    circle(x0, y0, 20);
    circle(x0, y0, 50);
    for(i=0;i<16;i++)
    {
        a=(2*PAI/16)*i;
        x=ceil(x0+48*cos(a));
        y=ceil(y0+48*sin(a)*B);
        setcolor(2); line(x0, y0, x, y);
    }
    setcolor(3);circle(x0, y0, 60);
    /* Make 0 time normal size letters */
    setttextstyle(DEFAULT_FONT, HORIZ_DIR, 0);
    outtextxy(10, 170, "press a key");
    getch();
    setfillstyle(HATCH_FILL, YELLOW);
    floodfill(202, 100, WHITE);
    getch();
    for(k=0;k<=500;k++)
    {
        setcolor(3);
        for(i=0;i<=16;i++)
        {
            a=(2*PAI/16)*i+(2*PAI/180)*k;
            x=ceil(x0+48*cos(a));
            y=ceil(y0+48*sin(a)*B);
            setcolor(2); line(x0, y0, x, y);
        }
        for(j=1;j<=50;j++)
        {
            a=(2*PAI/16)*i+(2*PAI/180)*k-1;
            x=ceil(x0+48*cos(a));
            y=ceil(y0+48*sin(a)*B);
            line(x0, y0, x, y);
        }
    }
}

```



```

    }
}
restorecrtmode();
}

```

60. 画图，综合例子 2（在 TC 中实现）

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include "graphics.h"
#define LEFT 0
#define TOP 0
#define RIGHT 639
#define BOTTOM 479
#define LINES 400
#define MAXCOLOR 15
int main()
{
    int driver, mode, error;
    int x1, y1;
    int x2, y2;
    int dx1, dy1, dx2, dy2, i=1;
    int count=0;
    int color=0;
    driver=VGA;
    mode=VGAHI;
    initgraph(&driver, &mode, "");
    x1=x2=y1=y2=10;
    dx1=dy1=2;
    dx2=dy2=3;
    while(!kbhit())
    {
        line(x1, y1, x2, y2);
        x1+=dx1; y1+=dy1;
        x2+=dx2; y2+=dy2;
        if(x1<=LEFT || x1>=RIGHT)
            dx1=-dx1;
        if(y1<=TOP || y1>=BOTTOM)
            dy1=-dy1;
        if(x2<=LEFT || x2>=RIGHT)
            dx2=-dx2;
    }
}

```

```

        if(y2<=TOP||y2>=BOTTOM)
            dy2=-dy2;
        if(++count>LINES)
        {
            setcolor(color);
            color=(color>=MAXCOLOR)?0:++color;
        }
    }
    closegraph();
}

```

61. 杨辉三角形

题目：打印出杨辉三角形（要求打印出 10 行）。

程序分析：

结构如下所示：

```

1
1   1
1   2   1
1   3   3   1
1   4   6   4   1

```

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include <stdio.h>

```

```

int main()
{
    int i, j;
    int a[10][10];
    printf("\n");
    for(i=0; i<10; i++) {
        a[i][0]=1;
        a[i][i]=1;
    }
    for(i=2; i<10; i++)
        for(j=1; j<i; j++)

```

```

        a[i][j]=a[i-1][j-1]+a[i-1][j];
    for(i=0;i<10;i++) {
        for(j=0;j<=i;j++)
            printf("%5d",a[i][j]);
        printf("\n");
    }
}

```

输出结果:

```

1
1  1
1  2  1
1  3  3  1
1  4  6  4  1
1  5 10 10  5  1
1  6 15 20 15  6  1
1  7 21 35 35 21  7  1
1  8 28 56 70 56 28  8  1
1  9 36 84 126 126 84 36  9  1

```

62. 学习 putpixel 画点, (在 TC 中实现)

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include "stdio.h"
#include "graphics.h"
int main()
{
    int i, j, driver=VGA, mode=VGAHI;
    initgraph(&driver, &mode, "");
    setbkcolor(YELLOW);
    for(i=50; i<=230; i+=20)
        for(j=50; j<=230; j++)
            putpixel(i, j, 1);
    for(j=50; j<=230; j+=20)
        for(i=50; i<=230; i++)
            putpixel(i, j, 1);
}

```

63. 画椭圆 ellipse (在 TC 中实现)

```
// Created by www.runoob.com on 15/11/9.  
// Copyright © 2015 年 菜鸟教程. All rights reserved.  
//
```

```
#include "stdio.h"  
#include "graphics.h"  
#include "conio.h"  
int main()  
{  
    int x=360,y=160,driver=VGA,mode=VGAHI;  
    int num=20,i;  
    int top,bottom;  
    initgraph(&driver,&mode,"");  
    top=y-30;  
    bottom=y-30;  
    for(i=0;i<num;i++)  
    {  
        ellipse(250,250,0,360,top,bottom);  
        top-=5;  
        bottom+=5;  
    }  
    getch();  
}
```

64. 利用 ellipse and rectangle 画图 (在 TC 中实现)

```
// Created by www.runoob.com on 15/11/9.  
// Copyright © 2015 年 菜鸟教程. All rights reserved.  
//
```

```
#include "stdio.h"  
#include "graphics.h"  
#include "conio.h"  
main()  
{  
    int driver=VGA,mode=VGAHI;  
    int i,num=15,top=50;  
    int left=20,right=50;  
    initgraph(&driver,&mode,"");  
    for(i=0;i<num;i++)  
    {
```

```

        ellipse(250, 250, 0, 360, right, left);
        ellipse(250, 250, 0, 360, 20, top);
        rectangle(20-2*i, 20-2*i, 10*(i+2), 10*(i+2));
        right+=5;
        left+=5;
        top+=10;
    }
    getch();
}

