前言

- 以下题目大部分来自于C语言经典100例

1、题目：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？

```
#include<stdio.h>

int main() {
    int g,s,b;
    int count = 0; // 计算的个数
    for (g = 1; g <= 4; g++) {
        for (s = 1; s <= 4; s++) {
            for (b = 1; b <= 4; b++) {
                if (g != s && g!= b && s && b){
                    printf("%d,%d,%d\n",g,s,b);
                    count++;
                }
            }
        }
    }
    printf("count : %d\n",count);
    return 0;
}
```

2、题目：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?

```
#include<stdio.h>

int main()
{

    double profit;
    double bonus1,bonus2,bonus4,bonus6,bonus10,bonus;
    printf("你的净利润是：");

    scanf("%lf",&profit);
```

```
    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(profit<=100000) {
        bonus=profit*0.1;
    } else if(profit<=200000) {
        bonus=bonus1+(profit-100000)*0.075;
    } else if(profit<=400000) {
        bonus=bonus2+(profit-200000)*0.05;
    } else if(profit<=600000) {
        bonus=bonus4+(profit-400000)*0.03;
    } else if(profit<=1000000) {
        bonus=bonus6+(profit-600000)*0.015;
    } else if(profit>1000000) {
        bonus=bonus10+(profit-1000000)*0.01;
    }
    printf("提成为：bonus=%lf",bonus);

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

3、题目：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?

```
#include<stdio.h>

int main ()
{
    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;
}
```

4、题目：输入某年某月某日，判断这一天是这一年的第几天?

```
#include<stdio.h>

int main()
{
    int year,month,day; int is_year(int);
    int m[13] = {0,31,28,31,30,31,30,31,31,30,31,30,31};
    printf("input year month day: ");
    fflush(stdout);
    scanf("%d %d %d",&year,&month,&day);
    int i;
    for(i = 1;i < month;i++)
        day += m[i];
    if(is_year(year) && month >= 2)
        day++;
    printf("这一天是这一年的第%d天\n",day);

    return 0;
}

int is_year(int y){
    return (((y%4==0)&&(y%100!=0))||(y%400==0));
}
```

5、题目：输入三个整数x,y,z，把这三个数由小到大输出。

```
#include<stdio.h>

int main()
{
    int x,y,z,t;
    scanf("%d %d %d",&x,&y,&z);

    if(x > y){t = x;x = y;y = t;}
    if(x > z){t = x;x = z;z = t;}
    if(y > z){t = y;y = z;z = t;}
    printf("%d %d %d\n",x,y,z);

    return 0;
}
```

6、题目：用*号输出字母C的图案。

```
#include<stdio.h>

int main()
{
```

```
    int i,j,n;
    scanf("%d",&n);
    for(i = 0;i < n;i++){
        printf("* ");
    }
    printf("\n");
    for(i = 1;i < n-1;i++){
        printf("*\n");
    }
    for(i = 0;i < n;i++){
        printf("* ");
    }

    return 0;
}
```

7、题目：输出9*9乘法表。

```
#include<stdio.h>

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

8、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）

（斐波那契数列）

```
#include<stdio.h>

int main()
{
    int cnt;
    int cnt1 = 1, cnt2 = 1;
    printf("第1个月：%d\n", cnt1);
    printf("第2个月：%d\n", cnt2);
```

```
    for (int i = 3; i < 41; ++i)
    {
        cnt = cnt1 + cnt2;
        printf("第%d个月：%d\n", i, cnt);
        cnt1 = cnt2;
        cnt2 = cnt;
    }
    return 0;
}
```

9、题目：判断101到200之间的素数。

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。

```
#include <stdio.h>
#include <math.h>
int main()
{
    int i, j;
    for (i = 101; i <= 200; i++) {
        for (j = 2; j <= sqrt(i); j++) {
            if (i % j == 0)
                break;
        }
        if (j > sqrt(i))
            printf("%d ", i);
    }
    return 0;
}
```

10、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。

```
#include<stdio.h>

int main()
{
    int i, g,s,b;
    for (i = 100; i <= 999; i++) {
        g = i % 10;
        s = i / 10 % 10;
        b = i / 100 % 10;
        if (i == g * g * g + s * s * s + b * b * b)
            printf("%d ", i);
    }
```

```
    return 0;
}
```

11、题目：将一个正整数分解质因数。例如：输入90，打印出90=233*5。

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)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。
(2)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。

```
#include<stdio.h>

int main()
{
    int i;
    int n = 90;
    printf("%d=", n);
    for (i = 2; i <= n; i++) {
        while (n % i == 0) {
            printf("%d", i);
            n = n / i;
            if (n != 1)
                printf("*");
        }
    }
    return 0;
}
```

12、题目：利用条件运算符的嵌套：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。

```
#include<stdio.h>

int main()
{
    int score;
    char rank;
    scanf("%d", &score);
    rank = (score >= 90) ? 'A' : ((score >= 60) ? 'B' : 'C');
    printf("%c", rank);
    return 0;
}
```

```c
#include<stdio.h>

#define GRAND(x) (x >= 90) ? 'A' : (x > 60 ? 'B' : 'C')

int main()
{
    int score;
    while (~scanf("%d", &score)) {
        printf("学习成绩为：%c\n", GRAND(score));
    }
    return 0;
}
```

13、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。

最小公倍数 * 最大公约数 = m * n

```c
#include<stdio.h>

int main()
{
    int m, n, r;
    scanf("%d %d", &m, &n);
    int tmp = m * n;
    r = m % n;
    while (r != 0) {
        m = n;
        n = r;
        r = m % n;
    }
    printf("最大公约数为%d,最小公倍数为%d", n, tmp / n);

    return 0;
}
```

14、题目：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。

```c
#include<stdio.h>

int main()
{

    int letter = 0;
    int space = 0;
    int figure = 0;
    int other = 0;
    char ch;
    while ((ch = getchar()) != '\n') {
        if (ch >= 'a' && ch <= 'z' || ch >= 'A' && ch <= 'Z')
```

```
            letter++;
        else if (ch == ' ')
            space++;
        else if (ch >= '1' && ch <= '9')
            figure++;
        else
            other++;
    }
    printf("字母 = %d, 数字 = %d, 空格 = %d, 其他 = %d\n", letter, figure, space,
other);
    return 0;
}
```

15、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)，几个数相加由键盘控制。

```
#include<stdio.h>

int main()
{
    int a, n;
    scanf("%d %d", &a, &n);
    int s = 0;
    int tmp = a;
    for (int i = 0; i < n; i++)
    {
        s += tmp;
        tmp = tmp * 10 + a;
    }
    printf("a + aa + ... = %d", s);

    return 0;
}
```

16、题目：一个数如果恰好等于它的真因子之和，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。

Tips:找数n的因子只需要找到n/2(n/2-n的数是n的1-2倍，所以当大于n/2时，不会有因数)

```
#include<stdio.h>

int main()
{
    int a[1000];
    for (int i = 1; i < 1000; ++i)
    {
```

```
        int sum = 0, k = 0;
        for (int j = 1; j <= i / 2; ++j)
        {
            if (i%j == 0)
            {
                sum += j;
                a[k++] = j;
            }
        }
        if (i == sum)
        {
            printf("%d = %d", i, a[0]);
            for (int m = 1; m < k; ++m)
            {
                printf(" + %d", a[m]);
            }
            printf("\n");
        }
    }

    return 0;
}
```

17、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？

```
#include<stdio.h>

int main()
{
    float height = 100;
    float sum = height;
    for (int i = 0; i < 9; ++i)
    {
        height /= 2;
        sum += 2 * height;
    }
    height /= 2;
    printf("共经过%f米\n", sum);
    printf("第10次反弹%f米\n", height);

    return 0;
}
```

```c
int main() {
    float m = 100.0; // 米
    int count = 10; //共有10次
    float sum = m; count--; // 经过的米数，所以次数要减一次
    float f = 0;   // 第10次反弹的高度
    while (count--) { // 已经经过了一次，所以就是9次
        f = m / 2;  // 每次下落一次除2
        sum += m;   // 米数增加
        m /= 2;
    }
    printf("第10次落地时，共经过%f米，第10次反弹高%f米\n", sum, f);
    return 0;
}
```

18、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，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，只剩下一个桃子了。求第一天共摘了多少。

```c
#include<stdio.h>

int main() {
    int day;
    int peach = 1; // 第十天的桃
    for (day = 9; day >= 1; day--) {
        peach = (peach + 1) * 2;
    }
    printf("第一天共摘：%d", peach);
    return 0;
}
```

