中国人民大学
RENMIN UNIVERSITY OF CHINA

期中复习专题-3

（字符串）

余力

buaayuli@ruc.edu.cn

字符串结束标志

- 为了测定字符串的实际长度，C语言规定了字符串结束标志'\0'
- '\0'代表ASCII码为0的字符，是一个''空操作符''，即它什么也不做

```
char c[]={"I  am  happy"};
```

可写成

```
char c[]="I  am  happy";
```

相当于

```
char c[11]={"I  am  happy"};
```

字符串结束标志

```
char c[10]={"China"};
char c[10]={'C',  'h',};
```

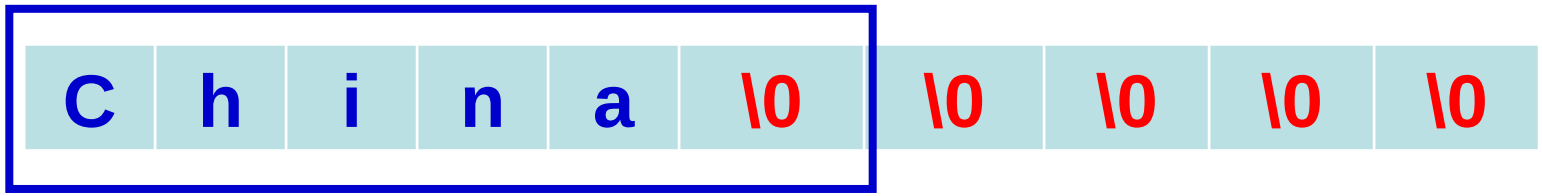
可写成

```
char c[10]="China";
```

从c[5]开始，元素值均为\0

只显示

```
printf("%s",c+3);
```



字符串函数

1. puts函数----输出字符串的函数

- 其一般形式为：　puts (字符数组)
- 作用是将一个字符串输出到终端

```
char str[20]="China";
puts(str+1);
```

2. gets函数----输入字符串的函数

- 其一般形式为：gets(字符数组)
- 作用是输入一个字符串到字符数组

```
char str[20];
gets(str);
```

Computer↙

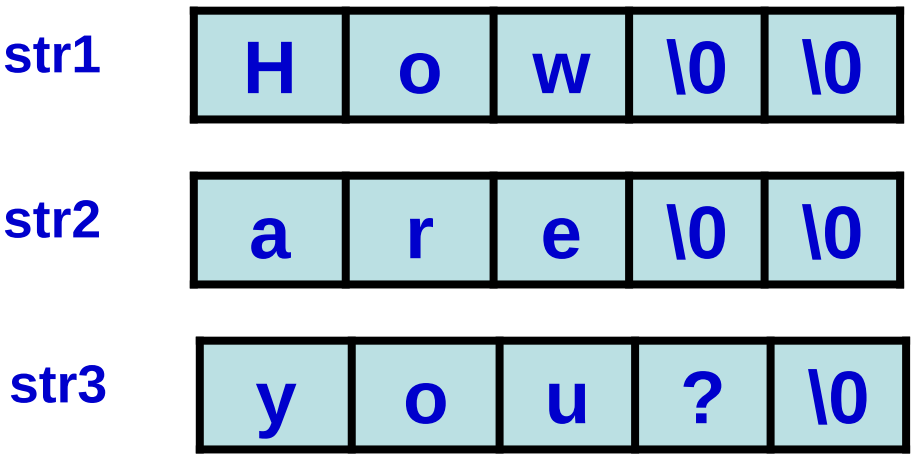
字符数串的输入输出

```
char c[6];
scanf("%s",&c[0]);   China↙
```

系统自动在China后面加一个'\0'

```
char str1[5],str2[5],str3[5];
scanf("%s%s%s",str1,str2,str3);
```

How are you? ↙




```
输入样例
5 //以整数形式输入
4662723075032711 56 //以字符串形式输入
2158564722070979 78 //以字符串形式输入
2348045804010783 65 //以字符串形式输入
4044757277000349 80 //以字符串形式输入
7103514088030931 65 //以字符串形式输入
*/
int i,n;
char a[1000][20];
scanf("%d",&n);
for(i=0;i<n;i++)
    {
        scanf("%s",a[i]);//正常
        ...
    }
```

