



中國人民大學
RENMIN UNIVERSITY OF CHINA

期中复习专题-2

(数值计算)

余力

buaayuli@ruc.edu.cn

题目类型

- 密码类
- 素数类
- 回文类
- 集合类
- 数位类
- 进制类
- 同构类
- 模拟类
- 字符类

#161 数据加密

某个公司采用公用电话传递数据，数据是四位的整数，数据在传递过程中是加密的：每位数字都加上 5，得到的结果除以 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。请你编写程序按照上述规则加密数据。

输入格式

□□输入只有一行，包括一个 4 位数的正整数 d ($1000 \leq d \leq 9999$)，表示加密前的数据。

输出格式

□□输出只有一行，也是一个 4 位数的正整数，表示加密后的数据。

输入样例

1235

输出样例

876

【样例 1 说明】1235 每位上数字加 5 后模 10 得到的新数字是 6780，按照要求第一位第四位交换，第二位第三位交换后是 876（先导 0 不输出）。

读取每一位

#161 数据加密

数位模块

```
#include<stdio.h> ↵
```

```
int main() ↵
```

```
{ ... int n, m=0; ↵
```

```
... scanf("%d",&n); ↵
```

```
    m+=(n%10+5)%10*1000 ↵
```

```
    m+=((n/10%10)+5)%10*100 ↵
```

```
    m+=((n/100%10)+5)%10*10 ↵
```

```
    m+=((n/1000)+5)%10) ↵
```

```
... printf("%d",m); ↵
```

```
... return 0; ↵
```

```
} ↵
```

#161 数据加密

```
#include <stdio.h>+  
main()+  
{  
    int a[4],m,n,i,c;+  
    scanf("%d",&c);+  
    for(i=0;i<4;i++)+  
    {  
        a[i]=(c%10+5)%10;+  
        c=c/10;    }+  
    m=a[0];    n=a[1];+  
    a[0]=a[3];    a[1]=a[2];+  
    a[3]=m;    a[2]=n;+  
    printf("%d",a[0]+10*a[1]+100*a[2]+1000*a[3]);+  
}
```

#161 数据加密

```
int main() {
```

```
→ int i, j, n;
```

```
→ scanf("%d", &n);
```

数位模块

```
→ int x[4], y[4];
```

```
→ x[3] = n % 10;
```

```
→ x[2] = n / 10 % 10;
```

```
→ x[1] = n / 100 % 10;
```

```
→ x[0] = n / 1000 % 10;
```

```
→ for (i = 0; i <= 3; i++) {
```

```
→     → x[i] += 5;
```

```
→     → if (x[i] >= 10) x[i] = x[i] % 10;
```

```
→     → x[i] = x[i] % 10;
```

```
→ }
```

```
y[0] = x[3]; → y[1] = x[2];
```

```
y[2] = x[1]; → y[3] = x[0];
```

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

```
→ if (y[i] != 0) break;
```

```
for (j = i; j <= 3; j++)
```

```
→ printf("%d", y[j]);
```

```
return 0;
```

逐个数位输出

整数和输出

数位和

```
int main() {  
    int n, i;  
    int a[100];  
    scanf("%d", &n);  
    for (i = 0; n > 0; i++) {  
        a[i] = n % 10;  
        n /= 10;  
    }  
  
    for (int j = i - 1; j >= 0; j--)  
        printf("%d", a[j]);  
  
}
```

计算数的位数

```
for(w=1,d=0; w<=i; d++)  
    w *= 10;
```

#287 拆分数字

给出一个不多于 4 位的十进制非负整数 N，求它是几位数，并按个十百千顺序打印出各位数字。

输入格式

□□一行，只包含一个十进制非负整数 N。

输出格式

□□一行，分为两部分，首先按个十百千输出各位数字；然后输出位数。数字之间以逗号分隔。

输入样例

123

输出样例

3,2,1,3


```

#include <stdio.h> ↵
int main() ↵
{
    int n, x, count = 0; ↵
    scanf("%d", &n); ↵
    if (n == 0) printf("0,1\n"); ↵
    else {
        while (n) ↵
        {
            x = n % 10; ↵
            printf("%d,", x); ↵
            n /= 10; ↵
            count++;
        } ↵
        printf("%d\n", count);
    } ↵
    return 0; ↵
} ↵

