

C语言程序设计

计算机科学与技术学院

数组

- 一维数组
- 二维数组
- 字符数组
- 数组相关算法



如何处理字符串？

- 字符串是程序中常见的数据
 - 用户名
 - 密码
 - 学号
 - 留言
 - ...
- 例：程序要求用户输入一个密码
 - 如果是“DHU”，则显示“密码正确”
 - 否则显示“密码错误”

字符串 与 字符数组

- 字符串是一个连续的字符序列
 - 由某个内存地址开始
 - 以空字符 '\0' 为字符串结束的标记
 - "DHU", "100086"
- 在C语言中，“字符串”不是标准的数据类型
- 在C语言中，字符串存放在字符数组中
- 字符数组：元素为字符型(char)的数组
 - `char ch[4]={'D', 'H', 'U', '\0'};`
- 字符串是特殊的字符数组
 - `printf("Hello, world");`

定义和初始化字符数组

// 每个元素为字符型

```
char str1[7] = {'D', 'o', 'n', 'g', 'h', 'u', 'a'};
```

// 不指定元素个数, 以实际字符个数为数组长度

```
char str2[] = {'D', 'o', 'n', 'g', 'h', 'u', 'a'};
```

//保存一个字符串, 必须要保存结束标记'\0'

```
char str3[8] = {'D', 'o', 'n', 'g', 'h', 'u', 'a', '\0'};
```

//初值个数小于数组长度, 则其余元素自动赋为 '\0'

//字符串的长度? 字符数组的长度?

```
char str4[10] = {'D', 'o', 'n', 'g', 'h', 'u', 'a'};
```

//简写形式, 保存字符串时自动追加'\0'

```
char str5[10] = {"Donghua"};
```

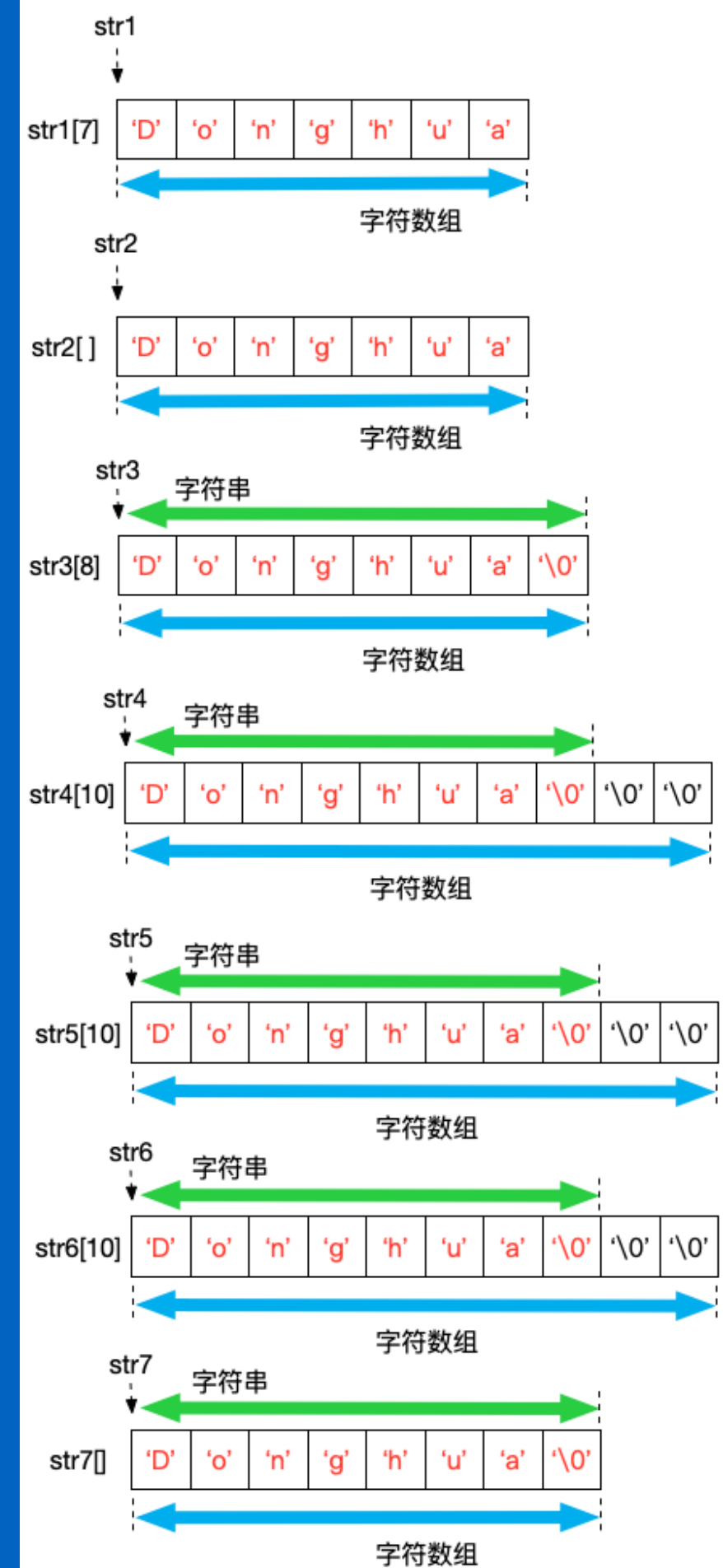
//常用形式1, 保存字符串时自动追加'\0'