19、题目：打印出如下图案（菱形）。

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

```c
#include<stdio.h>

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

```
    int n = 4;
    scanf("%d", &n);
    // 上半部分
    for (i = 0; i < n; i++) {
        for (j = 0; j < n - i - 1; j++)
            printf(" ");
        for (j = 0; j < 2 * i + 1; j++)
            printf("*");
        printf("\n");
    }
    // 下半部分
    for (i = n - 1; i > 0; i--) {
        for (j = n - 1; j >= i; j--)
            printf(" ");
        for (j = 2 * (i - 1); j >= 0; j--)
            printf("*");
        printf("\n");
    }
    return 0;
}
```

20、题目：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。

```
#include<stdio.h>

int main() {
    float s = 0;
    int i = 0,t;
    int fenmu = 1,fenzi = 2;

    for (i = 0;i < 20;i++) {
        t = fenzi; // 提前保存好上一项的分子
        s += (float)fenzi / fenmu; // 计算和
        fenzi = fenzi + fenmu; // 下一项的分子=上一项的分子和分母之和
        fenmu = t;// 分母=上一项的分子
    }

    printf("%f", s);// 32.660263
    return 0;
}
```

21、题目：求1+2!+3!+...+20!的和。

```
#include<stdio.h>

int main() {
    int i, j;
    unsigned long long sum = 0, ret = 1;
```

```
    for (i = 1; i <= 20; i++) {
        ret = 1;
        for (j = 1; j <= i; j++) {
            ret *= j;
        }
        sum += ret;
    }
    printf("%llu", sum);
    return 0;
}
```

22、题目：利用递归方法求5!。

```
#include<stdio.h>

int fun(int n) {
    if (n == 1)
        return 1;
    else
        return n * fun(n - 1);
}
int main() {
    int n = 5;
    printf("5的阶乘为：%d\n",fun(n));
    return 0;
}
```

23、题目：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。

```
#include<stdio.h>

void fun(int n) {
    if (n == 1)
    {
        char tmp = getchar();
        putchar(tmp);
    }
    else
    {
        char tmp = getchar();
        fun(n - 1);
        putchar(tmp);
    }
}

int main() {
    fun(5);
    return 0;
}
```

24、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，他说是10岁。请问第五个人多大?

```c
#include<stdio.h>

int age(int n) {
    if (n == 1)
        return 10;
    else
        return age(n - 1) + 2;
}

int main() {
    printf("第5个人是：%d\n", age(5));
    return 0;
}
```

25、题目：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。

```c
#include<stdio.h>

void fun(int n, int* ret) {
    if (n == 0)
        return;
    else {
        ++(*ret);
        printf("%d ", n % 10);
        fun(n / 10, ret);
    }
}

int main() {
    int n = 0;
    scanf("%d", &n);
    int ret = 0;
    fun(n, &ret);
    printf("\n该数是%d位\n", ret);
    return 0;
}
```

26、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。

```
#include<stdio.h>

int main() {
    int n;
    while (~scanf("%d", &n)) {
        if ((n / 10000 == n % 10) && (n / 1000 % 10 == n / 10 % 10))
            printf("该五位数是回文数\n");
        else
            printf("该五位数不是回文数\n");
    }
    return 0;
}
```

27、题目：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。

```
#include<stdio.h>

int main()
{
    char c1, c2;
    scanf("%c", &c1);
    switch (c1){
    case 'M':
        printf("Monday");
        break;
    case 'T':
        getchar();
        scanf("%c", &c2);
        if (c2 == 'u')
            printf("Tuesday");
        else
            printf("Thursday");
        break;
    case 'W':
        printf("Wednesday");
        break;
    case 'F':
        printf("Friday");
        break;
    case 'S':
        getchar();
        scanf("%c", &c2);
        if (c2 == 'a')
```

```
            printf("Saturday");
        else
            printf("Sunday");
        break;
    }
    return 0;
}
```

28、题目：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。

例如：字符串"aaaaaaaaaaaaaaaaaobfowbfwfwn"，删除其中的所有a字母

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

int main() {
    char str[256] = { 0 };
    scanf("%s", str);
    int i, j = 0, len = strlen(str);
    char c;
    getchar();

    scanf("%c", &c);

    for (i = 0; i < len; i++) {
        if (str[i] == c) {
            for (j = i; j < len; j++) {
                str[j] = str[j + 1];
            }
            i = -1; // 注意
        }
    }
    printf("%s", str);
    return 0;
}
```

29、题目：判断一个数字是否为质数（素数）。

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

int main()
{
    int n;
    scanf("%d", &n);
    int flag = 1;
    if (n <= 0)
        flag = 0;
```

```
    for (int i = 2; i < sqrt(n); ++i){
        if (n % i == 0){
            flag = 0;
            break;
        }
    }
    if (flag)
        printf("是素数");
    else
        printf("不是素数");
    return 0;
}
```

30、题目：字符串反转，如将字符串 "I am a student" 反转为 "tneduts a ma I"。

```
#include <stdio.h>

int main() {
    void reverse(char*);
    char s[30] = "I am a student";
    reverse(s);
    puts(s);
    return 0;
}


void reverse(char* s) {
    int i = 0;
    int len = strlen(s);
    int end = len - 1;
    char tmp;
    while (i < len / 2) {
        tmp = s[i];
        s[i] = s[end];
        s[end] = tmp;
        end--;
        i++;
    }
}
```

31、题目：求100~200中的素数。

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

int main() {
    for (int i = 100; i <= 200; i++) {
        int flag = 1;
        for (int j = 2; j <= sqrt(i); j++) {
            if (i % j == 0) {
                flag = 0;
```

```
                break;
            }
        }
        if (flag)
            printf("%d ", i);
    }
    return 0;
}
```

32、题目：对10个数进行排序。(冒泡排序)

```
#include <stdio.h>

int main() {
    int arr[10] = { 23, 2 ,27, 98, 234, 1 ,4, 90, 88, 34 };
    int i, j, tmp, flag = 0, n = sizeof(arr) / sizeof(arr[0]);
    for (i = 0; i < n - 1; i++) {
        flag = 1; // 假设数组本来就有序
        for (j = 0; j < n - 1 - i; j++) {
            if (arr[j] < arr[j + 1]) {
                tmp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = tmp;
                flag = 0;
            }
        }
        if (flag) break;
    }
    for (i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }

    return 0;
}
```

33、题目：求一个3*3矩阵对角线元素之和。

```
#include <stdio.h>

int main()
{
    int a[3][3];
    for (int i = 0; i < 3; ++i)
    {
        for (int j = 0; j < 3; ++j)
        {
            scanf("%d", &a[i][j]);
        }
    }

    int sum = 0;
    for (int i = 0; i < 3; ++i)
    {
```

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

    return 0;
}
```

34、题目：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。（二分查找）

```
#include<stdio.h>

int main() {

    int arr[11] = { 1,2,3,4,5,6,7,8,9,10 };
    int i, j, len, n;
    len = sizeof(arr) / sizeof(arr[0]) - 1;
    scanf("%d", &n);
    int left = 0, right = len, mid;
    for (i = 0;i < len;i++) {
        mid = left + (right - left) / 2;
        if (arr[mid] < n)
            left = mid + 1;
        else if (arr[mid] > n)
            right = mid - 1;
        else {
            // 找到了
            for (j = len;j >= mid;j--) {
                arr[j] = arr[j - 1];
            }
            break;
        }
    }
    for (i = 0;i < sizeof(arr) / sizeof(arr[0]);i++)
        printf("%d ", arr[i]);
    return 0;
}
```

35、题目：将一个数组逆序输出。

```
#include<stdio.h>

int main() {
    int arr[] = { 1,2,3,4,5,6,7,8,9 ,10 };
    int len = sizeof(arr) / sizeof(arr[0]);
    int i, j = len - 1, tmp;
    for (i = 0;i < len / 2;i++, j--) {
        tmp = arr[i];
        arr[i] = arr[j];
        arr[j] = tmp;
```

```
    }
    for (i = 0;i < len;i++) {
        printf("%d ", arr[i]);
    }
    return 0;
}
```

36、题目：输入3个数a,b,c，按大小顺序输出。

```
#include <stdio.h>

// 法一：使用按位或来进行交换，普通的使用第三个变量交换略过
int main() {
    int a, b, c;
    scanf("%d %d %d", &a, &b, &c);
    if (a < b) {
        b = a ^ b;
        a = a ^ b;
        b = a ^ b;
    }

    if (a < c) {
        c = a ^ c;
        a = a ^ c;
        c = a ^ c;