```

数位模块

#310 同构数

计算正整数[a,b]之间的全部“同构数”之和。所谓“同构数”，是指一个正整数 n 是它平方数的尾部，则称 n 为同构数。如 6 的平方是 36，6 出现在 36 的右端，6 就是同构数。76 的平方数是 5776，76 是同构数。

【输入格式】

输入只有一行，输入两个正整数 a 和 b，中间由一个空格分隔，其中： $1 \leq a \leq b \leq 10000$ 。

【输出格式】

输出一行，一个正整数，为 a、b 之间同构数之和。

【输入样例 1】 1 100	【输入样例 2】 80 100
【输出样例 1】 113	【输出样例 2】 0

【数据规模说明】

$1 \leq a \leq b \leq 10000$ 。

76的平方数是5776，76是同构数

数位模块

76的平方数是5776，76是同构数

```
#include <stdio.h>
#include <math.h>
int main() {
    int from, to, n, m, weishu, sum=0, i;
    scanf("%d%d", &from, &to);
    for (n = from; n <= to; ++n) {
        i = n;
        weishu = 0;
        do {i = i / 10;
            weishu++;
        } while (i != 0);
        m = n * n;
        for (i = 0; i < weishu; i++)
            m = m / 10;
        for (i = 0; i < weishu; i++)
            m = m * 10;
        if (n == n * n - m) sum += n;
    }
    printf("%d", sum);
    return 0;
}
```

```
for (n = from; n <= to; ++n) {
```

```
    i = n;
```

n=76

```
    weishu = 0;
```

```
    do {i = i / 10;
```

```
        weishu++;
```

```
    } while (i != 0);
```

```
    m = n * n;
```

m=5776

```
    for (i = 0; i < weishu; i++)
```

```
        m = m / 10;
```

m=57

```
    for (i = 0; i < weishu; i++)
```

```
        m = m * 10;
```

m=5700

```
    if (n == n * n - m) sum += n;
```

```
}
```

精练

```
int main() {  
    → int from, to, i, w = 1, sum = 0;  
    → scanf("%d %d", &from, &to);  
    → for(i = from; i <= to; i++) {  
        → for(w = 1; w <= i; w *= 10);  
        → //while(w <= i) w *= 10;  
        → if((i * i - i) % w == 0)  
            → sum += i;  
    }  
    → printf("%d", sum);  
    → return 0;  
}
```

复杂

```
#include <stdio.h>  
int main() {  
    → int count = 0, same[10000], i;  
    → for(i = 1; i <= 9999; i++) {  
        → if(i >= 1 && i <= 9) {  
            → if((i * i) % 10 == i) {  
                → same[count] = i;  
                → count++; }  
            → else if(i >= 10 && i <= 99) {  
                → if((i * i) % 100 == i) {  
                    → same[count] = i;  
                    → count++; }  
                → else if(i >= 100 && i <= 999) {  
                    → if((i * i) % 1000 == i) {  
                        → same[count] = i;  
                        → count++; }  
                    → else if(i >= 1000 && i <= 9999) {  
                        → if((i * i) % 10000 == i) {  
                            → same[count] = i;  
                            → count++; }  
                        → else if(i == 9999) { count++; }  
                    }  
                }  
            }  
            → //以下开始输出;  
            → int a, b, sum;  
            → scanf("%d %d", &a, &b);  
            → for(i = 0, sum = 0; i++) {  
                → if(same[i] >= a && same[i] <= b) {  
                    → sum = sum + same[i];  
                    → } else if(same[i] > b) → break;  
                }  
            → printf("%d", sum);  
            → return 0;  
        }  
    }
```

#287 拆分数字

给出一个不多于 4 位的十进制非负整数 N，求它是几位数，并按个十百千顺序打印出各位数字。

输入格式↵

□□一行，只包含一个十进制非负整数 N。↵

输出格式↵

□□一行，分为两部分，首先按个十百千输出各位数字；然后输出位数。数字之间以逗号分隔。↵

输入样例↵

123↵

输出样例↵

3,2,1,3↵

#304 整数计数

编写一个程序计算整数区间 $[a, b]$ 内，其个位数是 n ，且能被 k 整除的 m 位正整数共有多少个。

【输入格式】

输入只有一行，输入 5 个整数 a 、 b 、 n 、 k 、 m ，空格分隔，其中： $1 \leq a \leq b \leq 1,000,000$ ，且 $0 \leq n, k, m \leq 9$ 。