```

65. 一个最优美的图案（在 TC 中实现）

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

#include "graphics.h"
#include "math.h"
#include "dos.h"
#include "conio.h"
#include "stdlib.h"
#include "stdio.h"
#include "stdarg.h"
#define MAXPTS 15
#define PI 3.1415926
struct PTS {
    int x,y;
};
double AspectRatio=0.85;
void LineToDemo(void)
{
    struct viewporttype vp;
    struct PTS points[MAXPTS];
    int i, j, h, w, xcenter, ycenter;
    int radius, angle, step;
    double rads;
    printf(" MoveTo / LineTo Demonstration" );
    getviewsettings( &vp );
    h = vp.bottom - vp.top;
    w = vp.right - vp.left;
    xcenter = w / 2; /* Determine the center of circle */
    ycenter = h / 2;
    radius = (h - 30) / (AspectRatio * 2);

```

```

    step = 360 / MAXPTS; /* Determine # of increments */
    angle = 0; /* Begin at zero degrees */
    for( i=0 ; i<MAXPTS ; ++i ){ /* Determine circle intercepts */
        rads = (double)angle * PI / 180.0; /* Convert angle to radians
    */
        points[i].x = xcenter + (int)( cos(rads) * radius );
        points[i].y = ycenter - (int)( sin(rads) * radius *
AspectRatio );
        angle += step; /* Move to next increment */
    }
    circle( xcenter, ycenter, radius ); /* Draw bounding circle */
    for( i=0 ; i<MAXPTS ; ++i ){ /* Draw the cords to the circle */
        for( j=i ; j<MAXPTS ; ++j ){ /* For each remaining intersect */
            moveto(points[i].x, points[i].y); /* Move to beginning of
cord */
            lineto(points[j].x, points[j].y); /* Draw the cord */
        }
    }
}

int main()
{
    int driver,mode;
    driver=CGA;mode=CGAC0;
    initgraph(&driver,&mode,"");
    setcolor(3);
    setbkcolor(GREEN);
    LineToDemo();
}

```

66. 输入 3 个数 a, b, c, 按大小顺序输出

程序分析：利用指针方法。

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

#include<stdio.h>

void swap(int *, int *);
int main(void)
{
    int a, b, c;
    int *p1, *p2, *p3;

```

```

printf("输入 a, b ,c:\n");
scanf("%d %d %d", &a, &b, &c);
p1 = &a;
p2 = &b;
p3 = &c;
if(a>b)
    swap(p1, p2);
if(a>c)
    swap(p1, p3);
if(b>c)
    swap(p2, p3);
printf("%d %d %d\n", a, b, c);
}
void swap(int *s1, int *s2)
{
    int t;
    t = *s1; *s1 = *s2; *s2 = t;
}

```

输出结果:

```

输入 a, b ,c:
1 3 2
1 2 3

```

67. 输入数组，最大的与第一个元素交换，最小的与最后一个元素交换，输出数组

```

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

void fun(int *s,int n)
{
    int i;
    int max=s[0];
    int a=0;
    for(i=0;i<n;i++)
    {
        if(s[i]>max)
        {
            max=s[i];
            a=i;
        }
    }
}

```

```

    }
    s[a]=s[0];
    s[0]=max;
    int j;
    int min=s[n-1];
    int b=n-1;
    for(j=0;j<n;j++)
    {
        if(s[j]<min)
        {
            min=s[j];
            b=j;
        }
    }
    s[b]=s[n-1];
    s[n-1]=min;
}

void printf_s(int *s, int n)
{
    int i;
    for(i=0;i<n;i++)
        printf("%d ", s[i]);
    printf("\n");
}

int main()
{
    int s[20];
    int i, n;
    printf("设置数组长度(<20):");
    scanf("%d", &n);
    printf("输入 %d 个元素:\n", n);
    for(i=0;i<n;i++)
        scanf("%d", &s[i]);
    fun(s, n);
    printf_s(s, n);
    return 0;
}

```

输出结果:

设置数组长度(<20):5
 输入 5 个元素:


```
12 123 4 65 21
123 12 21 65 4
```

68. 有 n 个整数，使其前面各数顺序向后移 m 个位置，最后 m 个数变成最前面的 m 个数

```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//
```

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int arr[20];
    int i, n, offset;
    //输入数组大小和数组内容
    printf("Total numbers?\n");
    scanf("%d", &n);
    printf("Input %d numbers.\n", n);
    for(i=0; i<n; i++)
        scanf("%d", &arr[i]);
    //输入滚动偏移量
    printf("Set your offset.\n");
    scanf("%d", &offset);
    printf("Offset is %d.\n", offset);
    //打印滚动前数组
    print_arr(arr, n);
    //滚动数组并打印
    move(arr, n, offset);
    print_arr(arr, n);
}
```

```
//打印数组
void print_arr(int array[], int n)
{
    int i;
    for(i=0; i<n; ++i)
        printf("%4d", array[i]);
    printf("\n");
}
```

```

}
//滚动数组
void move(int array[], int n, int offset)
{
    int *p, *arr_end;
    arr_end=array+n;      //数组最后一个元素的下一个位置
    int last;