    }

    if (b < c) {
        c = b ^ c;
        b = b ^ c;
        c = b ^ c;
    }
    printf("%d %d %d\n", a, b, c);
    return 0;
}

// 法二：这样写更简洁一些
#define SWAP(x,y) (x)^=(y);(y)^=(x);(x)^=(y);
int main() {
    int a, b, c;
    scanf("%d %d %d", &a, &b, &c);
    if (a < b)
        SWAP(a, b);
    if (a < c)
        SWAP(a, c);
    if (b < c)
        SWAP(b, c);
    printf("%d %d %d\n", a, b, c);
    return 0;
}
```

37、题目：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>

int main() {
    int arr[10] = { 7,2,8,1,5,10,4,3,9,6 };
    int i = 0, min = arr[0], max = arr[0], tmp, len = sizeof(arr) /
sizeof(arr[0]);

    // 记录最大的与最小的下标
    int maxi, mini;
    for (i = 0; i < len;i++) {
        if (arr[i] > max) {
            max = arr[i];
            maxi = i;
        }
        if (arr[i] < min) {
            min = arr[i];
            mini = i;
        }
    }
    // 交换
    // 最大的与第一个交换
    tmp = arr[maxi];
    arr[maxi] = arr[0];
    arr[0] = tmp;

    // 最小的与最后一个元素交换
    tmp = arr[mini];
    arr[mini] = arr[len - 1];
    arr[len - 1] = tmp;

    for (i = 0; i < 10;i++) {
        printf("%d ", arr[i]);
    }

    return 0;
}
```

38、题目：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。

```
#include <stdio.h>

// 法一：
int main()
{
    int i, n, m;
```

```
    printf("要输入的数字个数：");
    scanf("%d", &n);

    int* num = (int*)malloc(sizeof(int) * n);
    if (num == NULL)return 0;

    printf("输入数字：");
    for (i = 0;i < n;i++)
        scanf("%d", &num[i]);
    printf("再输入要旋转几次：");
    scanf("%d", &m);
    m %= n;

    for (i = 0; i < m;i++) {
        int* p = num + n - 1;
        int tmp = *p;   // 保存最后一位
        while (p > num) { // 移动数据
            *p = *(p - 1);
            p--;
        }
        *p = tmp;
    }

    for (i = 0;i < n;i++)
        printf("%d ", num[i]);
    free(num);
    return 0;
}


// 法二：
void reverse(int* nums, int begin, int end) {
    while (begin < end) {
        int tmp = nums[begin];
        nums[begin] = nums[end];
        nums[end] = tmp;
        begin++;
        end--;
    }
}
int main()
{
    int i, n, m;
    printf("要输入的数字个数：");
    scanf("%d", &n);

    int* num = (int*)malloc(sizeof(int) * n);
    if (num == NULL)return 0;

    printf("输入数字：");
    for (i = 0;i < n;i++)
        scanf("%d", &num[i]);
    printf("再输入要旋转几次：");
    scanf("%d", &m);
    m %= n;
    reverse(num, 0, n - m - 1);
```

```
    reverse(num, n - m, n - 1);
    reverse(num, 0, n - 1);

    for (i = 0;i < n;i++)
        printf("%d ", num[i]);
    free(num);
    return 0;
}
```

39、题目：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。

```
int main() {
    int n, i = 0;
    printf("输入n个人：");
    scanf("%d", &n);
    int* a = (int*)malloc(sizeof(int) * n);
    //编号
    for (int i = 0; i < n; ++i)
        a[i] = i + 1;

    int cut = 0; // 报数
    i = 0;
    int num = n;  // 未挂人数
    while (num > 1) {
        if (a[i]) {
            cut++;
            if (cut == 3) {
                a[i] = 0;
                cut = 0;
                num--;
            }
        }
        i++;
        if (i == n)
            i = 0;
    }
    for (i = 0;i < n;i++)
        if (a[i])
            printf("最后留下来的是%d\n", a[i]);
    free(a);
    a = NULL;
    return 0;
}

// 法二：
int main()
{
    int arr[10] = { 1,2,3,4,5,6,7,8 };
    int i = 0;
```

```
    int j = 0;
    int count = 0;
    int len = sizeof(arr) / sizeof(int);
    for (j = 0; j < 5; j++)
    {
        printf("第%d回结束之后：", j + 1);
        for (i = 0; i < len; i++)
        {
            if (arr[i] != 0)
            {
                count++;
                if (count % 3 == 0)
                {
                    arr[i] = 0;
                    continue;
                }
                printf("%-4d", arr[i]);
            }
        }
        printf("\n");
    }
    return 0;
}
```

40、题目：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。

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

int main() {
    int my_strlen(char*);
    char* str = (char*)malloc(sizeof(char) * 100);
    scanf("%s", str);

    printf("字符串长度为：%d\n", my_strlen(str));

    free(str);
    str = NULL;
    return 0;
}

int my_strlen(const char* s) {
    assert(s);
    int count = 0;
    while (*s++ != '\0') count++;
    return count;
}
```

41、题目：编写input()和output()函数输入，输出5个学生的数据记录。

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

struct Stu {
    char name[20];
    char sex[3];
    int age;
};

void input(struct Stu** s, int* n) {
    printf("请输入学生姓名、性别、年龄：\n");
    for (int i = 0; i < *n;i++)
        scanf("%s %s %d", (*s + i)->name, &(*s + i)->sex, &(*s + i)->age);
}
void output(struct Stu** s, int* n) {
    printf("打印学生姓名、性别、年龄：\n");
    for (int i = 0; i < *n;i++)
        printf("%s %s %d\n", (*s + i)->name, (*s + i)->sex, (*s + i)->age);
}
int main() {
    int n;
    printf("请输入要几输入的几位：");
    scanf("%d", &n);
    struct Stu* s = (struct Stu*)malloc(sizeof(struct Stu) * n);

    input(&s, &n);
    output(&s, &n);

    free(s);
    s = NULL;
    return 0;
}
```

42、题目：创建一个链表。

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

struct LNode
{
    int data;
    struct LNode *next;
};

LNode* createList(int n)
{
    LNode *list, *p, *q;
    list = (LNode*)malloc(sizeof(LNode));
    list->next = nullptr;
```

```
    q = list;
    for (int i = 0; i < n; ++i)
    {
        p = (LNode*)malloc(sizeof(LNode));
        printf("请输入第%d个元素的值：", i + 1);
        scanf("%d", &(p->data));
        p->next = nullptr;
        q->next = p;
        q = p;
    }
    return list;
}

void print(LNode* list)
{
    printf("链表各值为:\n");

    LNode* p = list->next;
    while (p != nullptr)
    {
        printf("%d\n", p->data);
        p = p->next;
    }
}

int main()
{
    LNode* list = nullptr;
    int n;
    scanf("%d", &n);
    list = createList(n);
    print(list);

    return 0;
}
```

43、题目：反向输出一个链表。

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

struct LNode
{
    int data;
    struct LNode *next;
};

LNode* createList(int n)
{
    LNode *head, *p, *q;
    head = (LNode*)malloc(sizeof(LNode));
    printf("请输入第1个元素的值：");
    scanf("%d", &(head->data));
    head->next = nullptr;
```

```
    q = head;
    for (int i = 1; i < n; ++i)
    {
        p = (LNode*)malloc(sizeof(LNode));
        printf("请输入第%d个元素的值：", i + 1);
        scanf("%d", &(p->data));
        p->next = q;
        q = p;
    }
    return q;
}

void print(LNode* list)
{
    printf("链表各值为:\n");

    LNode* p = list;
    while (p != nullptr)
    {
        printf("%d\n", p->data);
        p = p->next;
    }
}

int main()
{
    int n;
    scanf("%d", &n);
    LNode* list = createList(n);
    print(list);

    return 0;
}
```

44、题目：连接两个链表。

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

struct LNode
{
    int data;
    struct LNode *next;
};

LNode* createList(int n)
{
    LNode *list, *p, *q;
    list = (LNode*)malloc(sizeof(LNode));
    list->next = nullptr;
    q = list;
    for (int i = 0; i < n; ++i)
    {
```

```
        p = (LNode*)malloc(sizeof(LNode));
        printf("请输入第%d个元素的值：", i + 1);
        scanf("%d", &(p->data));
        p->next = nullptr;
        q->next = p;
        q = p;
    }
    return list;
}

void print(LNode* list)
{
    printf("链表各值为:\n");