用gets(a[i])出错

```c
if (0) {//正常
    freopen("D:/C_Programing++++++/+++C_Programing[2023-09]/友学网测试点/#135_身份证排序/data10.in", "r", stdin);//
    int i, n;
    char a[1000][20];
    scanf("%d", &n);
    for (i = 0; i < n; i++) {
        scanf("%s", a[i]);
        puts(a[i]);
    }
}
```

```
5
46627230750327115 6
21585647220709797 8
23480458040107836 5
40447572770003498 0
71035140880309316 5
```

```
46627230750327115 6
21585647220709797 8
23480458040107836 5
40447572770003498 0


--------------------------------

Process exited after 0.2004 se
请按任意键继续. . .
```

注意scanf与gets的混合使用

```c
if (1) {//
    freopen("D:/C_Programing++++++/+++C_Programing[2023-09]/友学网测试点/#135_身份证排序/data10.in", "r", stdin);
    int i, n;
    char a[1000][20];
    scanf("%d", &n);
    getchar();//
    for (i = 0; i < n; i++) {
        gets(a[i]);
        puts(a[i]);
    }
}
```

```
46627230750327115 6
21585647220709797 8
23480458040107836 5
40447572770003498 0
71035140880309316 5


--------------------------------

Process exited after 0.2057
```

```c
//输入 "345china"  或 "345<回车>china"  都正确
int a;
char b[100];
scanf("%d%s", &a, b);
printf("a=%d b=%s", a, b);
```

```c
//输入 "345china"  或 "345<回车>china"  都正确
int a;
char b[100];
scanf("%d%s", &a, b);
printf("a=%d b=%s", a, b);
```

```c
//输入 "345china" OK   但 "345<回车>china" 错误，gets获得空串
int a;
char b[100];
scanf("%d", &a);
gets(b);
printf("a=%d b=%s", a, b);
```

```c
//输入 "1china" OK   但 "1<回车>china" 错误，gets获得空串
char a;
char b[100];
scanf("%c", &a);
gets(b);
printf("a=%c b=%s", a, b);
```

```c
//输入china<回车>  可以有空格
char s[1000];
gets(s);
puts(s);
```

友学网上字符串输入

尽量不要用这样

```
int main() {
    int i = 0, j = 0;
    char c, s[1000];
    while ((c = getchar()) != '\n') {
        s[i] = c; i++;}
    for (j = 0; j <= (i / 2); j++)
        if (s[j] == s[i - 1 - j]) continue;
        else { j = -1; break; }
    if (j != -1) printf("Yes");
    else printf("No");
    return 0;
}
```

#742112	#110 回文字符串	Runtime Error	0	79 ms

下边这个缩略输入输出，老师同意可以拉开看了^_^ (但考试时候不给看)

测试点 #0	2	得分: 0
测试点 #1	2	得分: 0
测试点 #2	2	得分: 0
测试点 #3	2	得分: 0
测试点 #4	2	得分: 0
测试点 #5	2	得分: 0
测试点 #6	2	得分: 0
测试点 #7	2	得分: 0
测试点 #8	2	得分: 0
测试点 #9	2	得分: 0

上述输入应该修改为下面

```
    gets(s);
    i = strlen(s);
```

字符串输入尽量用gets()

字符串函数

3. strcat函数----字符串连接函数

- 其一般形式为：strcat(字符数组1，字符数组2)
- 其作用是把两个字符串连接起来，把字符串2接到字符串1的后面，结果放在字符数组1中

```
char str1[30]="People";
 char str2[]="China";
 printf("%s", strcat(str1,str2));
  输出：PeopleChina
strcat(str1,str2);
printf("%s", str1);
```

字符串函数

4. strcpy和strncpy函数-字符串复制

- strcpy一般形式为：strcpy(字符数组1, 字符串2)