【输出格式】

输出一行，为符合要求的整数个数。

【样例输入 1】

1019 1 2 2

【样例输出 1】

0

【样例输入 2】

1050 4 2 2

【样例输出 2】

4

#194 进制转换

编程实现，将十进制整数 d ，转换为 h 进制数 a ，并按 h 进制规则输出 a 。

输入格式

□□一行，两个十进制整数 d 和 h ，之间由一个空格分隔，其中： $0 \leq d \leq 1,000,000,000$ ； $2 \leq h \leq 16$ 。

输出格式

□□一个 h 进制的整数。十进制以上数字，用大写字母 A、B、C、D、E、F 编码。

输入样例

123 16

输出样例

7B

while($N > 0$)

{ S1: 求余数 $N \% h$;

S2: 保存余数;

S3: 准备新 N , $N = N / h$; }

进制类题目

进制模块

```
int main() ↵
{
    int N, h, mod, index=0; ↵
    char result[100]={'\0'}; ↵
    scanf("%d%d",&N,&h); ↵
    while(N>0) ↵
    {
        mod=N%h; ↵
        if(mod>=0&&mod<=9) result[index++] = mod+'0'; ↵
        else result[index++] = mod-10+'A'; ↵
        N=N/h;
    } ↵
    for(int i=index-1; i>=0; i--) ↵
        printf("%c", result[i]); ↵
    return 0; ↵
} ↵
```


#491 调价

某条街道有 m 家商店依次排开，他们卖同一款商品，第 1 天他们各自定了一个价格，随后每天调一次价，新价格为前一天自己和相邻商店价格的平均值（去尾法取整），注意头尾两家商店只有一个相邻商店。现在需要计算，第几天开始所有商店价格一致。

【输入格式】

两行整数，第一行为商店数量 m ，第二行依次为每家商店第一天的定价。

【输出格式】

一个整数，所有商店价格一致是第几天。

【输入样例】

3

10 15 30

【输出样例】

7

【数据描述】

$m < 100$ ，定价 < 10000

迭代模拟类题目

New[] Old[]

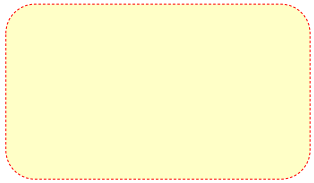
```
while(1)
{ 是否需要迭代;
  If 不需要就结束
  迭代操作;
  num++;
}
```

```

int main() {
int ShopN, i, num=1, SUCCESS, New[110]={0}, Old[110]={0};
scanf("%d", &ShopN);
for (int i = 1; i<=ShopN; i++) scanf("%d", &New[i]);

```

```

while (1) {
    
}
printf("%d", num);
}

```

```

while (1) {
    → for (SUCCESS=1, i=1; i<=ShopN; i++)
    ① → if (New[1]!=New[i]) SUCCESS=0; // 优化: 加 break;
    → if (SUCCESS==1) break;
    → for (int i=0; i<=ShopN+1; i++)
    ② → Old[i]=New[i];
    → for (int i=1; i<=ShopN; i++)
    ③ → if (i==1 || i==ShopN)
    → → New[i]=(Old[i-1]+Old[i]+Old[i+1])/2;
    → else
    → → New[i]=(Old[i-1]+Old[i]+Old[i+1])/3;
    → num++;
}

```

#461 回文数等式

回文数 x 是指正读和反读都一样的正整数，如 11, 121, 1221。输入一个 k ($1 \leq k \leq 1000$)，打印所有不超过 k 的数 i ($1 \leq i \leq k$) 的平方是回文数 x 的等式，即 $i*i=x$ 。输出每行一个等式。

输入格式

一行，只有一个整数 k

输出格式

$i*i=x$

输入样例

30

输出样例

$1*1=1$

$2*2=4$

$3*3=9$

$11*11=121$

$22*22=484$

$26*26=676$

“回文类”
对称循环类

```

int reverse(int x) {
    int y = 0;
    while (x) {
        y = y * 10 + x % 10;
        x /= 10;
    }
    return y;
}