    LNode* p = list->next;
    while (p != nullptr)
    {
        printf("%d\n", p->data);
        p = p->next;
    }
}


void connect(LNode* list1, LNode* list2)
{
    LNode* p;
    for (p = list1; p->next != nullptr; p = p->next)
    {
        continue;
    }
    p->next = list2->next;
    free(list2);
}

int main()
{
    int n;
    scanf("%d", &n);
    LNode* list1 = createList(n);
    LNode* list2 = createList(n);
    connect(list1, list2);

    print(list1);

    return 0;
}
```

45、题目：输入一个整数，并将其反转后输出。

```c
#include<stdio.h>

int main() {
    int n, newn = 0, x;
    scanf("%d", &n);
    while (n) {
        x = n % 10;
        newn = newn * 10 + x;
        n /= 10;
    }
    printf("反转的数为：%d", newn);
    return 0;
}
```

46、题目：编写一个函数，输入n为偶数时，调用函数求1/2+1/4+...+1/n,当输入n为奇数时，调用函数1/1+1/3+...+1/n(利用指针函数)。

```c
#include<stdio.h>

double sumup(int n, int flag) {
    double sum = 0;
    if (flag == 2) {
        for (int i = flag;i <= n;i += 2) {
            sum += (double)1 / i;
        }
    }
    else {
        for (int i = flag;i <= n;i += 2) {
            sum += (double)1 / i;
        }
    }
    return sum;
}

int main() {
    int n, flag = 0;
    double (*psum)(int, int);
    scanf("%d", &n);
    if (n % 2 == 0) { // n为偶数
        psum = sumup;
        flag = 2;
    }
    else { // n为奇数
        psum = sumup;
```

```
        flag = 1;
    }
    printf("%lf", (*psum)(n, flag));
    return 0;
}
```

47、题目：找到年龄最大的人，并输出。

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

typedef struct man {
    char name[20];
    int  age;
}man;

int main() {
    man person[] = { {"zhanagsan",18},{"lisi",30},{"wangwu",25} };
    man* p = person, * q = NULL;
    int max = person[0].age;
    int count = sizeof(person) / sizeof(person[0]);
    while (count--) {
        if (max < p->age) {
            max = p->age;
            q = p;
        }
        p++;
    }
    printf("年龄最大的人是%s，%d", q->name, q->age);
    return 0;
}
```

48、题目：字符串排序。

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

int compare(const void* a, const void* b) {
    return strcmp(*(const char**)a, *(const char**)b);
}

int main() {
    // 定义字符串数组
    char* str[] = { "ac", "abc", "zz", "yyy", NULL };
    // 计算数组大小（忽略 NULL）
    int n = 0;
    while (str[n] != NULL) n++;

    // 使用 qsort 排序字符串指针数组
    qsort(str, n, sizeof(char*), compare);

    for (int i = 0; i < n; i++)
        printf("%s ", str[i]);
```

```
    return 0;
}
```

49、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?

```c
#include <stdio.h>

int main() {
    int N = 1;  // 从1开始检查最小桃子数

    while (1) {
        int peaches = N;
        int valid = 1;

        // 模拟每只猴子分桃子的过程
        for (int i = 0; i < 5; i++) {
            if ((peaches - 1) % 5 != 0) {
                valid = 0;
                break;
            }
            peaches = (peaches - 1) / 5 * 4;
        }

        // 如果条件满足，输出结果
        if (valid) {
            printf("海滩上原来最少有 %d 个桃子。\n", N);
            break;
        }

        // 否则，增加N继续检查
        N++;
    }

    return 0;
}
```

50、题目：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。

```
#include <stdio.h>

int main() {
    for (int i = 10;i <= 99;i++) {
        if ((809 * i >= 1000 && 809 * i <= 10000) && (8 * i <= 99 && 9 * i >=
100)) {
            printf("%d = %d*%d + %d*%d", 809 * i, 800, i, 9, i); // 9708 = 800*12
+ 9*12
        }
    }
    return 0;
}
```

51、题目：八进制转换为十进制。

```
#include <stdio.h>

// 法一
int main()
{
    int n;
    scanf("%o", &n);
    printf("%d", n);
    return 0;
}

// 法二：
#include <stdio.h>
#include <string.h>

int main()
{
    char s[50];
    scanf("%s", s);
    int n = 0;
    for (int i = 0; i < strlen(s); i++)
    {
        n = n * 8 + s[i] - '0';
    }

    printf("%d", n);

    return 0;
}
```

52、题目：求0—7所能组成的奇数个数。

```
#include <stdio.h>

int main()
{
    int sum = 0, cnt = 0;
    for (int i = 0; i < 8; ++i)
    {
        if (i == 0)
        {
            cnt = 4;
        }
        else if (i == 1)
        {
            cnt *= 7;
        }
        else
        {
            cnt *= 8;
        }
        sum += cnt;
        printf("%d位数为奇数的个数为%d\n", i + 1, cnt);
    }

    printf("总数为%d\n", sum);

    return 0;
}
```

53、题目：一个偶数总能表示为两个素数之和。

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

int ver(int num)
{
    for (int i = 2; i <= sqrt(num); ++i)
    {
        if (num % i == 0)
        {
            return 0;
        }
    }
    return 1;

}

int main()
{
    int n;
```

```
    scanf("%d", &n);

    if (n % 2 != 0 || n == 2)
    {
        printf("请输入一个大于2的偶数！\n");
    }
    else
    {
        for (int i = 2; i < n; i++)
        {
            if (ver(i) && ver(n - i)) //
            {
                printf("%d = %d + %d\n", n, i, n - i);
                return 0;
            }
        }
    }

    return 0;
}
```

54、题目：判断一个素数能被几个9整除。

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

int is_prime(int num)
{
    for (int i = 2; i < sqrt(num); i++)
    {
        if (num % i == 0)
            return 0;
    }
    return 1;
}

int main()
{
    int x,i;
    scanf("%d", &i);
    if (is_prime(i))
    {
        for (x = 9; x % i != 0; x = x * 10 + 9)
            if (x > 1000000000)
                return 0;
        printf("素数%d可以被%d整除\n", i, x); // 素数13可以被999999整除
    }
    return 0;
}
```

55、题目：两个字符串连接。

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

```c
#include <assert.h>

char* my_strcat(char* s1, const char* s2) {
    assert(s1 && s2);
    int i = 0;
    int len = strlen(s1);
    while (s1[len++] = s2[i++]);
    return s1;
}

int main() {
    char str1[20] = "abc";
    char str2[10] = "def";

    printf("%s", my_strcat(str1, str2));
    return 0;
}
```

56、题目：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*。

```c
#include<stdio.h>

int main() {
    int n, count, j;
    printf("请输入7个数（1-50）的整数值：");
    count = 0;
    while(count < 7){
        scanf("%d", &n);
        if (n < 1 || n > 50) {
            printf("请输入7个数（1-50）的整数值：");
        }
        else
        {
            for (j = 0;j < n;j++)
                printf("*");
            count++;
            printf("\n");
        }
    }
    return 0;
}
```

57、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。

```
#include<stdio.h>

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

int main() {
    int n, i = 0, a[4];
    scanf("%d", &n);

    for (i = 0;i < 4;i++) {
        a[3 - i] = n % 10;
        n /= 10;
        a[3 - i] += 5; // 每位数字都加上5
        a[3 - i] %= 10; // 然后用和除以10的余数代替该数字
    }
    swap(&a[0], &a[3]); // 再将第一位和第四位交换
    swap(&a[1], &a[2]); // 第二位和第三位交换

    for (i = 0;i < 4;i++) {
        printf("%d", a[i]); /* 1234
                               9876*/
    }
    return 0;
}
```

58、题目：猜谜游戏。

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

//设置猜的数字次数多少
#define NUM 10

void menu()
{
    printf("**********************\n");
    printf("******  1.play *******\n");
    printf("******  0.exit *******\n");
```

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

void game()
{
    int guess = 0;
    int ret = rand() % 100 + 1;
    int count = NUM;
    while (1)
    {
        printf("你还有%d次机会！\n", count);
        printf("请猜数字(1~100):>");
        scanf("%d", &guess);
        if(guess < 1 || guess > 100)
        {
            printf("输入错误，请重新猜数字！\n");
            continue;
        }
        else if (guess > ret)
        {
            printf("猜大了！\n");
        }
        else if (guess < ret)
        {
            printf("猜小了！\n");
        }
        else
        {
            printf("恭喜你，猜对了！\n");
            break;
        }
        count--;
        if (count == 0)
        {
            printf("次数用完，正确的数字是%d\n", ret);
            break;
        }
    }
}
int main()
{
    int input = 0;
    srand((unsigned int)time(NULL));//生成随机数
    do
    {
        menu();
        printf("请选择：>");
        scanf("%d", &input);
        switch (input)
        {
        case 1:
            game();
            break;
        case 0:
            printf("退出游戏！\n");
            break;
```

