

# 专题（综合应用类）：中文数字

## 【应用背景】中文数字

阿拉伯数字转换为中文数字，如整数 123 或数字串“123”，转换为对应的字符串“一百二十三”。程序编写时，输入的阿拉伯数字，可以是一般整数或超长位整数字符串，转换为中文数字串时，可以使用常规的“零一二三四五六七八九十百千万亿”，也可以尝试使用中国古代大数中的“涧正载极”等计数法。

## 【各问摘要】

- 第1问：基础问答，单项选择题，理解“数穷则变”
- 第2问：程序改错，cnNum02.c，2位整数转中文数字，函数调用
- 第3问：规范优化，cnNum03.c，4位整数转中文数字，2个函数调用
- 第4问：程序集成，cnNum04.c，10位无符号数转中文数字，更多函数调用
- 第5问：功能提升，cnNum05.c，500位数字串，设计递归函数
- 第6问：综合应用，cnNum06.c，超长数字串，古代下数法中数法上数法

## 【第1问，基础问答，具体答题在考试客户端中单项选择题部分】

### 1. 单项选择题

中文汉字中有“万亿兆京垓秭穰沟涧正载极”等计数单位，为尽可能表示更大的数字，古代“上数法”定义各单位的换算关系为“数穷则变”，即 1 万万=1 亿，1 亿亿=1 兆，1 兆兆=1 京，1 京京=1 垣，以此类推。那么，按“上数法”，1 垣等于（\_\_\_\_\_）

- A. 1 亿
- B. 1 亿亿
- C. 1 亿亿亿亿
- D. 1 亿亿亿亿亿亿亿

## 【第2问，程序调试，cnNum02.c，2位整数对应的中文数字】

**调试要求：**打开文件“C:\KS\cnNum02Err.c”，该程序包含了3个错误，找出错误点，在错误行后加上“//\_”注释，调试修改后保存为C:\KS\cnNum02.c。

**程序功能：**输入1个一或两位整数（取值0至99），输出对应的中文数字串，输入超界时输出“无效”。如输入“30”，则输出“三十”；输入49，则输出“四十九”。

**运行示例及说明**（第1行为输入，第2行为输出）：



**注：**为方便程序设计，当输入 12 时，可输出“十二”或“一十二”，本专题不做限定。

**含错误的源程序：**（包含3个错误点，函数prtChnNum1不含错误）

```
//cnNum02, 输入一或两位整数, 输出对应的中文数字串
#include <stdio.h>
void prtChnNum1(int n)
{ //输出数字 n 对应的中文汉字, 整数 n 取值: 0~9, 10
    switch (n)
    {
        case 0: printf("零"); break;
        case 1: printf("一"); break;
        case 2: printf("二"); break;
        case 3: printf("三"); break;
        case 4: printf("四"); break;
        case 5: printf("五"); break;
```

```

        case 6: printf("六"); break;
        case 7: printf("七"); break;
        case 8: printf("八"); break;
        case 9: printf("九"); break;
        case 10: printf("十"); break;
        default: printf("×"); break;
    }
} //本行及以上代码不含错误
int main(void)
{ //主函数
    int n, a1, a2;
    scanf("%d", &n); //n为输入数
    a1 = n%10; //个位数
    a2 = n/10 % 10; //十位数
    if (n<0 && n>=100)
        printf("无效");
    else if (n==0)
        prtChnNum1(0);
    else
    {
        if (a2>=1)
        {
            prtChnNum1(a2); //输出十位数
            prtChnNum1(10); //输出“十”
        }
        if (a1>=0)
            prtChnNum1(a1); //输出个位数
    }
    printf("\n"); //程序结束前“换行”
    return 0;
}

```

### 【第3问，程序优化，cnNum03.c，4位整数对应的中文数字】

**程序功能：**输入1个四位以内整数（取值0至9999），输出对应的中文数字串，输入超界时输出“无效”。如输入“1234”，则输出“一千二百三十四”；输入“3012”，则输出“三千零一十二”；输入“9001”，则输出“九千零一”；输入6000，则输出“六千”。编写程序，保存为C:\KS\cnNum03.c。