//字符串的长度? 字符数组的长度?

```
char str6[10] = "Donghua";
```

//常用形式2, 保存字符串时自动追加'\0'

```
char str7[] = "Donghua";
```



字符数组的初始化

//定义数据并初始化

```
char str[20] = "Donghua";
```

```
char str1[20], str2[20];
```

```
str1 = "Donghua"; //错误：数组不能被赋值
```

```
str2 = str1; //错误：数组不能被赋值
```

```
char str3[20] = "Donghua", str4[20];
```

```
str4 = str3; //错误：数组不能被赋值
```

二维字符数组

```
char fruit[5][7]={"Apple", "Orange", "Grape", "Pear", "Peach"};
```

```
char fruit[][7]={"Apple", "Orange", "Grape", "Pear", "Peach"};
```

fruit[5][7]

fruit[0]	A	p	p	l	e	\0	\0
fruit[1]	O	r	a	n	g	e	\0
fruit[2]	G	r	a	p	e	\0	\0
fruit[3]	P	e	a	r	\0	\0	\0
fruit[4]	P	e	a	c	h	\0	\0

字符数组元素的引用

```
#include <stdio.h>
int main (){
    char str1[9]    = {'C',' ','P','r','o','g','r','a','m'};
    char str2[100]  = {'C',' ','P','r','o','g','r','a','m','\0'};
    char str3[]     = "C Program";
    int i;
    //打印字符数组中的逐个元素
    for(i = 0; i < 9; i++)
        printf("%c", str1[i]);
    printf("\n");
    //打印字符数组中的字符串 ('\0'结束)
    for(i = 0; str2[i] != '\0'; i++)
        printf("%c", str2[i]);
    printf("\n");
    //打印字符数组中的字符串 ('\0'结束)
    printf("%s", str2);
    printf("\n");