```
        default:
            printf("选择错误，请重新选择！\n");
            break;
        }
    } while (input);

    return 0;
}
```

59、题目：计算字符串中子串出现的次数。

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

int main() {
    int len1, len2, i = 0, j = 0, count = 0;
    char* str1 = "abcdababcaacabc";
    char* str2 = "bc";
    len1 = strlen(str1);
    len2 = strlen(str2);

    while (i < len1 && j < len2)
    {
        if (str1[i] == str2[j])
        {
            i++;
            j++;
            if (j == len2)
            {
                j = 0;
                count++; // 次数++
            }
            continue;    // 如果想等就继续运行，不加这个i就会加两次
        }
        i++;
    }
    printf("%s出现的次数有%d\n", str2, count);
    return 0;
}
```

60、题目：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。

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

int main() {
    char c;

    // 打开文件
    FILE* pf = fopen("data.txt", "w+");
    if (pf == NULL) {
```

```
        perror("Open File fail!");
        exit(-1);
    }

    // 输入文件内容
    c = getchar();
    while (c != '#') {
        fputc(c, pf);
        c = getchar();
    }

    // 关闭文件
    fclose(pf);
    pf = NULL;
    return 0;
}
```

61、题目：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。

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

int main() {
    int i;
    FILE* pf = fopen("test", "w+");
    if (pf == NULL) {
        perror("Open File fail!");
        exit(-1);
    }

    char str[255] = "";
    scanf("%s", str);
    int len = strlen(str);

    // 将小写字母全部转换成大写字母
    for (i = 0;i < len;i++) {
        if (str[i] >= 'a' && str[i] <= 'z') {
            str[i] -= 32;
            // 直接输入到磁盘test中保存
            fputc(str[i], pf);
        }
    }

    // 关闭文件
    fclose(pf);
    pf = NULL;
    return 0;
}
```

```c
int main() {
    int i;
    FILE* pf = fopen("test", "w+");
    if (pf == NULL) {
        perror("Open File fail!");
        exit(-1);
    }

    char str[255] = "";
    scanf("%s", str);
    int len = strlen(str);

    // 将小写字母全部转换成大写字母
    for (i = 0;i < len;i++) {
        if (str[i] >= 'a' && str[i] <= 'z') {
            str[i] -= 32;
        }
    }

    fprintf(pf, "%s", str);

    // 关闭文件
    fclose(pf);
    pf = NULL;
    return 0;
}
```

62、题目：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。

先创建a.txt写入xyz
再创建b.txt写入abc

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

FILE* open(char* s, char* mode) {
    FILE* p = fopen(s, mode);
    if (p == NULL) {
        perror("Open File fail!");
        exit(-1);
    }
    return p;
}

void close(FILE* p) {
    fclose(p);
    p = NULL;
}
```

```c
void read(FILE* p, char* s) {
    fgets(s, 50, p);
}

int cmp_char(const void* e1, const void* e2)
{
    return strcmp(e1, e2);
}

void merge(char* s, int len) {
    qsort(s, len, sizeof(s[0]), cmp_char);
}

int main() {
    // 有两个磁盘文件A和B 先创建a.txt写入xyz, 再创建b.txt写入abc
    FILE* pA = open("a.txt", "w+");
    fprintf(pA, "xyz");
    FILE* pB = open("b.txt", "w+");
    fprintf(pB, "abc");

    // 确保文件指针在开头
    rewind(pA);
    rewind(pB);

    // 读取取这两个文件中的数据到数组
    char strA[100], strB[50];
    read(pA, strA);
    read(pB, strB);

    // 要求把这两个文件中的信息合并（按字母顺序排列)
    strcat(strA, strB);
    int len = strlen(strA);
    merge(strA, len);

    // 打开文件c
    FILE* pC = open("c.txt", "w+");
    fprintf(pC, "%s", strA); // c.txt中已写入abcxyz
    // fputs(strA, pC);

    // 关闭文件
    close(pA);
    close(pB);
    close(pC);
    return 0;
}
```

63、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，况原有的数据和计算出的平均分数存放在磁盘文件"stud"中。

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

typedef struct Student
{
    int id;
    char name[20];
    int grade1;
    int grade2;
    int grade3;
    int averageGrade;
}Student;

int main()
{
    Student stu[5];
    for (int i = 0; i < 5; ++i)
    {
        scanf("%d %s %d %d %d", &stu[i].id, &stu[i].name, &stu[i].grade1,
&stu[i].grade2, &stu[i].grade3);
        stu[i].averageGrade = (stu[i].grade1 + stu[i].grade2 + stu[i].grade3) /
3;
    }
    /* 输入
    1 a 10 20 30
    2 b 20 30 40
    3 c 30 40 50
    4 d 40 50 60
    5 e 50 60 70
    */
    FILE* fp = fopen("stud.txt", "w");
    if (fp == NULL)
    {
        perror("Open File fail!");
        exit(-1);
    }

    for (int i = 0; i < 5; ++i)
        fprintf(fp, "%d %s %d %d %d %d\n", stu[i].id, stu[i].name, stu[i].grade1,
stu[i].grade2, stu[i].grade3, stu[i].averageGrade);
    /* 文件中输出
    1 a 10 20 30 20
    2 b 20 30 40 30
    3 c 30 40 50 40
    4 d 40 50 60 50
```

```
    5 e 50 60 70 60
    */
    fclose(fp);
    fp = NULL;
    return 0;
}
```

64、题目：喝汽水，1瓶汽水1元，2个空瓶可以换一瓶汽水，给20元，可以喝多少汽水（编程实现）。

```c
#include <stdio.h>

// 方法一：
int main() {
    int money = 20;
    int total = money;
    int empty = money;

    while (empty >= 2) {
        total += empty / 2;
        empty = empty / 2 + empty % 2;
    }

    printf("总共可以喝 %d 瓶汽水\n", total);
    return 0;
}
```

```c
// 方法二：
// 其实就是个等差数列：money * 2 - 1
int main() {
    int money = 20;
    printf("总共可以喝 %d 瓶汽水\n", money * 2 - 1);
    return 0;
}
```

65、题目：递归和非递归分别实现求第n个斐波那契数

```c
#include <stdio.h>

// 递归
int f(int i) {
    if (i == 1 || i == 2)
        return 1;
    else
        return f(i - 1) + f(i - 2);
}

int main() {
```

```
    int n;
    scanf("%d", &n);
    printf("%d ", f(n));
    return 0;
}

// 非递归
int main() {

    int n, i, a = 1, b = 1,c = 1;
    scanf("%d", &n);

    if (n <= 2) {
        c = 1;
    }
    else {
        for (i = 2; i <= n;i++) {
            c = a + b;
            a = b;
            b = c;
        }
    }
    printf("%d ", a);
    return 0;
}
```

66、题目：编写一个函数实现n的k次方，使用递归实现。

```
int f(int n, int k) {
    if (k == 0)
        return 1;
    else if (k == 1)
        return n;
    else
        return n * f(n, k - 1);
}

int main() {
    int n = 10, k = 2;
    printf("%d", f(n, k));
    return 0;
}
```

67、题目：写一个递归函数`DigitSum(n)`，输入一个非负整数，返回组成它的数字之和

例如，调用`DigitSum(1729)`，则应该返回1 + 7 + 2 + 9，它的和是19
输入：1729，输出：19

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

```c
int DigitSum(int n) {
    if (n < 10)
        return n;
    else
        return n % 10 + DigitSum(n / 10);
}

int main() {
    int n = 1729;
    printf("%d", DigitSum(n));
    return 0;
}
```

68、题目：递归和非递归分别实现求n的阶乘（不考虑溢出的问题）

题目：递归和非递归分别实现求n的阶乘（不考虑溢出的问题）

```c
#include <stdio.h>

// 递归
int f(int n) {
    if (n == 1 || n == 0)
        return 1;
    else
        return n * f(n - 1);
}

int main() {
    int n = 5;
    printf("%d", f(n));
    return 0;
}
```

```c
// 非递归
int main() {
    int n = 5;
    int sum = 1;
    for (int i = 1;i <= 5;i++)
    {
        sum *= i;
    }
    printf("%d", sum);
    return 0;
}
```