**运行示例**（第1行为输入，第2行为输出）：

1234 一千二百三十四 请按任意键继续... □	6000 六千 请按任意键继续... □	708 七百零八 请按任意键继续... □
---------------------------------	----------------------------	-----------------------------

**设计建议：**

(1) 修改第2问的函数prtChnNum1()，使该函数增加输出“百”、“千”等中文。

```
void prtChnNum1(int n); //输出数字n对应的中文汉字，n取值0~9, 10, 100, 1000
```

(2) 设计函数prtChnNum4()，输出1至9999的中文数字，该函数需调用prtChnNum1函数以实现单个汉字输出。

```
void prtChnNum4(int n); //输出4位数字n对应的中文数字串，n取1~9999
```

(3) 修改主函数main()，主函数调用prtChnNum4()等函数。

### 【第4问，程序集成，cnNum04.c，10位无符号整数对应的中文数字】

**程序功能：**输入1个无符号整数（不超过10位），输入整数后不再检查数据是否有效（是否

超界), 输出对应的中文数字串。编写程序, 保存为 C:\KS\cnNum04.c。

运行示例及说明 (图中第 2 行为输入):

输入不超过10位的非负整数（≤4294967295）： 1234567890 1234567890: 一十二亿三千四百五十六万七千八百九十 请按任意键继续. . .	输入不超过10位的非负整数（≤4294967295）： 1000023 1000023: 一百万零二十三 请按任意键继续. . .
---	--

注: unsigned 数的最大值为 4294967295, 测试时输入整数不要超过该数值。

设计建议：

- (1) 修改第3问的函数prtChnNum1(), 使该函数能输出“万”、“亿”等中文。

```
void prtChnNum1(int n); //输出数字n对应的中文汉字
```

- (2) 设计函数prtChnNum8(), 输出1至99999999的中文数字, 该函数需调用第3问中的函数prtChnNum4()以实现4位数字输出。

```
void prtChnNum8(int n); //输出8位数字n对应的中文数字, n取1-99999999
```

- (3) 设计函数prtChnNum16(), 输出1至“亿亿”的中文数字, 该函数需调用prtChnNum8()函数以实现8位数字输出。

```
void prtChnNum16(unsigned n); //输出无符号整数 n 对应的中文数字, n>0
```

- (4) 修改主函数 main(), 主函数不需要判断 n 的有效范围, 调用 prtChnNum16() 等函数。

```
int main(void)
```

```
{ //主函数
    unsigned n = -1;
    printf("输入不超过 10 位的非负整数 (≤%u): \n", n);
    scanf("%u", &n);
    printf("%u: ", n);
    .....
    printf("\n");           //程序结束前“换行”
    return 0;
}
```

【第5问，功能提升，cnNum05.c，超长位整数对应的中文数字】

**程序功能:** 输入 1 个不超过 50 位的无符号整数串, 输出对应的中文数字串。编写程序, 保存为 C:\KS\cnNum05.c。

**运行示例及说明** (图中第 2 行为输入):

输入1个500位以内正整数：

一十二亿亿亿亿亿三千四百五十六万七千八百九十一亿亿亿亿  
一千三百三十四万五千六百七十八亿亿亿九千零一十二万三千  
四百五十六亿亿亿七千八百九十万一千二百三十四亿亿五千六百  
七十八万九千零一十二亿三千四百五十六万七千八百九十  
请按任意键继续... -

输入 1: 12345678901234567890123456789012345678901234567890

输入 2: 120000000000000000000000000003

**注:** 为方便程序设计,本专题在“万”与“千”之间,或者“亿”与“千万”之间,不规定是否加“零”,如输入102000时,可输出“一十万二千”,也可输出“十万零二千”。

#### 设计建议：

- (1) 输入为字符串, 为方便程序设计, 假设字符串长度 500 以内, 输入串仅有数字字符组成, 数字串不以'0'开始, 且数值 $\geq 1$ 。

- (2) 设计函数 `prtChnNum()`，该函数将长度为 `len` 的数字字符串 `s`，转换为中文数字并输出，该函数使用递归方式实现。

```
void prtChnNum(char *s, int len);
```