```

```

int Huiwen(int x) {
    int temp = x, y = 0;
    while (x) {
        y = y * 10 + x % 10;
        x /= 10;
    }
    return y == temp;
}

```

```

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

```

#493 回文素数位和

如果一个数从左边读和从右边读都是同一个数，就称为回文数，既是素数又是回文数的数，称为回文素数，例如 500 到 1000 之间的回文素数有 6 个：727、757、787、797、919、929，其中数位和最大的是 797（数位和为 $7+9+7=23$ ）。

编程求 $[m, n]$ 区间数位和最大的回文素数。

【输入格式】

输入一行，两个正整数 m 和 n ，用空格隔开。

【输出格式】

输出一行，包含 2 个整数，依次是区间 $[m, n]$ 中数位和最大的回文素数及其数位和，2 个整数之间一个用空格隔开。

如果给定的 $[m, n]$ 区间没有素数回文数，则输出 0 0。

【输入样例 1】 【输入样例 2】

500 1000 20 50

【输出样例 1】 【输出样例 2】

797 23 0 0

【数据描述】

$2 < m < n < 100000$

```
int su(int x){ //判断素数 1-yes 0-no
```

```
→ for(int i=2; i*i<=x; i++)
```

```
→     if(x%i==0)
```

```
→         return 0;
```

```
→ return 1;
```

```
} 
```

```
int digitsum(int x){
```

```
→ int sum=0;
```

```
→ while(x!=0){
```

```
→     sum+=x%10;
```

```
→     x/=10; }
```

```
→ return sum;
```

```
} 
```

```
int hui(int x){ //判断回文 1-yes 0-no
```

```
→ char a[20];
```

```
→ int num=0;
```

```
→ while(x!=0){
```

```
→     a[num++]=x%10;
```

```
→     x/=10;
```

```
→     num++; }
```

```
→ for(int i=1; 2*i<=num; i++)
```

```
→     if(a[i]!=a[num--i+1])
```

```
→         return 0;
```

```
→ return 1;
```

```
} 
```

判断素数

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

#299 质因数分解

给定一个整数 n ，请给出它的质因数分解。如 28 可以如下分解： $28 = 2^2 \times 7$ 。
请输出的每个质因数以及该质因数的指数。

输入格式

一行，包含一个整数 n ($2 \leq n \leq 10000$)。

输出格式

若干行，每行输出整数 n 的一个质因数，以及该质因数的指数，之间用一个冒号：隔开，且按质因数从小到大顺序输出。

输入样例

28

输出样例

2:2

7:1

数值计算类


```
int main() {  
    → int n, count;  
    → scanf("%d", &n);  
    →  
    →  
    → return 0;  
}
```

书写风格

```
for (int i = 2; i <= n; i++) {  
    → count = 0;  
    → while (1) {  
        → if (n % i == 0) {  
            → {count++; n /= i;}  
        → else {  
            → if (count != 0) printf("%d:%d\n", i, count);  
            → break; }  
        → }  
    → }  
}
```

```
for (int i = 2; i <= n; i++) {  
    → for (count = 0; n % i == 0; n /= i) {  
        → count++;  
        → if (count != 0) printf("%d:%d\n", i, count);  
    }  
}
```

#185 迭代法求平方根

给定一个正整数 a ，用迭代法求出 a 的平方根并输出迭代次数。迭代公式如下：

$$\dots\dots\dots X[n+1] = 1/2(X[n] + a/X[n])$$

要求前后两次求出的 x 的差绝对值小于 10^{-5} ，迭代初值取 $a/2$ 。

输入格式

□□ 输入数据一行，包含一个整数 a ($2 \leq a \leq 2^{31}-1$)。

输出格式

2 行。□□

第一行一个数 x ，表示 a 的平方根，保留 6 位小数。

第二行一个数 t ，表示迭代次数。

输入样例

2

输出样例

1.414214

4

“循环模拟”类题

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

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

```
int main()
```

```
{  int a, t = 0;
```

```
    double x1, x2;
```