    //滚动直到偏移量为 0
    while(offset)
    {
        last=*(arr_end-1);
        for(p=arr_end-1;p!=array;--p)    //向右滚动一位
            *p=*(p-1);
        *array=last;
        --offset;
    }
}

```

69. 有 n 个人围成一圈，顺序排号。从第一个人开始报数（从 1 到 3 报数），凡报到 3 的人退出圈子，问最后留下的是原来第几号的那位

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include <stdio.h>
void main()
{
    int num[50], n, *p, j, loop, i, m, k;
    printf("请输入这一圈人的数量:\n");
    scanf("%d", &n);
    p=num;
    //开始给这些人编号
    for (j=0; j<n; j++)
    {
        *(p+j)=j+1;
    }
}

```

```

i=0;//i 用于计数,即让指针后移
m=0;//m 记录退出圈子的人数
k=0;//k 报数 1, 2, 3
while(m<n-1)//当退出的人数不大于总人数时,即留下的人数至少是一个人
    //这句不能写成 m<n, 因为假设有 8 人, 当退出了 6 人时, 此时还是进
    行人数退出, 即 m++,
    //这时是 7<8, 剩下的一个人自己喊 1, 2, 3 那么他也就退出了, 将不
    会有输出
    {
        if (*(p+i)!=0)//如果这个人的头上编号不是 0 就开始报数加 1, 这里
        采用的方法是报数为 3 的人头上编号重置为 0
        {
            k++;
        }
        if (k==3)
        {
            k=0;    //报数清零, 即下一个人从 1 开始报数
            *(p+i)=0;//将报数为 3 的人编号重置为 0
            m++;    //退出人数加 1
        }
        i++;    //指针后移
        if (i==n)//这句很关键, 如果到了队尾, 就要使指针重新指向对头
            //并且它只能放在 i++后面, 因为只有 i++了才有可能 i==n
        {
            i=0;
        }
    }

}
printf("现在剩下的人是:");
for (loop=0;loop<n;loop++)
{
    if (num[loop]!=0)
    {
        printf("%2d 号\n", num[loop]);
    }
}
}

```

输出结果:

请输入这一圈人的数量:

8

现在剩下的人是: 7 号

70. 写一个函数，求一个字符串的长度，在 main 函数中输入字符串，并输出其长度

```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

#include <stdio.h>
#include <stdlib.h>
int main()
{
    int len;
    char str[20];
    printf("请输入字符串:\n");
    scanf("%s", str);
    len=length(str);
    printf("字符串有 %d 个字符。", len);
}
//求字符串长度
int length(char *s)
{
    int i=0;
    while(*s!='\0')
    {
        i++;
        s++;
    }
    return i;
}
```

输出结果：

```
请输入字符串：
www.runoob.com
字符串有 14 个字符。
```

71. 编写 input() 和 output() 函数输入，输出 5 个学生的数据记录

```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

#include<stdio.h>
#include<stdlib.h>
typedef struct{
    char name[20];
    char sex[5];
    int age;
}Stu;
void input(Stu*stu);
void output(Stu*stu);
int main()
{
    Stu stu[5];
    printf("请输入 5 个学生的信息：姓名 性别 年龄:\n");
    input(stu);
    printf("5 个学生的信息如下：\n 姓名 性别 年龄\n");
    output(stu);

    system("pause");
    return 0;
}
void input(Stu*stu)
{
    int i;
    for(i=0;i<5;i++)
        scanf("%s%s%d", stu[i].name, stu[i].sex, &(stu[i].age));
}
void output(Stu*stu)
{
    int i;
    for(i=0;i<5;i++)
        printf("%s %s %d\n", stu[i].name, stu[i].sex, stu[i].age);
}
```

输出结果：

```
请输入 5 个学生的信息：姓名 性别 年龄：
aaa m 15
bbb m 16
ccc m 15
ddd m 17
```

```
eee m 16
5 个学生的信息如下:
姓名  性别  年龄
aaa m 15
bbb m 16
ccc m 15
ddd m 17
eee m 16
```

72. 创建一个链表

```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//
```

```
#include<stdio.h>
#include<stdlib.h>
#include<malloc.h>
typedef struct LNode{
    int      data;
    struct LNode *next;
}LNode,*LinkList;

LinkList CreateList(int n);
void print(LinkList h);
int main()
{
    LinkList Head=NULL;
    int n;

    scanf("%d",&n);
    Head=CreateList(n);

    printf("刚刚建立的各个链表元素的值为: \n");
    print(Head);

    printf("\n\n");
    system("pause");
    return 0;
}
LinkList CreateList(int n)
{
    LinkList L,p,q;
```

```

    int i;
    L=(LNode*)malloc(sizeof(LNode));
    if(!L)return 0;
    L->next=NULL;
    q=L;
    for(i=1;i<=n;i++)
    {
        p=(LinkList)malloc(sizeof(LNode));
        printf("请输入第%d 个元素的值:", i);
        scanf("%d",&(p->data));
        p->next=NULL;
        q->next=p;
        q=p;
    }
    return L;
}

void print(LinkList h)
{
    LinkList p=h->next;
    while(p!=NULL) {
        printf("%d ",p->data);
        p=p->next;
    }
}

```

73. 反向输出一个链表

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include<stdio.h>
#include<stdlib.h>
#include<malloc.h>
typedef struct LNode{
    int      data;
    struct LNode *next;
}LNode,*LinkList;

```

```

LinkList CreateList(int n);
void print(LinkList h);
int main()
{

```

```

LinkedList Head=NULL;
int n;

scanf("%d",&n);
Head=CreateList(n);

printf("刚刚建立的各个链表元素的值为: \n");
print(Head);

printf("\n\n");
system("pause");
return 0;
}
LinkedList CreateList(int n)
{
    LinkedList L,p,q;
    int i;
    L=(LNode*)malloc(sizeof(LNode));
    if(!L)return 0;
    L->next=NULL;
    q=L;
    for(i=1;i<=n;i++)
    {
        p=(LinkedList)malloc(sizeof(LNode));
        printf("请输入第%d 个元素的值:", i);
        scanf("%d",&(p->data));
        p->next=NULL;
        q->next=p;
        q=p;
    }
    return L;
}
void print(LinkedList h)
{
    LinkedList p=h->next;
    while(p!=NULL) {
        printf("%d ",p->data);
        p=p->next;
    }
}

```

74. 连接两个链表