    //printf("%s", str1); 为何不行?
    //打印字符数组中的字符串 ('\0'结束)
    printf("%s", str3);
    return 0;
}
```

```
#include <stdio.h>
int main (){
    char diamond[5][6] = {{ ' ',' ','*'},{' ','*',' ','*'},
    {'*',' ',' ',' ','*'},{' ','*',' ','*'},{' ',' ',' ','*'}};
    int i,j;
    for(i = 0; i < 5; i++){
        //for(j = 0; j < 5 ; j++)
        for(j = 0; diamond[i][j] != '\0' ; j++)
            printf("%c", diamond[i][j]);
        printf("\n");
    }

    //for(i = 0; i < 5; i++){
    //printf("%s",diamond[i]);

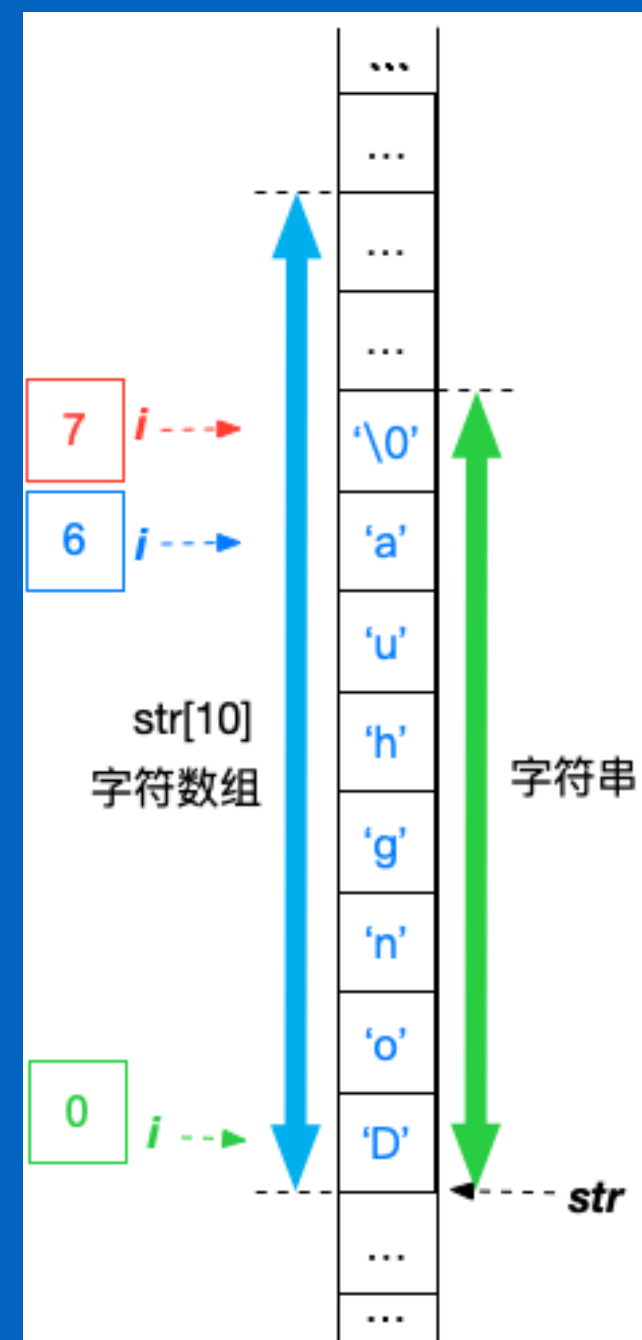
    return 0;
}
```


字符串相关算法

- 字符串的输入输出
- 求字符串的长度
- 字符串的复制
- 字符串的连接
- 字符串的子串
- 字符串的倒置
- 字符串中字符的删除
- 判断回文
- 统计字符类型与个数
- 字符串处理函数

字符串输入输出

- 方式一：字符数组角度
 - 将其视为字符数组中的元素
 - 逐个处理字符数组str[SIZE]中的每个元素 str[i]
 - 输入：str[i] = getchar(), scanf("%c", &str[i])
 - 输出：putchar(str[i]), printf("%c", str[i])
 - 长度不定时，常使用 while 循环
 - 从键盘输入字符串时，常使用 '\n' 判别是否结束
 - 输出字符串时，常使用 '\0' 判别是否结束
- 方式二：字符串角度
 - 将其视为存储在字符数组 str[SIZE] 中的字符串
 - 使用字符串输入输出函数
 - 输入：scanf("%s", str), gets(str)
 - 输出：printf("%s", str), puts(str)
 - 多个字符串，有可使用二维字符数组，如 str[5][6]
 - 每个字符串是二维字符数组中的一行：str[0], str[1], ... , str[i], ...