  作用是将字符串2复制到字符数组1中去

- strncpy将字符串2中前n个字符复制到字符数组1中去

  strncpy(字符数组1, 字符串2, n)

char str1[10],str2[]=" China" ;

str1=" China" ; 错误

str1=str2; 错误

char str1[10],str2[]="China";

strcpy(str1,str2);

```
#include<stdio.h>
#include<string.h>
int main(){
char name[]={"Chinanet"},destin[20]={};
strncpy(destin,name,3);
printf("%s\n",destin);
}
```

str1	C	h	i	n	a	\0	\0	\0	\0	\0

小心使用strncpy

```
char dest[20] = "Hello\0Hi";

char src[6] = "World";

strncpy(dest, src, 2);

printf("将src的字符串赋值到dest:%s\n", dest);

//输出结果：将src的字符串赋值到dest: Wollo
```

字符串函数

5. strcmp函数----字符串比较函数

- 其一般形式为

  strcmp(字符串1, 字符串2)

- 比较字符串1和字符串2大小
- strcmp(str1,str2);
- strcmp("China", "Korea");
- strcmp(str1, "Beijing");

”A”<”B”　　”a”>”A”

”computer”>”compare”

”these”>”that”　”1A”>”$20”

”CHINA”>”CANADA”

”DOG”<”cat”

”Tsinghua”>”TSINGHUA”

if(str1>str2)　printf("yes");　错误

if(**strcmp**(str1,str2)>0)

　　printf("yes");　　正确

比较s1和s2字符串的前n个字符

strncmp(字符串1，字符串2, n)

字符串函数

6. strlen函数----测字符串长度的函数

- 其一般形式为：

  strlen (字符数组)

- 它是测试字符串长度的函数
- 函数的值为字符串中的实际长度

```
char str[10]= "China”；
 printf("%d", strlen(str));
```

- 也可以直接测试字符串常量的长度

  strlen("China");

字符串函数

7. strlwr函数----转换为小写的函数

- 其一般形式为　　strlwr (字符串)
- 函数的作用是将字符串中大写字母换成小写字母

8. strupr函数----转换为大写的函数

- 其一般形式为　　strupr (字符串)
- 函数的作用是将字符串中小写字母换成大写字母

```
char a[] = "ABCabc";
puts(strupr(a));
puts(strlwr(a));
```

字符串到整数、浮点数

```
char s[100] = "123";
double f;
f = atof(s);
printf("%lf\n", f);

char s1[100] = "123";
int n;
n = atoi(s1);
printf("%d\n", n);

char s2[100] = "123";
long long int m;
m = atol(s1);
printf("%lld\n", m);
```

atoi(p) 转换到 int 整型

atof(p) 串转换到 double 符点数

atol(p) 串转换到 long 整型

```
#include <stdio.h>
#include <stdlib.h>
int main() {
  char a[] = "-100";
  char b[] = "123";
  int c;
  c = atoi(a) + atoi(b);
  printf("c=%d\n", c);
  return 0;
}
```

#160 字符串移位包含

给定两个字符串 s1 和 s2，要求判定 s2 是否能够被通过 s1 作**向右循环移位**（rotate）得到的字符串包含。例如，给定 s1 为 AABCD 和 s2 为 CDAA，返回 true；给定 s1 为 ABCD 和 s2 为 ACBD，返回 false。注：右循环移位是指将字符串中每个字符向右侧移动一个位置，最右侧的字符移动到最左侧。例如字符串 ABC 做一次循环移位是 CAB。