69、题目：递归方式实现打印一个整数的每一位

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

```c
// 法一：
int f(int n) {
    if (n == 0)
        return 0;
    else
        return f(n / 10), printf("%d ", n % 10);
}

int main() {
    int n = 12345;
    f(n);
    return 0;
}
```

```c
// 法二：
int f(int n) {
    if (n > 9)
        f(n / 10);
    printf("%d ", n % 10);
}

int main() {
    int n = 12345;
    f(n);
    return 0;
}
```

70、题目：找单身狗1

在一个整型数组中，只有一个数字出现一次，其他数组都是成对出现的，请找出那个只出现一次的数字。

例如：数组中有：`1 2 3 4 5 1 2 3 4`，只有5出现一次，其他数字都出现2次，找出5

```c
#include <stdio.h>

int main() {
    int arr[] = { 1, 2, 3, 4, 5, 1, 2, 3, 4 };
    int x = 0;
    for (int i = 0;i < sizeof(arr) / sizeof(arr[0]);i++) {
        x ^= arr[i];
    }
    printf("%d", x);
    return 0;
}
```

71、题目：不允许创建临时变量，交换两个整数的内容

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

```c
int main() {
    int a = 10;
    int b = 20;

    printf("交换前：a = %d,b = %d\n", a, b);
    a ^= b;
    b ^= a;
    a ^= b;
    printf("交换后：a = %d,b = %d\n", a, b);

    return 0;
}
```

72、题目：输入一个整数 n ，输出该数32位二进制表示中1的个数。其中负数用补码表示。

输入：10

返回值：2
说明：十进制中10的32位二进制表示为`0000 0000 0000 0000 0000 0000 0000 1010`，其中有两个1。

输入：-1

返回值：32

```c
#include <stdio.h>

int NumberOf1(int n) {
    int count = 0;
    for (int i = 0; i < 32; i++) {
        if ((n >> i) & 1)
            count++;
    }
    printf("%d", count);
}

int main() {
    NumberOf1(-1);
    return 0;
}
```

73、题目：获取一个整数二进制序列中所有的偶数位和奇数位，分别打印出二进制序列

```c
int main() {
    int n = 10;
    int i;
    // 偶数位
    for (i = 31;i >= 1;i -= 2) {
```

```
        printf("%d ", (n >> i) & 1);
    }
    printf("\n");
    // 奇数位
    for (i = 30;i >= 0;i -= 2) {
        printf("%d ", (n >> i) & 1);
    }
    return 0;
}
```

74、题目：输入两个整数，求两个整数二进制格式有多少个位不同

```
#include <stdio.h>

int main() {
    int n, m;
    scanf("%d %d", &n, &m);
    int i, count = 0;
    for (i = 0; i < 32; i++) {
        if (((n >> i) & 1) != ((m >> i) & 1)) {
            count++;
        }
    }
    printf("%d", count);
    return 0;
}
```

思路：

1. 先将m和n进行按位异或，此时m和n相同的二进制比特位清零，不同的二进制比特位为1
2. 统计异或完成后结果的二进制比特位中有多少个1即可

```
#include <stdio.h>
int calc_diff_bit(int m, int n)
{
    int tmp = m ^ n;
    int count = 0;
    while (tmp)
    {
        tmp = tmp & (tmp - 1);
        count++;
    }
    return count;
}

int main()
{
    int m, n;
    while (scanf("%d %d", &m, &n) == 2)
    {
        printf("%d\n", calc_diff_bit(m, n));
    }
```

```
    return 0;
}
```

75、题目：下面代码的结果是：

```
#include <stdio.h>
int i;
int main()
{
    i--;
    if (i > sizeof(i))
    {
        printf(">\n");
    }
    else
    {
        printf("<\n");
    }
    return 0;
}
```

> A.>

> B.<

> C.不输出

> D.程序有问题

答案:

C语言中，0为假，非0即为真。

全局变量，没有给初始值时，编译其会默认将其初始化为0。

- i的初始值为0，i--结果-1，i为整形，`sizeof(i)`求i类型大小是4，按照此分析来看，结果应该选择B，但是`sizeof`的返回值类型实际为无符号整形，因此编译器会自动将左侧i自动转换为无符号整形的数据，-1对应的无符号整形是一个非常大的数字，超过4或者8，故实际应该选择`A`

这道题其实很隐蔽，真是虾仁猪心！！！

76、题目：实现一个函数，可以左旋字符串中的k个字符。

例如：

ABCD左旋一个字符得到BCDA

ABCD左旋两个字符得到CDAB

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

void reverse_1(char* s, int k) {
```

```
    int len = strlen(s);
    k %= len;
    int i, j, t;
    for (j = 1;j <= k;j++) {
        t = s[0];
        for (i = 0;i < len - 1;i++) {
            s[i] = s[i + 1];
        }
        s[i] = t;
    }
}

void reverse_2(char* s, int k) {
    int len = strlen(s);
    int pos = k % len;
    char* t = (char*)malloc(sizeof(s) * len + 1);

    strcpy(t, s + pos);
    strncat(t, s, pos);
    strcpy(s, t);

    free(t);
    t = NULL;
}

void reverse_3(char* s, int begin, int end) {
    char tmp;
    while (begin < end) {
        tmp = s[begin];
        s[begin] = s[end];
        s[end] = tmp;
        begin++;
        end--;
    }
}
int main() {
    char str[] = "abcde";
    int k = 4;
    // 方法一：
    // reverse_1(str, k);

    // 方法二：
    // reverse_2(str, k);

    // 方法三：
    int len = strlen(str);
    int pos = k % len;
    reverse_3(str, 0, pos - 1);  // 逆序前段
    reverse_3(str, pos, len - 1);// 逆序后段
    reverse_3(str, 0, len - 1);  // 整体逆序

    printf("%s", str);
    return 0;
}
```

77、题目：模拟实现库函数strlen

```c
#include <stdio.h>

unsigned int my_strlen_1(const char* s) {
    assert(s);
    unsigned int count = 0;
    const char* p = s;
    while (*p) {
        count++;
        p++;
    }
    return count;
}

unsigned int my_strlen_2(const char* s) {
    assert(s);
    const char* end = s;
    while (*end++);
    return end - s - 1;
}

int main() {
    char* s = "abcedf";
    //方法一：
    // int len = my_strlen_1(s);

    // 方法二：
    unsigned int len = my_strlen_2(s);
    printf("%d\n", len);
    return 0;
}
```

78、题目：调整数组使奇数全部都位于偶数前面。

输入一个整数数组，实现一个函数，

来调整该数组中数字的顺序使得数组中所有的奇数位于数组的前半部分，

所有偶数位于数组的后半部分。

```c
#include <stdio.h>

int main() {
    int a[10] = { 1,2,3,4,5,6,7,8,9,10 };
    int len = sizeof(a) / sizeof(a[0]);
    int begin = 0, end = len - 1,tmp;
    while (begin < end) {
        // 找偶数
        while (begin < end && a[begin] % 2 == 1) begin++;
        // 找奇数
        while (begin < end && a[end] % 2 == 0) end--;
```

```
        // 找到了，交换
        if (begin < end) {
            tmp = a[begin];
            a[begin] = a[end];
            a[end] = tmp;
        }
    }

    for (begin = 0;begin < len;begin++) {
        printf("%d ", a[begin]);
    }
    return 0;
}
```

79、题目：写一个函数打印arr数组的内容，不使用数组下标，使用指针。

```
#include <stdio.h>

int main() {
    int arr[10] = { 1,2,3,4,5,6,7,8,9,10 };
    int* p = arr;
    for (int i = 0;i < sizeof(arr) / sizeof(arr[0]);i++, p++) {
        printf("%d ", *p);
    }
    return 0;
}
```

80、题目：实现一个对整形数组的冒泡排序

```
#include <stdio.h>

enum SORT {
    RISE,
    FALL
};

void bbuble_sort(int arr[], int len, int sort) {
    int i, j,tmp, flag = 1;
    for (i = 0;i < len - 1;i++) {
        flag = 0;
        for (j = 0;j < len - 1 - i;j++) {
            if (sort == RISE && arr[j] > arr[j + 1]) {
                tmp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = tmp;
                flag = 1;
            }
            else if (sort == FALL && arr[j] < arr[j + 1])
            {
                tmp = arr[j];
```