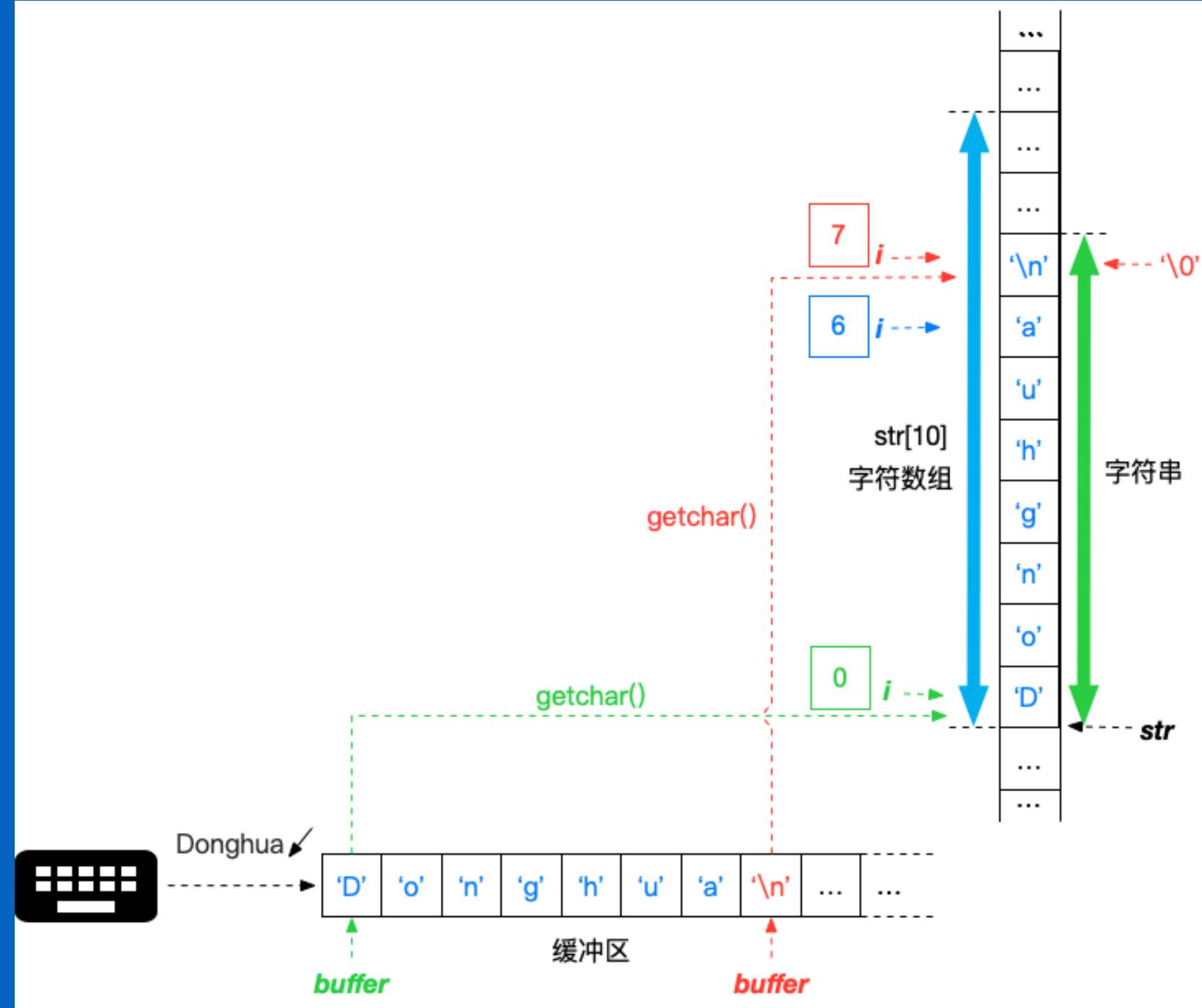
方式一：字符数组角度

```
#include <stdio.h>
#define SIZE 100
int main (){
    char str[SIZE];
    int i = 0;

    //从键盘读取字符（包括'\n'）
    while( (str[i] = getchar( )) != '\n' )
        i++;
    //写入字符串结束标记
    str[i] = '\0';

    //输出字符串
    i = 0;
    while( str[i] != '\0' ){
        putchar( str[i] );
        i++;
    }

    return 0;
}
```



方式二：字符串角度

- 输入：scanf("%s", str)
 - 遇空格或回车结束
 - 存放入字符数组（数组名前不应加 &）
 - 自动加 '\0' 作为结束标记
- 输出：printf("%s", str)
 - 从 str 开始，直到 '\0' 结束（与数组长度无关）

```
#include <stdio.h>
#define SIZE 100
int main (){
    char str[SIZE];
    scanf("%s",str);
    //若输入：Donghua University
    printf("%s",str);
    //打印：Donghua
    return 0;
}
```