```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//
```

```
#include <stdlib.h>
#include <stdio.h>
struct list
{
    int data;
    struct list *next;
};
typedef struct list node;
typedef node *link;
link delete_node(link pointer, link tmp)
{
    if (tmp==NULL) /*delete first node*/
        return pointer->next;
    else
    {
        if(tmp->next->next==NULL)/*delete last node*/
            tmp->next=NULL;
        else /*delete the other node*/
            tmp->next=tmp->next->next;
        return pointer;
    }
}
void selection_sort(link pointer, int num)
{
    link tmp, btmp;
    int i, min;
    for(i=0; i<num; i++)
    {
        tmp=pointer;
        min=tmp->data;
        btmp=NULL;
        while(tmp->next)
        {
            if(min>tmp->next->data)
            {
                min=tmp->next->data;
                btmp=tmp;
            }
            tmp=tmp->next;
        }
    }
}
```

```

        printf("\40: %d\n",min);
        pointer=delete_node(pointer,btmp);
    }
}
link create_list(int array[],int num)
{
    link tmp1,tmp2,pointer;
    int i;
    pointer=(link)malloc(sizeof(node));
    pointer->data=array[0];
    tmp1=pointer;
    for(i=1;i<num;i++)
    {
        tmp2=(link)malloc(sizeof(node));
        tmp2->next=NULL;
        tmp2->data=array[i];
        tmp1->next=tmp2;
        tmp1=tmp1->next;
    }
    return pointer;
}
link concatenate(link pointer1,link pointer2)
{
    link tmp;
    tmp=pointer1;
    while(tmp->next)
        tmp=tmp->next;
    tmp->next=pointer2;
    return pointer1;
}
int main(void)
{
    int arr1[]={3,12,8,9,11};
    link ptr;
    ptr=create_list(arr1,5);
    selection_sort(ptr,5);
}

```

75. 输入一个整数，并将其反转后输出

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include <stdio.h>
int main()
{
    int n, reversedNumber = 0, remainder;

    printf("输入一个整数: ");
    scanf("%d", &n);

    while(n != 0)
    {
        remainder = n%10;
        reversedNumber = reversedNumber*10 + remainder;
        n /= 10;
    }

    printf("反转后的整数: %d", reversedNumber);

    return 0;
}

```

输出结果:

输入一个整数: 12345
 反转后的整数: 54321

76. 编写一个函数，输入 n 为偶数时，调用函数求 $1/2+1/4+\dots+1/n$ ，当输入 n 为奇数时，调用函数 $1/1+1/3+\dots+1/n$ (利用指针函数)

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include<stdio.h>
#include<stdlib.h>
double evennumber(int n);
double oddnumber(int n);

int main()

```

```

{
    int n;
    double r;
    double (*pfunc)(int);
    printf("请输入一个数字: ");
    scanf("%d",&n);
    if(n%2==0) pfunc=evennumber;
    else pfunc=odddnumber;

    r=(*pfunc)(n);
    printf("%lf\n",r);

    system("pause");
    return 0;
}
double evennumber(int n)
{
    double s=0,a=0;
    int i;
    for(i=2;i<=n;i+=2)
    {
        a=(double)1/i;
        s+=a;
    }
    return s;
}
double oddnumber(int n)
{
    double s=0,a=0;
    int i;
    for(i=1;i<=n;i+=2)
    {
        a=(double)1/i;
        s+=a;
    }
    return s;
}

```

输出结果:

请输入一个数字: 2
0.500000

77. 填空练习（指向指针的指针）

```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

#include<stdio.h>
#include<stdlib.h>
int main()
{
    const char *s[]={"man","woman","girl","boy","sister"};
    const char **q;
    int k;
    for(k=0;k<5;k++)
    {
        q=&s[k];          /*在这里填入内容*/
        printf("%s\n",*q);
    }
    return 0;
}
```

输出结果:

```
man
woman
girl
boy
sister
```

78. 找到年龄最大的人，并输出。请找出程序中有什么问题

```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

#include<stdio.h>
#include<stdlib.h>
struct man{
    char name[20];
    int age;
}
person[3]={"li",18,"wang",25,"sun",22};
int main()
{
```

```

    struct man *q,*p;
    int i,m=0;
    p=person;
    for(i=0;i<3;i++)
    {
        if(m<p->age)
        {
            m=p->age;
            q=p;
        }
        p++;
    }
    printf("%s %d\n",q->name,q->age);
    return 0;
}

```

输出结果:

wang 25

79. 字符串排序

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.

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

void swap(char*str1, char*str2);
int main()
{
    char str1[20], str2[20], str3[20];
    printf("请输入 3 个字符串, 每个字符串以回车结束!:\n");
    fgets(str1, (sizeof str1 / sizeof str1[0]), stdin);
    fgets(str2, (sizeof str2 / sizeof str2[0]), stdin);
    fgets(str3, (sizeof str3 / sizeof str3[0]), stdin);
    if(strcmp(str1, str2)>0) swap(str1, str2);
    if(strcmp(str2, str3)>0) swap(str2, str3);
    if(strcmp(str1, str2)>0) swap(str1, str2);
    printf("排序后的结果为: \n");
    printf("%s\n%s\n%s\n", str1, str2, str3);
    return 0;
}

```

```
void swap(char*str1, char*str2)
{
    char tem[20];
    strcpy(tem, str1);
    strcpy(str1, str2);
    strcpy(str2, tem);
}
```

输出结果:

请输入 3 个字符串, 每个字符串以回车结束!:

b

a

t

排序后的结果为:

a

b

t

80. 海滩上原来最少有多少个桃子?