输入格式

□□输入为 2 行，依次为 s1 和 s2 字符串。两个字符串的长度都不超过 26，字符为大写英文字母。

输出格式

□□分别为 true 或 false。

输入样例

【输入样例 1】

AABCD

CDAA

【输出样例 1】

true

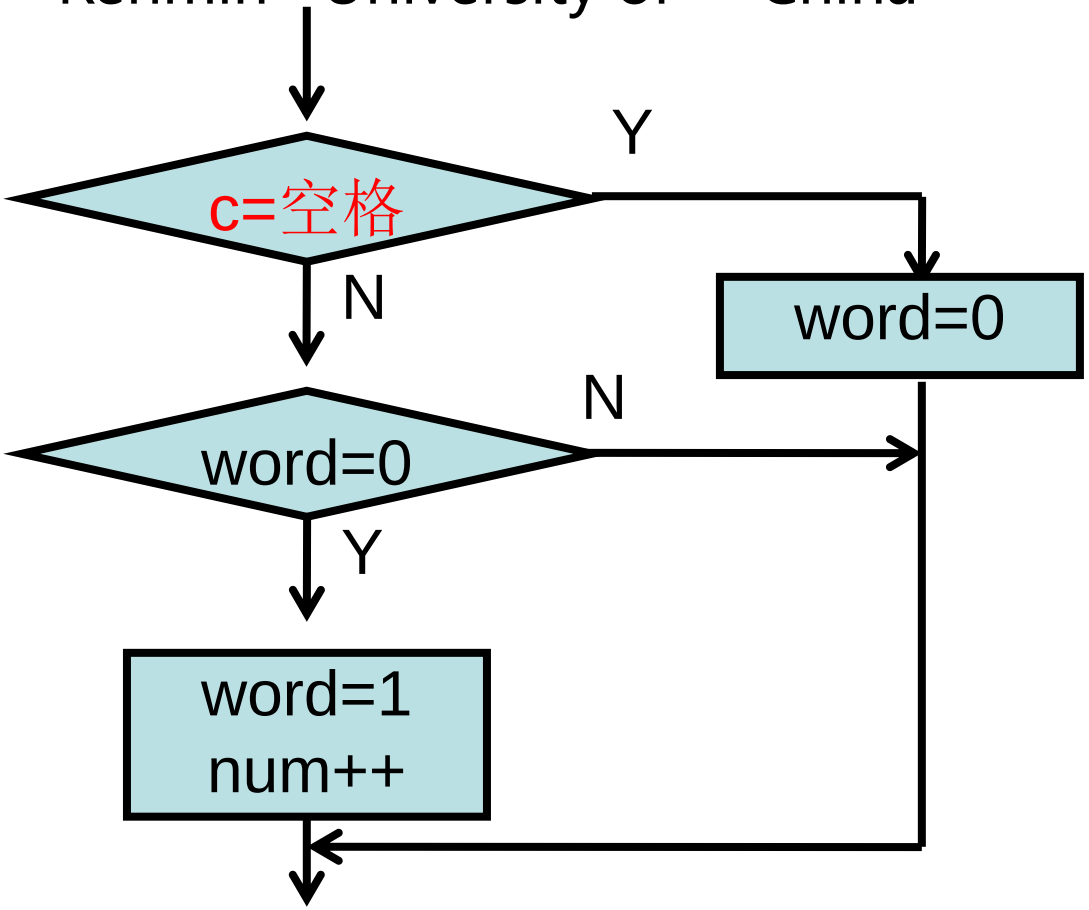
```
#include <stdio.h>
#include <string.h>
int main() {
    char s1[1000] = {0}, s3[2000] = {0}, s2[1000] = {0};
    int i;
    gets(s1);  gets(s2);
    strcpy(s3, s1);  strcat(s3, s1);
    for (i = 0; i < strlen(s1); i++)
        if (strncmp(s3 + i, s2, strlen(s2)) == 0) {
            printf("true");  return 0;
        }
    printf("false");
    return 0;
}
```

s1 为 AABCD 和 s2 为 CDAA，

统计单词

输入一行字符，统计其中有多少个单词，单词之间用空格分隔开。

Renmin University of China



前一个	当前	判断
非空	空	No
非空	非空	No
空	空	No
空	非空	Yes

统计单词

当前字符	I		a	m		a		b	o	y	.
是否空格	**否**	是	**否**	否	是	**否**	是	**否**	否	否	否
word原值	**0**	1	**0**	1	1	**0**	1	**0**	1	1	1
新单词开始否	是	否	是	否	否	是	否	是	否	否	否
word新值	1	0	1	1	0	1	0	1	1	1	1
num值	1	1	2	2	2	3	3	4	4	4	4