- 输入：gets(str)
 - 遇回车结束
 - 存放入字符数组（数组名前不应加 &）
 - 自动加 '\0' 作为结束标记
- 输出：puts(str)
 - 从 str 开始，直到 '\0' 结束（与数组长度无关）
 - 输出后自动换行

```
#include <stdio.h>
#define SIZE 100
int main (){
    char str[SIZE];
    gets(str);
    //若输入：Donghua University
    puts(str);
    //打印：Donghua University
    return 0;
}
```

求字符串的长度：字符串 '\0' 前面的字符个数

- 不使用字符串函数

```
#include <stdio.h>

#define SIZE 100
int main (){
    char str[SIZE];
    int i = 0, length;
    gets(str);
    //对'\0'之前的字符进行计数
    while( str[i] != '\0' )
        i++;
    //i 指向 '\0'
    length = i;
    printf("%d",length);

    return 0;
}
```

- 使用字符串函数

```
#include <stdio.h>
#include <string.h>
#define SIZE 100
int main (){
    char str[SIZE];
    int length;
    gets(str);

    length = strlen(str);
    printf("%d",length);

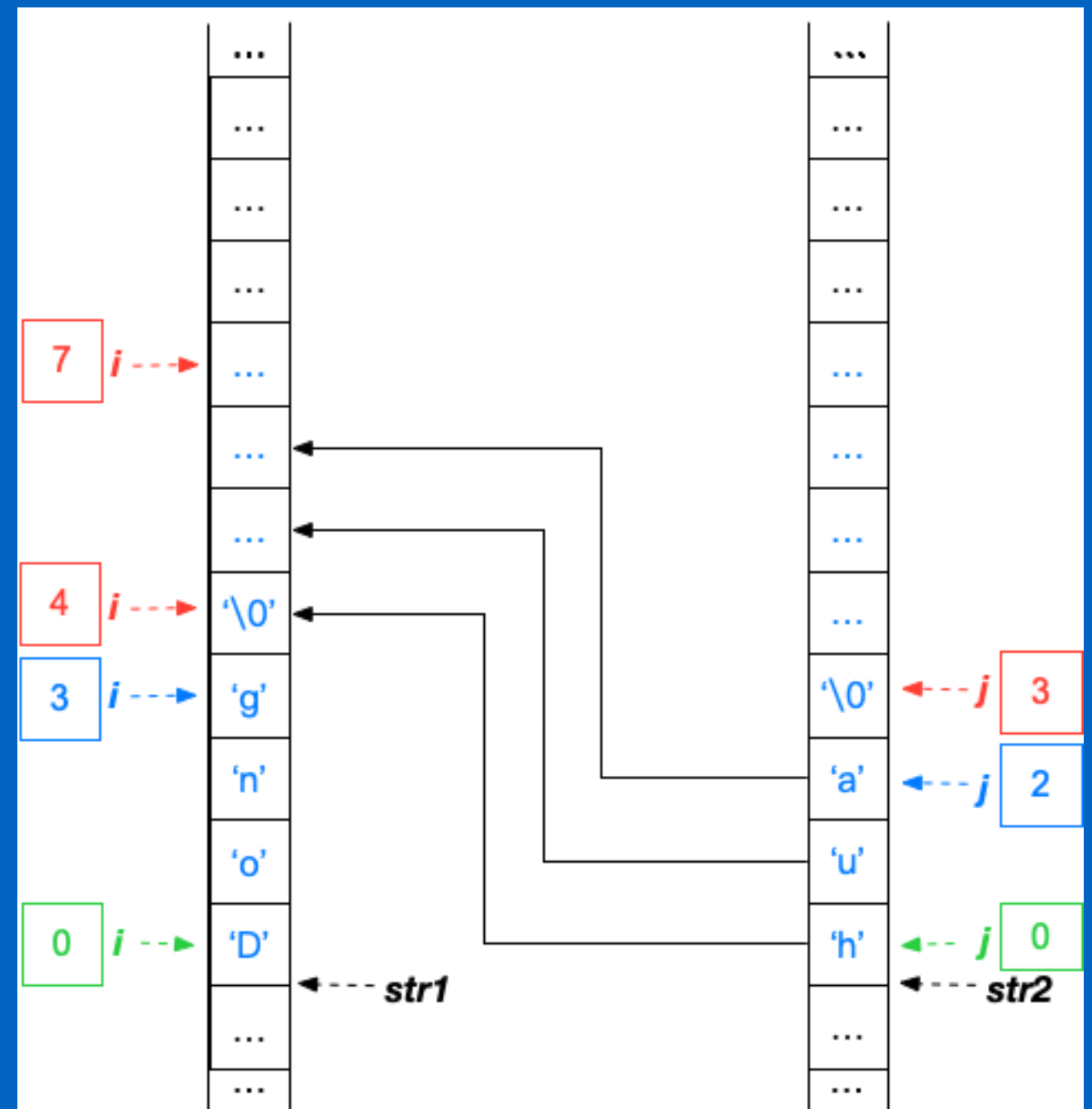
    return 0;
}
```