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

```
    x1 = a;  x2 = a/2.0;
```

```
    while ( fabs(x2 - x1) >= 1e-5 )
```

```
        { x1 = x2;
```

```
          x2 = (x1+ a/x1)/2;
```

```
          t++; }
```

```
    printf("%.6f\n%d\n", x2,t);
```

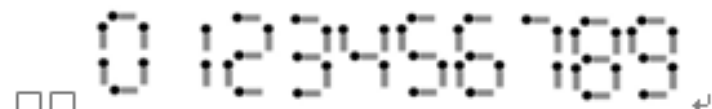
```
    return 0;
```

```
}
```

$$X[n+1] = 1/2(X[n] + a/X[n])$$

#131 火柴棒等式

给你 n 根火柴棒，你可以拼出多少个形如 “ $A + B = C$ ” 的等式？等式中的 A 、 B 、 C 是用火柴棒拼出的整数（若该数非零，则最高位不能是 0）。用火柴棒拼数字 0-9 的拼法如图所示：



注意：

1. 加号与等号各自需要 2 根火柴棒

2. 如果 $A \neq B$ ，则 $A + B = C$ 与 $B + A = C$ 视为不同的等式（ $A, B, C \geq 0$ ）

3. n 根火柴棒必须全部用上

输入格式：共一行，有一个整数 n ($n \leq 24$)。

输出格式：共一行，有一个整数，表示能拼成的不同等式的数目。

输入样例

18

输出样例

9

$$0 + 4 = 4$$

$$0 + 11 = 11$$

$$1 + 10 = 11$$

$$2 + 2 = 4$$

$$2 + 7 = 9$$

$$4 + 0 = 4$$

$$7 + 2 = 9$$

$$10 + 1 = 11$$

$$11 + 0 = 11$$

```

for (int i = 0; i <= 1000; i++) {
    → for (int j = 0; j <= 1000; j++) {
        → if (a[i] + a[j] + a[i + j] + 4 == n) {
            → → → total++; //方案数加 1
        }
    }
}

```

```

for (int i = 1; i <= 2000; i++) {

```

a[i]=?

```

}

```

难度分解法

两步法

逆向思维

#484 重复元素

统计数列有重复的元素及其个数。已知有不超过短整型数值范围的 n 个数 ($n \leq 1000$)，请查找统计有重复的数列项，按行从小到大输出有重复的项及其重复次数。

【输入格式】：两行，第一行表示数列元素的个数 n ；第二行为该数列的元素，以空格分隔。

【输出格式】：若干行，每行两个数，冒号分隔，分别为有重复的数项值及其重复的次数。输出顺序由数列项的值确定。若数列中没有重复项，则输出 NO。

【样例输入 1】

10

36 30 68 38 2 30 36 30 68 30

【样例输出 1】

30:4

36:2

68:2

#111 Coconuts

有 n 个 coconuts (椰子), m 个人。第一天, 第一个人将这些 coconuts 平均分成 m 份, 恰好多出 1 个, 他把多出的那个 coconuts 送给了小猴子, 并把属于自己的那份拿走了。第二天, 第二个人同样把剩下的 coconuts 平均分成 m 份, 又恰好多出 1 个, 他把多出的那个 coconuts 送给了小猴子, 并把属于自己的那份拿走了.....如此重复了 m 天。第 $m+1$ 天, 这 m 个人一起来了, 他们将剩下的 coconuts 恰好平均分成 m 份。这一次, 就没有剩余的 coconuts 了。

输入格式

□□ 输入数据包含多个测试组, 组数 ≤ 20 。

□□ 每个测试组仅包含一个整数 n ($1 \leq n \leq 2000000000$), 表示 coconuts 的个数。

□□ 最后一个测试组仅包含一个整数 0, 不需要对其进行运算。

输出格式

□□ 对于每个测试组, 输出一行, 即最大的满足要求的人数 m 。如果没有这样的 m 输出 "no-solution"

输入样例

25

30

3121

0

输出样例

3

no-solution

5

```
int main() {
```

```
→ int a[20], num = 0, i, j, m, times, n, flag;
```

```
→ for (i = 0; ; i++)
```

```
{ scanf("%d", &a[i]); num++;
```

```
if (a[i] == 0) → break; }
```

```
→ for (i = 0; i < num - 1; i++) {
```

```
→ for (flag = 0, m = sqrt(a[i]); m >= 2; m--) {
```

```
→ for (n = a[i], times = 0, j = 0; j < m; j++)
```

```
→ if ((n - 1) % m == 0) {
```