……

```
char string[81], c;  int i, num=0,word=0;
gets(string);
for (i=0; (c=str[i])!='\0';  i++)
    if(c==' ') word=0;
    else if(word==0)
        { word=1;
          num++;
        }
printf("%d words\n", num);
```

一定要设初始值

相当于
c=str[i];
c!='\0'

```
I am a boy.
4 words
```

#统计单词.cpp

……

最大字符串

```
#include<stdio.h>
#include<string.h>
int main ( )
{char str[3][10]; char string[10]; int i;
  for (i=0;i<3;i++)
       gets (str[i]);
  if (strcmp(str[0],str[1])>0)
        strcpy(string,str[0]);
  else
        strcpy(string,str[1]);
  if (strcmp(str[2],string)>0)
        strcpy(string,str[2]);
    printf("\nthe largest:\n%s\n",string);
    return 0;
}
```

```
China
Japan
India

the largest:
Japan
```

#110 回文判断

- 回文
  - Able was I ere I saw Elba
  - 落败孤岛孤败落
- 写一个程度，让用户任意输入一个字符串，判断是否是回文

- 分析思路
  - 判断位置i的字符与n-1-i的字符是否相等
  - 循环开始和终止的边界？

#110 回文判断

```
#include <stdio.h>
#include <string.h>
int main()
{   int n,j,i;
    char a[1000];
    gets(a);
    n=strlen(a);
    for (i=0,j=n-1;i<(n+1)/2;i++,j--)
        if (a[i]!=a[j])
            {printf("No");break;}
    if (i==(n+1)/2)   printf("Yes");
    return 0;
}
```

#110 回文判断.cpp

#492 加法表达式

给定一个数字字符串 S，在其中添加 3 个加号使其形成一个加法表达式。遍历所有可以构成合法加法表达式的加号添加方案，计算这些合法加法表达式的值，并按从大到小的顺序输出。

例如：对于数字字符串 12345，添加 3 个加号可以构成的加法表达式包括：

1+2+3+45=51

1+2+34+5=42

1+23+4+5=33

12+3+4+5=24

按从大到小的顺序输出的结果是：51 42 33 24

字符串类

【输入格式】一行，包含一个只含有数字的字符串，长度不超过 15。

【输出格式】一行，若干整数，依次是从大到小输出所有构造合法的加法表达式的计算结果，每 2 个整数之间用一个空格隔开。

【输入样例】

12345

12+34+5

【输出样例】

51 42 33 24

1+234+5

【数据描述】数字字符串 S 的长度不超过 15。

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

int main() {
    char shuru[16], replace[16];
    int i, j, k, l, ic, jc, kc, lc, length, count = 0;
    long long sum[1000];

    gets(shuru);
    length = strlen(shuru);
    for (i = 0; i < length; i++)
        replace[i] = shuru[i] - '0';
```

①

```
for (i = 1; i <= length - 3; i++)
    for (j = 1; j <= length - 3; j++)
        for (k = 1; k <= length - 3; k++)
            for (l = 1; l <= length - 3; l++)
                if (i+j+k+l==length) {
                    long long num1 = 0, num2 = 0, num3 = 0, num4 = 0;
                    for (ic = 1; ic <= i; ic++)
                        num1 = num1 * 10 + replace[ic - 1];
                    for (jc = 1; jc <= j; jc++)
                        num2 = num2 * 10 + replace[i + jc - 1];
                    for (kc = 1; kc <= k; kc++)
                        num3 = num3 * 10 + replace[i + j + kc - 1];
                    for (lc = 1; lc <= l; lc++)
                        num4 = num4 * 10 + replace[i + j + k + lc - 1];
                    sum[count] = num1 + num2 + num3 + num4;
                    count++; }
```

②

```
for (i = 0; ; i++) {
    scanf("%c", &shuru[i]);
    if (s[i] == '\n') break; }
```

不建议这样输入字符串

```
for (i = 0; i < count - 1; i++)
    for (j = 0; j < count - 1; j++)
        if (sum[j] < sum[j + 1]) {
            long long swap;
            swap = sum[j];
            sum[j] = sum[j + 1];
            sum[j + 1] = swap;
        }
```

③