题目: 海滩上有一堆桃子, 五只猴子来分。第一只猴子把这堆桃子平均分为五份, 多了一个, 这只 猴子把多的一个扔入海中, 拿走了一份。第二只猴子把剩下的桃子又平均分成五份, 又多了 一个, 它同样把多的一个扔入海中, 拿走了一份, 第三、第四、第五只猴子都是这样做的, 问海滩上原来最少有多少个桃子?

```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//
```

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int x
    , i=0, j=1;
    while(i<5) {
        x=4*j;
        for(i=0; i<5; i++)
        {
            if(x%4!=0) {break;}
            x=(x/4)*5+1;
        }
        j++;
    }
}
```

```

    }
    printf("%d\n", x);

    return 0;
}

```

输出结果:

3121

81. $809 * ?? = 800 * ?? + 9 * ??$ 其中??代表的两位数, $809 * ??$ 为四位数, $8 * ??$ 的结果为两位数, $9 * ??$ 的结果为3位数。求??代表的两位数, 及 $809 * ??$ 后的结果

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.

```

```

#include <stdio.h>

void output(long int b, long int i){
    printf("\n%ld = 800 * %ld + 9 * %ld\n", b, i, i);
}

int main(){

    void output(long int b, long int i);
    long int a, b, i;
    a = 809;
    for(i = 10; i < 100; i++){
        b = i * a;
        if (b >= 1000 && b <= 10000 && 8 * i < 100 && 9 * i >= 100){
            output(b, i);
        }
    }
    return 0;
}

```

输出结果:

9708 = 800 * 12 + 9 * 12

82. 八进制转换为十进制

```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

#include<stdio.h>
#include<stdlib.h>
int main()
{
    int n=0,i=0;
    char s[20];
    printf("请输入一个 8 进制数:\n");
    gets(s);
    while(s[i]!='\0') {
        n=n*8+s[i]-'0';
        i++;
    }
    printf("刚输入的 8 进制数转化为十进制为\n%d\n",n);

    return 0;
}
```

输出结果：

```
请输入一个 8 进制数:
16
刚输入的 8 进制数转化为十进制为
14
```

83. 求 0—7 所能组成的奇数个数

题目：求 0—7 所能组成的奇数个数。

程序分析：

这个问题其实是一个排列组合的问题，设这个数为 $sun=a_1a_2a_3a_4a_5a_6a_7a_8$, a_1-a_8 表示这个数的某位的数值，当一个数的最后一位为奇数时，那么这个数一定为奇数，不管前面几位是什么数字。如果最后一位数为偶数，则这个数一定为偶数。

a1-a8 可以取 0-7 这个八个数字，首位数字不为 0。

从该数为一位数到该数为 8 位数开始统计奇数的个数：

- 1. 当只有一位数时也就是该数的最后一位，奇数个数为 4
- 2. 当该数为两位数时，奇数个数为 $4*7=28$
- 3. 当该数为三位数时，奇数个数为： $4*8*7=224$
- ...
- 8. 当该数为八位数时，奇数个数为： $4*8*8*8*8*8*8*7$ (依次为最后一位到第一位)

```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

#include<stdio.h>
int main(int argc, char*argv[])
{
    long sum = 4, s = 4;//sum 的初始值为 4 表示，只有一位数字组成的奇数
    个数为 4 个
    int j;
    for (j = 2; j <= 8; j++)
    {
        printf("%d 位数为奇数的个数%d\n", j-1, s);
        if (j <= 2)
            s *= 7;
        else
            s *= 8;
        sum += s;
    }
    printf("%d 位数为奇数的个数%d\n", j-1, s);
    printf("奇数的总个数为: %d\n", sum);
    // system("pause");
    return 0;
}
```

输出结果：

```
1 位数为奇数的个数 4
2 位数为奇数的个数 28
3 位数为奇数的个数 224
4 位数为奇数的个数 1792
5 位数为奇数的个数 14336
6 位数为奇数的个数 114688
7 位数为奇数的个数 917504
```

8 位数为奇数的个数 7340032
奇数的总个数为：8388608

84. 一个偶数总能表示为两个素数之和

程序分析：我去，这是什么题目，要我证明这个问题吗？真不知道怎么证明。那就把一个偶数分解成两个素数吧。

```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

#include<stdio.h>
#include<stdlib.h>
int Isprimer(unsigned int n);
int main()
{
    unsigned int n,i;
    do{
        printf("请输入一个偶数:\n");
        scanf("%d",&n);
    }while(n%2!=0);
    for(i=1;i<n;i++)
        if(Isprimer(i)&&Isprimer(n-i))
            break;
    printf("偶数%d 可以分解成%d 和%d 两个素数的和\n",n,i,n-i);

    return 0;
}
int Isprimer(unsigned int n)
{
    int i;
    if(n<4)return 1;
    else if(n%2==0)return 0;
    else
        for(i=3;i<sqrt(n)+1;i++)
            if(n%i==0)return 0;

    return 1;
}
```