(3) 设计主函数 main(), 主函数调用 prtChnNum() 函数。

```
int main(void)
{ //主函数
    char s[500];
    printf("输入1个500位以内正整数: \n");
    scanf("%s", s);
    prtChnNum(s, strlen(s));
    printf("\n");
    return 0;
}
```

【第 6 问，综合应用，cnNum06.c，中国古代数字】

**背景说明：**中国古代有“万亿兆京垓秭穰沟涧正载极”等用于表示大数的计数单位，计数关系有“下数法”、“中数法”和“上数法”等不同方式：

- 下数法：十变之，如 $10\text{万}=1\text{亿}$ ， $10\text{亿}=1\text{兆}$ ， $10\text{兆}=1\text{京}$ 等。
  - 中数法：万万变之，如 $1\text{万万}=1\text{亿}$ ， $1\text{万亿}=1\text{兆}$ ， $1\text{万兆}=1\text{京}$ 等。
  - 上数法：数穷则变，如 $1\text{万万}=1\text{亿}$ ， $1\text{亿亿}=1\text{兆}$ ， $1\text{兆兆}=1\text{京}$ 等。

同一个汉字在不同计数法下有不同含义，如“兆”字，下数法时  $1\text{兆}=1\text{百万}$ ，中数法时  $1\text{兆}=1\text{万亿}$ ，上数法时  $1\text{兆}=1\text{亿亿}$ 。

**程序功能:** 输入 1 个不超过 10000 位的整数串, 消除输入数的前导 0 和正负符号, 统计输入数字串的长度, 再分别使用第 5 问的方法(称为现代法), 以及古代下数法、中数法、上数法等方式, 输出对应的中文数字串。编写程序, 保存为 C:\KS\cnNum06.c。

**运行示例** (图中第 2 行为输入):

输入1个5000位以内正整数：  
1234567890123456789012345678901234567890  
1234567890123456789012345678901234567890 (长度=40)  
【现代法】一千二百三十四万五千六百七十八亿亿亿九千零一十二万三千四百五十六亿亿七千八百九十万一千二百三十四亿亿五千六百七十八万九千零一十二亿三千四百五十六万七千八百九十  
【下数法】一秭二垓三京四兆五亿六万七千八百九十极极一载二正三涧四沟五穰六秭七垓八京九兆零一万二千三百四十五极六载七正八涧九沟零一秭二垓三京四兆五亿六万七千八百九十  
【中数法】一千二百三十四涧五千六百七十八沟九千零一十二穰三千四百五十六秭七千八百九十垓一千二百三十四京五千六百七十八兆九千零一十二亿三千四百五六十万七千八百九十  
【上数法】一千二百三十四万五千六百七十八京九千零一十二万三千四百五十六亿七千八百九十万一千二百三十四兆五千六百七十八万九千零一十二亿三千四百五六十万七千八百九十  
请按任意键继续... .

输入 1: 1234567890123456789012345678901234567890

**设计建议:**

(1) 输入为数字字符串, 假设字符串长度不超过 10000 位, 为提高程序可靠性, 输入串需要消除正负符号及前导 0, 数字串中后续的非数字字符自动视为数字串结束。

(2) 在第 5 问 `prtChnNum()` 的基础上, 增加参数以识别不同的计数法, 重新设计递归算法, 使用单个函数取代原有的 `prtChnNum1/4/8/16` 等多个函数。

(3) 修改主函数 main(), 对输入字符串进行预处理, 再通过不同参数多次调用 prtChnNum() 函数并输出, 以得到不同中文数字结果。

【提间拓展：cpNum07.c，尝试更多挑战（不计分数）】

- (1) 先将阿拉伯数字串转换为中文数字串，再输出，不是直接输出结果。
- (2) 中文数字串逆向转换为阿拉伯数字串，转换时按不同计数方法产生不同解析。
- (3) 中文数字串用人民币大写金额，如“零壹贰叁肆伍陆柒捌玖拾佰仟万亿整元角分”等。