```
for (i = 0; i < count; i++)
    printf("%lld ", sum[i]);
return 0;
```

```
for (i = 1; i <= length - 3; i++)
    for (j = 1; j <= length - 3; j++)
        for (k = 1; k <= length - 3; k++)
            for (l = 1; l <= length - 3; l++)
                if (i + j + k + l == length) {
                    long long num1 = 0, num2 = 0, num3 = 0, num4 = 0;
                    for (ic = 1; ic <= i; ic++)
                        num1 = num1 * 10; + replace[ic - 1];
                    for (jc = 1; jc <= j; jc++)
                        num2 = num2 * 10 + replace[i + jc - 1];
                    for (kc = 1; kc <= k; kc++)
                        num3 = num3 * 10 + replace[i + j + kc - 1];
                    for (lc = 1; lc <= l; lc++)
                        num4 = num4 * 10 + replace[i + j + k + lc - 1];
                    sum[count] = num1 + num2 + num3 + num4;
                    count++; }
```

②

从0起步连续的i个

从i起步连续的j个

从i+j起步连续的k个

从i+j+k起步连续的k个

```
char s[17];
long long int f(int m, int n){
    long long int i, j, sum = 0;
    for (i = m; i < n; i++)
        sum = sum + pow(10, n - i - 1) * (s[i] - 48);
    return (sum);
}
int main(){
    gets(s);
    long long int i, j, t, x, y, z, b, a, c[10000] = {0}, e = 0, max;
    for (i = 1; i <= strlen(s); i++)
        for (j = i + 1; j <= strlen(s); j++)
            for (t = j + 1; t < strlen(s); t++)
                c[e++] = f(0, i) + f(i, j) + f(j, t) + f(t, strlen(s));
    for (i = 0; i < e; i++){
        for (j = i + 1; j < e; j++)
            if (c[i] < c[j])
                {max = c[i]; c[i] = c[j]; c[j] = max; }
        printf("%lld ", c[i]);
    }
}
```

计算数组s中[m,n)中的元素之和

函数风格

①

②

#223 获取密码

某人有一个保险箱，内存贵重物品。他特别害怕忘记了保险箱密码，就将密码记在纸条上，又担心保险箱密码失窃。为此，他设计一个给保险箱密码加密的方法。**保险箱密码为一串数字，加密后还是一串数字**，根据下面规则可以得到明码：首先删除第一个数，紧接着将第二个数放到这串数字的末尾；再将新数字串的第一个数删除，并将第二个数放到这串数字的末尾；如此循环，直到剩下最后一个数；将最后这个数也删除；按照刚才删除的顺序，将这些数字连在一起就是明码。

输入格式：输入保险箱密码的密码，即一串十进制数字，长度在 6 位至 9 位之间，包括 6 位和 9 位。

输出格式：保险箱密码的明码，即长度在 6 位至 9 位之间的一串十进制数字，包括 6 位和 9 位。

输入样例

【输入样例 1】

631758924

【输出样例 1】

615947283

【输入样例 2】

010203

【输出样例 2】

000132

特殊提示：不含数字 0 的输入数据占 50%

密码类题

模拟迭代类题

```
#include <stdio.h>
#include <string.h>
int main() {
    char code[10], decode[10], second;
    gets(code);
    int m, len=strlen(code);
    for(int i=1, m=len; i<=len; i++){
        if(m==1){ decode[len]=code[0]; break; }
①       decode[i]=code[0];   second=code[1];
        for(int j=0; j<=m-3; j++)
②           code[j]=code[j+2];
③       code[m-2]=second;  code[m-1]='\0';
        m--;
    }
    for(int i=1;i<=len;i++)
        printf("%c", decode[i]);
    return 0;
}
```

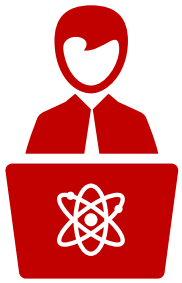
code密码 ⟶ decode明码

① **密文前两个字符的处理**

② **密文后续字符前移动两步**

③ **密文后两字符**

中国人民大学
RENMIN UNIVERSITY OF CHINA



谢谢大家！