输出结果：

请输入一个偶数：

4

偶数 4 可以分解成 1 和 3 两个素数的和

85. 判断一个素数能被几个 9 整除

程序分析：丫的！这题目的意思是判断一个素数能整除几个 9 组成的数吧？我就这么理解吧。素数是不能被除 1 和自身之外的数整除的

```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

#include<stdio.h>
#include<stdlib.h>
int main()
{
    int p,i;
    long int sum=9;
    printf("请输入一个素数:\n");
    scanf("%d",&p);
    for(i=1;;i++)
        if(sum%p==0)break;
        else sum=sum*10+9;

    printf("素数%d 能整除%d 个 9 组成的数%d\n",p,i,sum);
    return 0;
}
```

输出结果：

请输入一个素数：

13

素数 13 能整除 6 个 9 组成的数 999999

86. 两个字符串连接程序

```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

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

```

char* strconnect(char *str1, char *str2);

int main()
{
    char str1[20], str2[20];
    char *str;
    puts("请输入两个字符串，用回车分开:");
    scanf("%s%s", str1, str2);
    str=strconnect(str1, str2);
    puts("连接后的字符串为:");
    puts(str);
    return 0;
}

char* strconnect(char *str1, char *str2)
{
    char*str;
    str=(char*)malloc(strlen(str1)+strlen(str2)+1);
    str[0]='\0';
    strcat(str, str1);
    strcat(str, str2);
    return str;
}

```

87. 回答结果（结构体变量传递）

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include<stdio.h>

```

```

struct student
{
    int x;
    char c;
} a;

```

```

int main()
{
    a.x=3;
    a.c='a';
    f(a);
}

```

```

        printf("%d,%c", a.x, a.c);
    }
    f(struct student b)
    {
        b.x=20;
        b.c='y';
    }

```

输出结果:

3, a

88. 读取 7 个数（1—50）的整数值，每读取一个值，程序打印出该值个数的 *

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

#include<stdio.h>
#include<stdlib.h>
int main()
{
    int n, i, j;
    printf("请输入数字:\n"); i--;
    for(i=0; i<7; i++)
    {
        scanf("%d", &n);
        if(n>50) {
            printf("请重新输入:\n"); i--;
        }
        else
        {
            for(j=0; j<n; j++)
                printf("*");
        }
        printf("\n");
    }
    return 0;
}

```

输出结果:

请输入数字:

5

89. 电话号码加密

题目：某个公司采用公用电话传递数据，数据是四位的整数，在传递过程中是加密的，加密规则如下： 每位数字都加上 5, 然后用和除以 10 的余数代替该数字，再将第一位和第四位交换，第二位和第三位交换。

```
// Created by www.runoob.com on 15/11/9.  
// Copyright © 2015 年 菜鸟教程. All rights reserved.  
//
```

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

```
int main()  
{  
    int a, i, aa[4], t;  
    printf("请输入四位数字: ");  
    scanf("%d", &a);  
    aa[0]=a%10;  
    aa[1]=a%100/10;  
    aa[2]=a%1000/100;  
    aa[3]=a/1000;  
    for(i=0; i<=3; i++)  
    {  
        aa[i]+=5;  
        aa[i]%=10;  
    }  
    for(i=0; i<=3/2; i++)  
    {  
        t=aa[i];  
        aa[i]=aa[3-i];  
        aa[3-i]=t;  
    }  
    printf("加密后的数字: ");  
    for(i=3; i>=0; i--)  
        printf("%d", aa[i]);  
    printf("\n");  
}
```

输出结果：

请输入四位数字：1234
加密后的数字：9876

90. 专升本一题，读结果

```
// Created by www.runoob.com on 15/11/9.  
// Copyright © 2015 年 菜鸟教程. All rights reserved.  
//
```

```
#include<stdio.h>  
#include<stdlib.h>  
  
#define M 5  
int main()  
{  
    int a[M]={1,2,3,4,5};  
    int i,j,t;  
    i=0;j=M-1;  
    while(i<j)  
    {  
        t=*(a+i);  
        *(a+i)=*(a+j);  
        *(a+j)=t;  
        i++;j--;  
    }  
    for(i=0;i<M;i++) {  
        printf("%d\n",*(a+i));  
    }  
}
```

输出结果：

```
5  
4  
3  
2  
1
```

91. 时间函数举例 1

```
// Created by www.runoob.com on 15/11/9.  
// Copyright © 2015 年 菜鸟教程. All rights reserved.
```



```
//

#include <stdio.h>
#include <time.h>

int main ()
{
    time_t rawtime;
    struct tm * timeinfo;

    time ( &rawtime );
    timeinfo = localtime ( &rawtime );
    printf ( "当前本地时间为: %s", asctime (timeinfo) );

    return 0;
}
```

输出结果:

当前本地时间为: Tue Nov 10 16:28:49 2015

92. 时间函数举例 2

```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

#include <stdio.h>
#include <time.h>

int main()
{
    time_t start,end;
    int i;
    start=time(NULL);
    for(i=0;i<300000;i++)
    {
        printf("\n"); // 返回两个 time_t 型变量之间的时间间隔
    }
    end=time(NULL);

    // 输出执行时间
    printf("时间间隔为 %6.3f\n",difftime(end,start));
}
```

输出结果:

时间间隔为 1.000

93. 时间函数举例 3

```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

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

int main()
{
    long i=100000000L;
    clock_t start, finish;
    double TheTimes;
    printf("做%d 次空循环需要的时间为", i);
    start=clock();
    while(i--);
    finish=clock();
    TheTimes=(double) (finish-start)/CLOCKS_PER_SEC;
    printf("%f 秒。 \n", TheTimes);
    return 0;
}
```

输出结果:

做 10000000 次空循环需要的时间为 0.025367 秒。

94. 猜谜游戏