```
                arr[j] = arr[j + 1];
                arr[j + 1] = tmp;
                flag = 1;
            }
        }
        if (!flag) return;
    }
}

int main() {
    int arr[10] = { 2,4,6,8,10,1,3,5,7,9 };
    int len = sizeof(arr) / sizeof(arr[0]);

    bbuble_sort(arr,len, RISE); // 升序
    bbuble_sort(arr,len, FALL); // 降序

    for (int i = 0;i < len;i++) {
        printf("%d ", arr[i]);
    }
    return 0;
}
```

81、题目：写一个函数，判断一个字符串是否为另外一个字符串旋转之后的字符串。

例如：给定s1 = AABCD和s2 = BCDAA，返回1
给定s1 = abcd和s2 = ACBD，返回0.
AABCD左旋一个字符得到ABCDA
AABCD左旋两个字符得到BCDAA
AABCD右旋一个字符得到DAABC

```
#include <stdio.h>

int find_substr(const char* s1, const char* s2, char* s3) {
    assert(s1 && s2 && s3);
    strcpy(s3, s1);
    strcat(s3, s1); // 总共连接两次
    int len_2 = strlen(s2);
    int len_3 = strlen(s3);

    int i = 0, j = 0;
    while (i < len_3 && j < len_2)
    {
        if (s3[i] == s2[j])
        {
            i++;
            j++;
            if (j == len_2)
                return 1;
            if (s3[i] != s2[j])  // 如果不等于就让j保持到一个位置
                j = 0;
            continue;    // 如果想等就继续运行，不加这个i就会加两次
        }
```

```
        i++;
    }
    return 0;
}

int main() {
    char s1[] = "AABCD";
    char s2[] = "BCDAA";
    char* s3 = (char*)malloc(sizeof(char) * (sizeof(s1) * 2));
    if (s3 == NULL) return;

    int flag = find_substr(s1, s2, s3);
    printf("%d\n", flag);

    free(s3);
    s3 = NULL;
    return 0;
}
```

82、题目：有一个数字矩阵，矩阵的每行从左到右是递增的，矩阵从上到下是递增的，请编写程序在这样的矩阵中查找某个数字是否存在。

要求：时间复杂度小于O(N);

```
#include <stdio.h>

int main() {
    int i = 0, j = 3 - 1;   //从右上角开始遍历
    int a[][3] = { {1, 2, 3},
                   {3, 4, 7},
                   {7, 8, 9}
    };
    int x = 4; // 要找的数字

    while (i < 3 && j >= 0) {
        if (a[i][j] < x)
            i++;
        else if (a[i][j] > x)
            j--;
        else
            break;
    }

    if (a[i][j] == x)
        printf("找到了\n");
    else
        printf("找不到\n");
    return 0;
}
```

83、题目：日本某地发生了一件谋杀案，警察通过排查确定杀人凶手必为4个嫌疑犯的一个。

以下为4个嫌疑犯的供词:

A说：不是我。

B说：是C。

C说：是D。

D说：C在胡说

已知3个人说了真话，1个人说的是假话。

现在请根据这些信息，写一个程序来确定到底谁是凶手。

```
#include <stdio.h>

int main()
{
    int killer = 0;
    // 分别假设凶手是a,b,c,d,看谁是凶手时满足3个人说了真话，一个人说了假话
    for (killer = 'a'; killer <= 'd'; killer++)
    {
        if ((killer != 'a') + (killer == 'c') + (killer == 'd') + (killer != 'd')
== 3)
            printf("凶手是：%c", killer);
    }
    return 0;
}
```

84、题目：在屏幕上打印杨辉三角。

```
#include <stdio.h>

// 法一：
int main() {
    int arr[10][10] = { 0 };
    int i, j;
    for (i = 0;i < 10;i++) {
        for (j = 0;j <= i;j++) {
            arr[i][0] = 1;
            if (i == j)
                arr[i][j] = 1;
        }
    }

    for (i = 2;i < 10;i++) {
        for (j = 1;j < i;j++) {
            arr[i][j] = arr[i - 1][j] + arr[i - 1][j - 1];
        }
```

```
    }

    for (i = 0;i < 10;i++) {
        for (j = 0;j <= i;j++) {
            printf("%-5d", arr[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

```
#include <stdio.h>

// 法二
int main() {
    int arr[10][10] = { 1 };
    int i, j;

    for (i = 1;i < 10;i++) {
        arr[i][0] = 1;
        for (j = 1;j <= i;j++) {
            arr[i][j] = arr[i - 1][j] + arr[i - 1][j - 1];
        }
    }

    for (i = 0;i < 10;i++) {
        for (j = 0;j <= i;j++) {
            printf("%-5d", arr[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

85、题目：转移表代码实现

```
#include <stdio.h>

void menu()
{
    printf("*************************\n");
    printf("********1:add 2:sub****** \n");
    printf("********3:mul 4:div******\n");
    printf("********0:exit     ******\n");
    printf("*************************\n");
}

int add(int a, int b)
{
    return a + b;
}
int sub(int a, int b)
{
    return a - b;
```

```
}
int mul(int a, int b)
{
    return a * b;
}
int div_(int a, int b)
{
    if (b == 0)
        exit(-1);
    return a / b;
}
int main()
{
    int x, y;
    int input = 1;
    int ret = 0;
    int(*pfArr[5])(int x, int y) = { 0, add, sub, mul, div_ }; //转移表
    do
    {
        menu();
        printf("请选择: ");
        scanf("%d", &input);
        if ((input <= 4 && input >= 1)) {
            printf("输入操作数: ");
            scanf("%d %d", &x, &y);
            ret = (*pfArr[input])(x, y);
            printf("ret = %d\n", ret);
        }
        else if (input == 0)
            printf("退出计算器\n");
        else
            printf("输入有误,请重新选择\n");
    } while (input);
    return 0;
}
```

86、题目：找单身狗2

一个数组中只有两个数字是出现一次，其他所有数字都出现了两次。

编写一个函数找出这两个只出现一次的数字。

例如：

有数组的元素是：1，2，3，4，5，1，2，3，4，6

只有5和6只出现1次，要找出5和6.

```
#include <stdio.h>

int main() {
    int arr[] = { 1,2,3,4,5,1,2,3,4,6 };
    int i, x = 0, len = sizeof(arr) / sizeof(arr[0]), n = 0, m = 0;

    // 1. 找出两个单身异或的结果
    for (i = 0;i < len;i++) {
```

```
        x ^= arr[i];
    }
    int pos = 0;
    // 2. 算出这结果的哪一位是1，然后记录下来
    for (i = 0;i < 32;i++) {
        if (((x >> i) & 1) == 1)
            pos = i;
    }
    // 3. 再次遍历数组，看哪个数字的pos下标是0还是1，
    for (i = 0;i < len;i++) {
        if (((arr[i] >> pos) & 1) == 1)
            n ^= arr[i];
        else
            m ^= arr[i];
    }
    printf("%d %d", n, m);
    return 0;
}
```

87、题目：获得月份天数

```
#include <stdio.h>

int leap_year(int year) {
    return ((year % 4 == 0) && (year % 100 != 0)) || (year % 400 == 0);
}

int main() {
    int a[] = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
    int month, day = 0, year;
    while (~scanf("%d %d", &year, &month)) {
        if (month == 2 && leap_year(year))
            printf("%d\n", a[month] + 1);
        else
            printf("%d\n", a[month]);
    }
    return 0;
}
```

88、题目：模仿qsort的功能实现一个通用的冒泡排序

```
#include <stdio.h>

void print_arr(int arr[], int sz)
{
    int i = 0;
    for (i = 0; i < sz; i++)
    {
        printf("%d ", arr[i]);
    }
}
```

```
void Swap(char* buf1, char* buf2, size_t size)
{
    for (int i = 0; i < size; i++)
    {
        char tmp = *buf1;
        *buf1 = *buf2;
        *buf2 = tmp;
        ++buf1;
        ++buf2;
    }
}

int cmp_int(const void* e1, const void* e2)
{
    return *(int*)e1 - *(int*)e2;
}

void my_bubble_qsort(void* base, size_t num, size_t size, int(*cmp)(const void*
e1, const void* e2))
{
    int i = 0;
    int j = 0;
    for (i = 0; i < num - 1; i++)
    {
        for (j = 0; j < num - i - 1; j++)
        {
            // 返回了大于0的数就交换
            if (cmp((char*)base + j * size, (char*)base + (j + 1) * size) > 0)
            {
                Swap((char*)base + j * size, (char*)base + (j + 1) * size, size);
            }
        }
    }
}

int main()
{
    int arr[] = { 3,2,5,6,8,7,9,1,4,0 };
    int sz = sizeof(arr) / sizeof(arr[0]);
    my_bubble_qsort(arr, sz, sizeof(arr[0]), cmp_int);
    print_arr(arr, sz);
    return 0;
}
```

89、模拟实现strcpy

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

char* my_strcpy(char* dest, const char* src) {
    assert(dest && src);
    char* ret = dest;
```

```
    while (*dest++ = *src++);

    return ret;
}

int main() {
    char a[20];
    char b[10] = "abcdef";
    my_strcpy(a, b);
    puts(a);
    return 0;
}
```