①

```
→ n = (n - 1) / m * (m - 1); times++; }
```

```
→ else break;
```

```
→ if (times == m && n % m == 0) {
```

②

```
→ printf("%d\n", m);
```

```
→ flag++; → break; → } - - - }
```

```
if (flag == 0) → printf("no solution\n"); }
```

```
return 0;
```

```
} 
```

模拟类题

判断素数

函数风格

```
int main() {  
    → int a[20], m, i, flag;  
    → for (i = 0; i < 20; i++) {  
        → scanf("%d", &a[i]);  
        → if (a[i] == 0) → break;  
        → flag = 0;  
        → for (m = sqrt(a[i]); m > 1; m--) {  
            → if (!IsOK(a[i], m)) {  
                → { printf("%d\n", m); flag = 1; break; }  
            → if (flag == 0) → printf("no solution\n");  
        }  
    }  
    → return 0;  
}
```

```
int IsOK(int n, int m) {  
    → for (int i = 1; i <= m + 1; i++)  
        → if (i == m + 1)  
            → if (n % m == 0) → return 1;  
            → else → return 0;  
        → else  
            → if ((n - 1) % m != 0) → return 0;  
            → else n -= (n - 1) / m + 1;  
    }  
}
```

#290 购物车共同性

电商平台在购物狂欢节期间，从某类顾客中随机抽取 n 个顾客($n > 1$)，调查他们购物车中预选保存的商品类别，以掌握顾客购物偏好的共同性，得到最受欢迎的商品，为拓展市场销量服务。购物车中商品会被分类，假定种类是以整数进行编号，此问题为在 n 个集合中寻找交集的问题。

输入格式

□□ 第一行一个整数 n ，表示选取的顾客数。此后有 n 行，分别表示每位顾客所选商品构成，其中第一个数表示该顾客购物车中的不同商品数，其后为空格分隔的整数。调查抽取的顾客数 n 不超过 20，每个顾客所选的商品种类不超过 100。

输出格式

□□ 若不存在共同偏好，无交集，输出 NO；若有，把共同的商品编号从小到大输出，空格分隔。

输入样例

3

5 1 2 3 4 5

5 2 4 6 8 10

8 1 2 3 4 5 6 8 10

输出样例

2 4

数组交叉

集合交叉

```
int main() {
    int GoodN1, GoodN, UserN, i, j, k, GoodID, a[100], b[100] = {0}, num, c[100];
    scanf("%d", &UserN);
```

```
    scanf("%d", &GoodN1);
```

1 for (j = 0; j < GoodN1; j++) // 输入第 1 个的商品编号

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

```
    for (i = 0; i < UserN - 1; i++) { // 输入其他 (UserN - 1) 个人的商品编号
```

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

```
        for (j = 0; j < GoodN; j++) {
```

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

```
            for (k = 0; k < GoodN1; k++)
```

```
                if (GoodID == a[k]) b[k]++;
```

```
        }
```

A

a1	a2	...	aN
ID	ID	...	ID

X

B

b1	b2	...	bM
----	----	-----	----

```
    for (num = 0, j = 0; j < GoodN1; j++)
        if (b[j] == UserN - 1) c[num++] = a[j];
```

```
for (i = 0; i < num - 1; i++)  
    for (j = 0; j < num - i - 1; j++)  
        if (c[j] > c[j + 1]) {  
            int temp = c[j]; c[j] = c[j + 1]; c[j + 1] = temp; }  
    }
```

③

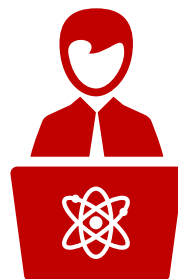
```
if (num == 0) printf("NO");  
else  
    for (j = 0; j < num; j++)  
        printf("%d ", c[j]);  
return 0;
```

④

```
}  
}
```



中國人民大學
RENMIN UNIVERSITY OF CHINA



谢谢大家！