```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

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

void caizi(void)
```

```

{
    int n;
    char begin;
    int count = 1;
    srand((int)time(NULL));
    int m = (rand() % 100) + 1;
    puts("游戏开始, 请输入数字:");
    while (1)
    {
        scanf("%d", &n);
        if (n == m)
        {
            printf("猜中了, 使用了 %d 次! \n", count);
            if (count == 1)
            {
                printf("你是神级人物了! 膜拜\n");
                getchar();
                printf("你已经达到最高级别, 还需要玩吗? Y/N \n");
                scanf("%c", &begin);
                if (begin == 'Y' || begin == 'y')           //重复玩的一个
                    嵌套循环
                {
                    caizi();
                }
                else
                {
                    printf("谢谢, 再见!\n");
                }
            }
            else if (count <= 5)
            {
                printf("你是王级人物了! 非常赞\n");
                getchar();
                printf("需要挑战最高级别不? Y/N \n");
                scanf("%c", &begin);
                if (begin == 'Y' || begin == 'y')
                {
                    caizi();
                }
                else
                {
                    printf("谢谢, 再见!\n");
                }
            }
        }
    }
}

```

```

else if (count <= 10)
{
    printf("你是大师级人物了！狂赞\n");
    getchar();
    printf("需要挑战最高级别不？ Y/N \n");
    scanf("%c", &begin);
    if (begin == 'Y' || begin == 'y')
    {
        caizi();
    }
    else
    {
        printf("谢谢，再见!\n");
    }
}
else if (count <= 15)
{
    printf("你是钻石级人物了！怒赞\n");
    getchar();
    printf("需要挑战最高级别不？ Y/N \n");
    scanf("%c", &begin);
    if (begin == 'Y' || begin == 'y')
    {
        caizi();
    }
    else
    {
        printf("谢谢，再见!\n");
    }
}
else
{
    getchar();
    printf("你的技术还有待提高哦！重玩？ Y/N\n");
    scanf("%c",&begin);
    if (begin == 'Y' || begin == 'y')
    {
        caizi();
    }
    else
    {
        printf("谢谢，再见!\n");
    }
}
}

```

```

        break;
    }
    else if (n < m)
    {
        puts("太小了!");
        puts("重新输入:");
    }
    else
    {
        puts("太大了!");
        puts("重新输入:");
    }
    count++; //计数器
}
}

int main(void)
{
    caizi();
    system("pause");
    return 0;
}

```

输出结果:

游戏开始，请输入数字：

50

太大了！

重新输入：

25

太小了！

重新输入：

40

太大了！

重新输入：

30

太大了！

重新输入：

27

太小了！

重新输入:
28
猜中了, 使用了 6 次!
你是大师级人物了! 狂赞
需要挑战最高级别不? Y/N
N
谢谢, 再见!

95. 简单的结构体应用实例

```
// Created by www.runoob.com on 15/11/9.  
// Copyright © 2015 年 菜鸟教程. All rights reserved.  
//  
  
#include <stdio.h>  
  
struct programming  
{  
    float constant;  
    char *pointer;  
};  
  
int main()  
{  
    struct programming variable;  
    char string[] = "菜鸟教程: http://www.runoob.com";  
  
    variable.constant = 1.23;  
    variable.pointer = string;  
  
    printf("%f\n", variable.constant);  
    printf("%s\n", variable.pointer);  
  
    return 0;  
}
```

输出结果:

1.230000
菜鸟教程: http://www.runoob.com

96. 计算字符串中子串出现的次数

```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int main()
{
    int i, j, k, TLen, PLen, count=0;
    char T[50], P[10];
    printf("请输入两个字符串，以回车隔开，母串在前，子串在后：\n");
    gets(T);
    gets(P);
    TLen=strlen(T);
    PLen=strlen(P);
    for(i=0; i<=TLen-PLen; i++)
    {
        for(j=0, k=i; j<PLen&&P[j]==T[k]; j++, k++)
            ;
        if(j==PLen) count++;
    }
    printf("%d\n", count);
    system("pause");
    return 0;
}
```

输出结果：

请输入两个字符串，以回车隔开，母串在前，子串在后：

abca

a

2

97. 从键盘输入一些字符，逐个把它们送到磁盘上去，直到输入一个#为止

```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//
```

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

```

int main()
{
    FILE*fp=NULL;
    char filename[25];
    char ch;
    printf("输入你要保存到的文件的名称: \n");
    gets(filename);
    if((fp=fopen(filename, "w"))==NULL)
    {
        printf("error: cannot open file!\n");
        exit(0);
    }
    printf("现在你可以输入你要保存的一些字符, 以#结束: \n");
    getchar();
    while((ch=getchar())!='#') {
        fputc(ch, fp);
    }
    fclose(fp);
    system("pause");
    return 0;
}

```

输出结果:

```

输入你要保存到的文件的名称:
test.txt
现在你可以输入你要保存的一些字符, 以#结束:
www.runoob.com
#

```

98. 从键盘输入一个字符串, 将小写字母全部转换成大写字母, 然后输出到一个磁盘文件"test"中保存。输入的字符串以! 结束

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

```

```

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

```



```

#include<string.h>

int main()
{
    FILE*fp=NULL;
    char str[50];
    int i,len;
    printf("输入一个字符串: \n");
    gets(str);
    len=strlen(str);
    for(i=0;i<len;i++)
    {
        if(str[i]<='z' && str[i]>='a')
            str[i]-=32;
    }
    if((fp=fopen("test", "w"))==NULL)
    {
        printf("error: cannot open file!\n");
        exit(0);
    }
    fprintf(fp, "%s", str);
    fclose(fp);

    system("pause");
    return 0;
}