90、模拟实现strcat

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

char* my_strcat(char* dest, const char* src) {
    assert(dest && src);
    char* ret = dest;
    size_t len = strlen(dest);
    dest += len;
    while (*dest++ = *src++);

    return ret;
}

int main() {
    char a[20] = "abc";
    char b[10] = "abcdef";
    my_strcat(a, b);
    puts(a);
    return 0;
}
```

91、模拟实现strcmp

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

int my_strcmp(const char* dest, const char* src) {
    assert(dest && src);
    while (*dest && *src) {
        if (*dest == *src)
        {
            dest++;
            src++;
        }
        else
        {
```

```
            break;
        }
    }
    if (*dest == *src)
        return 0;
    else if (*dest > *src)
        return 1;
    else
        return -1;
}

int main() {
    char a[20] = "d";
    char b[10] = "abcdef";
    if (my_strcmp(a, b) == 0)
        printf("a == b\n");
    else if (my_strcmp(a, b) > 0)
        printf("a > b\n");
    else
        printf("a < b\n");
    return 0;
}
```

92、模拟实现strstr

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

char* my_strstr(const char* str, const char* substr) {
    assert(str && substr);
    const char* cnt = str;
    if (!*substr) // 字串等于空串直接返回
        return str;
    char* cp = (char*)str;
    char* s1, * s2;
    while (*cp)
    {
        s1 = cp;
        s2 = (char*)substr;
        while (*s1 && *s2 && !(*s1 - *s2))
            s1++, s2++;

        if (!*s2)
            return(cp);
        cp++;
    }
    return NULL;
}

int main() {
    char a[20] = "acdcaccdef";
    char b[10] = "cd";
    printf("%s",my_strstr(a, b));
```

```
    return 0;
}
```

93、模拟实现strncpy

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

char* my_strncpy(char* dest, const char* src, size_t count) {
    assert(dest && src);
    char* ret = dest;
    if (count > strlen(src))
    {
        while (*dest++ = *src++);
    }
    else
    {
        size_t n = count;
        while (n--) {
            *dest++ = *src++;
        }
    }
    *dest = '\0';
    return ret;
}

int main() {
    char a[20];
    char b[10] = "abcdef";
    my_strncpy(a, b,0);
    puts(a);
    return 0;
}
```

94、模拟实现strncat

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

char* my_strncat(char* dest, const char* src, size_t count) {
    assert(dest && src);
    char* ret = dest;
    size_t len_dest = strlen(dest);
    dest += len_dest;
    if (count > strlen(src))
    {
        while (*dest++ = *src++);
    }
    else
    {
        size_t len = count;
```

```
        while (len--)
            *dest++ = *src++;
    }
    *dest = '\0';
    return ret;
}

int main() {
    char a[20] = "abc";
    char b[10] = "abcdef";
    my_strncat(a, b,0);
    puts(a);
    return 0;
}
```

95、模拟实现memcpy

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

void* my_memcpy(void* dest, const void* src, size_t count) {
    assert(dest && src);
    void* ret = dest;
    while (count--) {
        *(char*)dest = *(char*)src;
        ((char*)dest)++;
        ((char*)src)++;
    }
    return ret;
}

int main() {
    int arr1[10];
    int arr2[] = { 1,2,3,4,5,6,7,8,9,10 };
    my_memcpy(arr1, arr2, 10 * sizeof(int));
    for (int i = 0;i < 10;i++) {
        printf("%d ", arr1[i]);
    }
}
```

96、模拟实现memmove

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

void my_memmove(void* dest, const void* src, size_t count) {
    assert(dest && src);
    char* ret = dest;
    if (dest < src) {
        // 从前往后
        while (count--) {
            *(char*)dest = *(char*)src;
            ((char*)dest)++;
```

```
            ((char*)src)++;
        }
    }
    else {
        // 从后往前
        while (count--)
            *((char*)dest + count) = *((char*)src + count);
    }
    return ret;
}

int main() {
    int arr1[] = { 1,2,3,4,5,6,7,8,9,10 };
    my_memmove(arr1 + 3, arr1, 5 * sizeof(int));
    for (int i = 0;i < 10;i++) {
        printf("%d ", arr1[i]);
    }
}
```

97、编写判断大小端程序

写一个函数判断当前机器是大端还是小端，如果是小端返回1，如果是大端返回0.

```
#include <stdio.h>

int check_sys() {
    int a = 1; // 01 00 00 00
    return *(char*)&a;
}

int main() {
    if(check_sys())
        printf("小端机器\n");
    else
        printf("大端机器\n");
    return 0;
}
```

```
#include <stdio.h>

int check_sys() {
    union {
        int i;
        char j;
    } m;
    m.i = 1;
    return m.j;
}

int main() {
    if (check_sys())
        printf("小端机器\n");
    else
        printf("大端机器\n");
```

```
    return 0;
}
```

98、变种水仙花数

变种水仙花数 - Lily Number：把任意的数字，从中间拆分成两个数字，比如1461 可以拆分成（1和461），（14和61），（146和1),如果所有拆分后的乘积之和等于自身，则是一个Lily Number。
例如：
655 = 6 * 55 + 65 * 5
1461 = 1*461* + *146*1 + 146*1
求出 5位数中的所有 Lily Number。
输入描述：
无
输出描述：
一行，5位数中的所有 Lily Number，每两个数之间间隔一个空格。

```c
int main() {
    int i, j;
    for (i = 10000; i <= 99999; i++) {
        int n = 0;
        for (j = 10; j <= 10000; j *= 10) {
            n += (i % j) * (i / j);
        }
        if (n == i) {
            printf("%d ", i);
        }
    }
    return 0;
}
```

99、序列中删除指定数字

```c
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    int* arr = (int*)malloc(sizeof(int) * n);
    if (arr == NULL) return 0;
    int i, j;
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    int del; // 要删除的数字
    scanf("%d", &del);

    for (i = 0, j = 0; i < n; i++) {
        if (arr[i] != del) {
            arr[j++] = arr[i];
        }
    }
```

```
    for (i = 0; i < j; i++) {
        printf("%d ", arr[i]);
    }

    free(arr);
    arr = NULL;
    return 0;
}
```

100、模拟实现atoi

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


int my_atoi(const char* str) {
    // 空格，正号，负号，遇到非数字字符停止计算

    // 1. 跳过空格
    if (*str == '\0')
        return 0;
    while (*str == ' ')
        str++;
    // 2. 遇到正负号
    int flag = 1;
    if (*str == '+'){
        flag = 1;
        str++;
    }
    else if(*str == '-'){
        flag = 0;
        str++;
    }
    // 3. 进行转换
    long long ret = 0;
    while (*str >= '1' && *str <= '9') {
        ret = ret * 10 + *str - '0';
        str++;
        if (ret >= INT_MAX) {
            return INT_MAX;
        }
    }
    if (flag == 0)
        ret = -ret;
    return ret;
}
int main() {
    const char* s = "     -1211111a31";
    int ret = my_atoi(s);
    printf("%d\n", ret);
    return 0;
}
```

101、写一个宏，计算结构体中某变量相对于首地址的偏移，并给出说明

> 考察：`offsetof`宏的实现

```
#define MY_OFFSETOF(s,m) (size_t)&(((s*)0)->m)

typedef struct {
    short a;
    int b;
    double c;
}A;

int main() {
    printf("%d", MY_OFFSETOF(A, c));
    return 0;
}
```

102、写一个宏，可以将一个整数的二进制位的奇数位和偶数位交换

```
#define SWAP_BIT_INT(n) ((((n) & 0x55555555) << 1) | (((n) & 0xaaaaaaaa) >> 1))

int main() {
    int n = 13;
    int ret = SWAP_BIT_INT(n);
    printf("%d", ret);
    return 0;
}
```