```

输出结果：

输入一个字符串：
www.runoob.com

99. 有两个磁盘文件 A 和 B, 各存放一行字母，要求把这两个文件中的信息合并（按字母顺序排列），输出到一个新文件 C 中

程序分析：你需要先创建 A.txt 与 B.txt。

A.txt 文件内容：

B. txt 文件内容:

456

```
// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int main()
{
    FILE*fa,*fb,*fc;
    int i,j,k;
    char str[100],str1[100];
    char tem;
    if((fa=fopen("A.txt","r"))==NULL) // A.txt 文件需要存在
    {
        printf("error: cannot open A file!\n");
        exit(0);
    }
    fgets(str,99,fa);
    fclose(fa);
    if((fb=fopen("B.txt","r"))==NULL) // B.txt 文件需要存在
    {
        printf("error: cannot open B file!\n");
        exit(0);
    }
    fgets(str1,100,fb);
    fclose(fb);
    strcat(str,str1);
    for(i=strlen(str)-1;i>1;i--)
        for(j=0;j<i;j++)
            if(str[j]>str[j+1])
            {
                tem=str[j];
                str[j]=str[j+1];
                str[j+1]=tem;
            }
}
```

```

    }

    if((fc=fopen("C.txt","w"))==NULL) // 合并为 C.txt
    {
        printf("error: cannot open C file!\n");
        exit(0);
    }
    fputs(str,fc);
    fclose(fc);
    system("pause");
    return 0;
}

```

输出结果：

123456

100. 计算平均成绩

题目：有五个学生，每个学生有 3 门课的成绩，从键盘输入以上数据（包括学生号，姓名，三门课成绩），计算出平均成绩，将原有的数据和计算出的平均分数存放在磁盘文件“stud”中。

```

// Created by www.runoob.com on 15/11/9.
// Copyright © 2015 年 菜鸟教程. All rights reserved.
//

#include<stdio.h>
#include<stdlib.h>
typedef struct{
    int ID;
    int math;
    int English;
    int C;
    int avargrade;
    char name[20];
}Stu;
int main()
{
    FILE*fp;
    Stu stu[5];
    int i, avargrade=0;
    printf("请输入 5 个同学的信息：学生号，姓名，3 门成绩:\n");
    for(i=0;i<5;i++)

```

```

    {

scanf("%d %s %d %d %d",&(stu[i].ID),stu[i].name,&(stu[i].math),&(stu[i].
English),&(stu[i].C));
    stu[i].avargrade=(stu[i].math+stu[i].English+stu[i].C)/3;
    }

    if((fp=fopen("stud","w"))==NULL)
    {
        printf("error :cannot open file!\n");
        exit(0);
    }
    for(i=0;i<5;i++)

fprintf(fp,"%d %s %d %d %d %d\n",stu[i].ID,stu[i].name,stu[i].math,stu[
i].English,
        stu[i].C,stu[i].avargrade);

    fclose(fp);
    // system("pause");
    return 0;
}

```

输出结果:

请输入 5 个同学的信息： 学生号， 姓名， 3 门成绩：

```

1 a 60 70 80
2 b 60 80 90
3 c 59 39 89
4 e 56 88 98
5 d 43 88 78

```

打开 stud 文件， 内容如下

```

1 a 60 70 80 70
2 b 60 80 90 76
3 c 59 39 89 62
4 e 56 88 98 80
5 d 43 88 78 69

```

101. 矩阵交换行

描述:

给定一个 **5*5** 的矩阵（数学上，一个 $r \times c$ 的矩阵是一个由 r 行 c 列元素排列成的矩形阵列），将第 n 行和第 m 行交换，输出交换后的结果。

输入输入共 6 行，前 5 行为矩阵的每一行元素，元素与元素之间以一个空格分开。

第 6 行包含两个整数 m 、 n ，以一个空格分开。（ $1 \leq m, n \leq 5$ ）输出输出交换之后的矩阵，矩阵的每一行元素占一行，元素之间以一个空格分开。样例输入：

```
1 2 2 1 2
5 6 7 8 3
9 3 0 5 3
7 2 1 4 6
3 0 8 2 4
1 5
```

样例输出：

```
3 0 8 2 4
5 6 7 8 3
9 3 0 5 3
7 2 1 4 6
1 2 2 1 2
```

```
#include <iostream>
using namespace std;
int main()
{
    int a[5][5], m, n, i, j;
    for (i=0; i<5; i++)
        for (j=0; j<5; j++)
            cin >> a[i][j];
    cin >> m >> n;
    for (j=0; j<5; j++){
        i = a[m-1][j];
        a[m-1][j] = a[n-1][j];
        a[n-1][j] = i;
    }
    for (i=0; i<5; i++){
        for (j=0; j<5; j++){
            cout << a[i][j] << " ";
        }
        cout << endl;
    }
}
```

```
    }  
    return 0;  
}
```

102. 猴子吃桃问题

一只小猴子一天摘了许多桃子，第一天吃了一半，然后忍不住又吃了一个；第二天又吃了一半，再加上一个；后面每天都是这样吃。到第 10 天的时候，小猴子发现只有一个桃子了。问小猴子第一天共摘了多少个桃子。

```
#include<stdio.h>  
  
int main()  
{  
    int i = 1;  
    int j = 1;  
    for (j = 10; j > 1;j --)  
    {  
        i++;  
        i = 2 * i;  
    }  
    printf("第一天摘了 %d 个桃子。", i);  
}
```

输出结果：

第一天摘了 1534 个桃子。