字符串连接

- 将字符串2连接在字符串1之后

```
#include <stdio.h>
#define SIZE 100
int main (){
    char str1[SIZE], str2[SIZE];
    int i = 0, j = 0;
    printf("Enter string 1:");
    gets(str);
    printf("Enter string 2:");
    gets(str);
    //定位 '\0'
    while(str1[i]!='\0')
        i++;
    //把str2中的字符复制到str1
    while(str2[j]!='\0')
        str1[i++]=str2[j++];
    //添加结束标志
    str1[i]='\0';

    return 0;
}
```



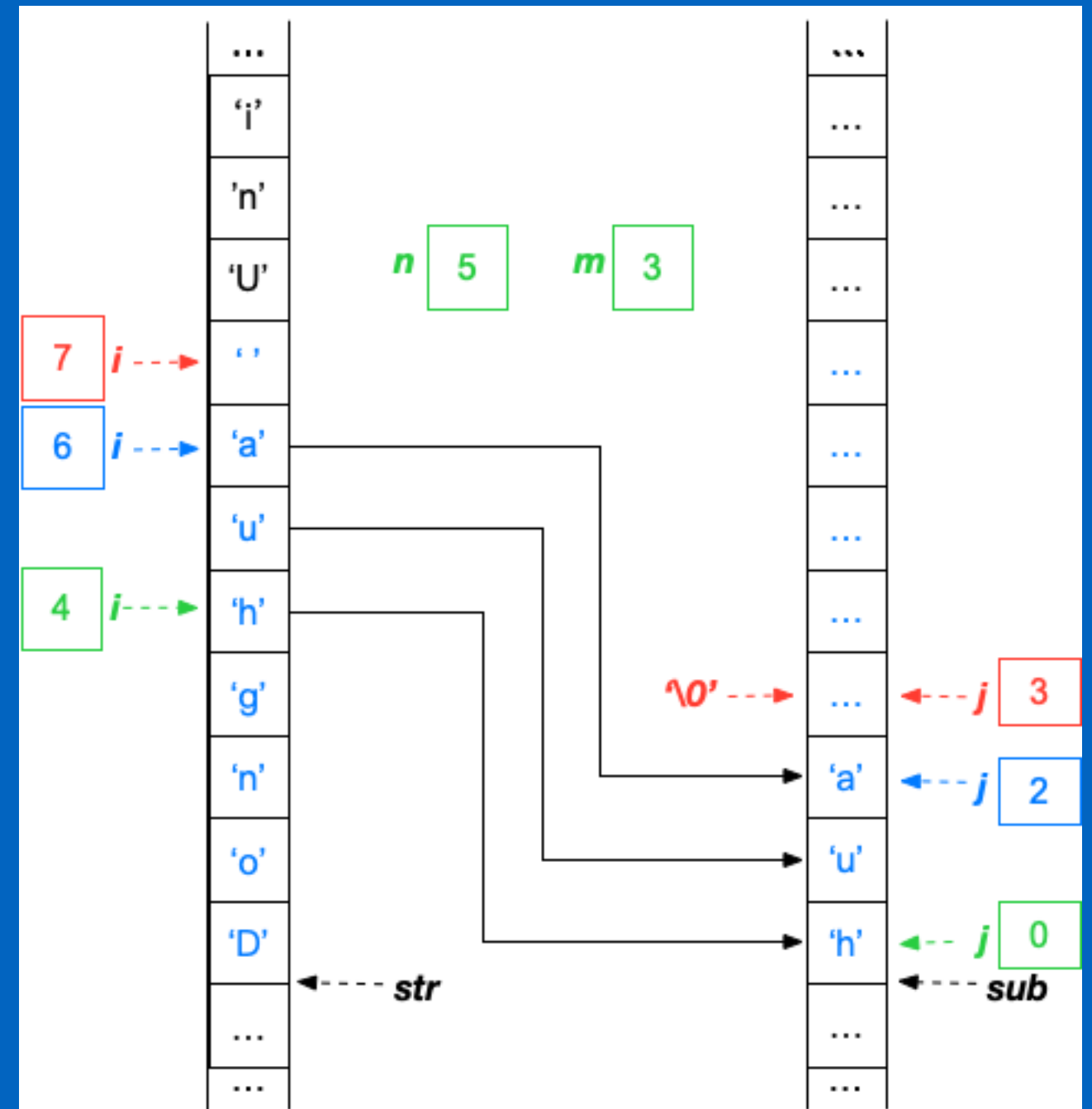
字符串的子串

- 从字符串中自第n个字符起,连续取m个字符,生成一个子串

```
#include <stdio.h>
#define SIZE 100
int main (){
    char str[SIZE], sub[SIZE];

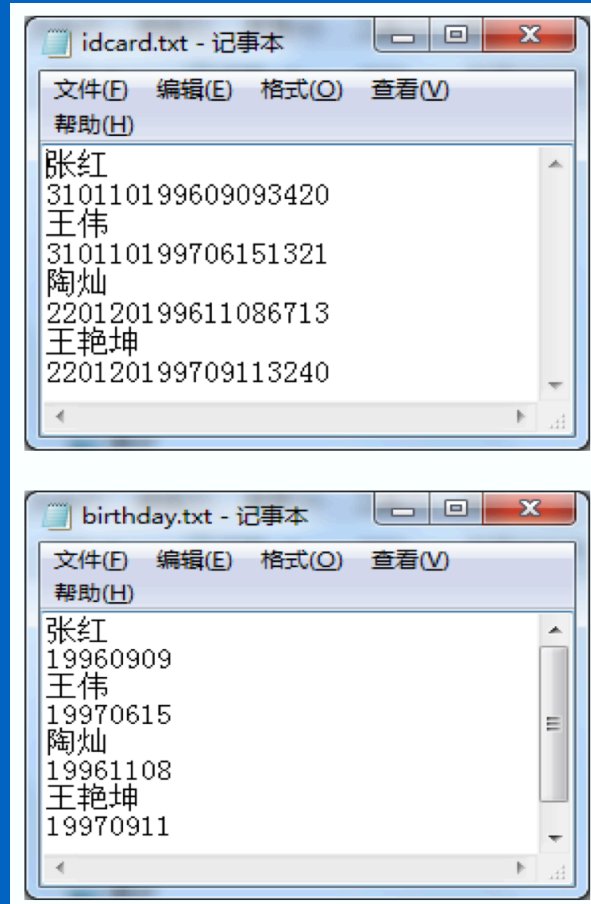
    int i, j, n, m;
    printf("Enter a string:");
    gets(str);
    printf("Enter n,m:");
    scanf("%d, %d", &n, &m);
    //复制部分字符
    for(i = n-1, j = 0; j <= m-1; i++, j++)
        sub[j] = str[i];
    //添加结束标志
    sub[j] = '\0';
    puts(sub);

    return 0;
}
```



从身份证号码中提取生日信息

- 从文件读入若干个人的姓名和身份证号码
- 将该人的姓名和生日信息写入另一个磁盘文件



```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define SIZE 20
int main (){
    char name[SIZE],id[SIZE],bd[SIZE];
    int i, j;
    FILE *fp1, *fp2;

    fp1 = fopen("idcard.txt","r");
    if(fp1 == NULL){
        puts("can't open idcard.txt");
        exit(0);
    }
    fp2=fopen("birthday.txt","w");
    if(fp2 == NULL){
        puts("can't open birthday.txt");
        exit(0);
    }
    // feof() 文件结束(end of file)
    while( ! feof(fp1) ){
        fgets(name, SIZE, fp1); // "张红\n"
        fgets(id, SIZE, fp1);
        for(i = 6, j = 0; i < 14; i++,j++)
            bd[j] = id[i];
        bd[j] = '\0' ; // "19960909"
        fputs(name,fp2);
        fputs(bd,fp2);
        fputc('\n',fp2);
    }

    fclose(fp1);
    fclose(fp2);

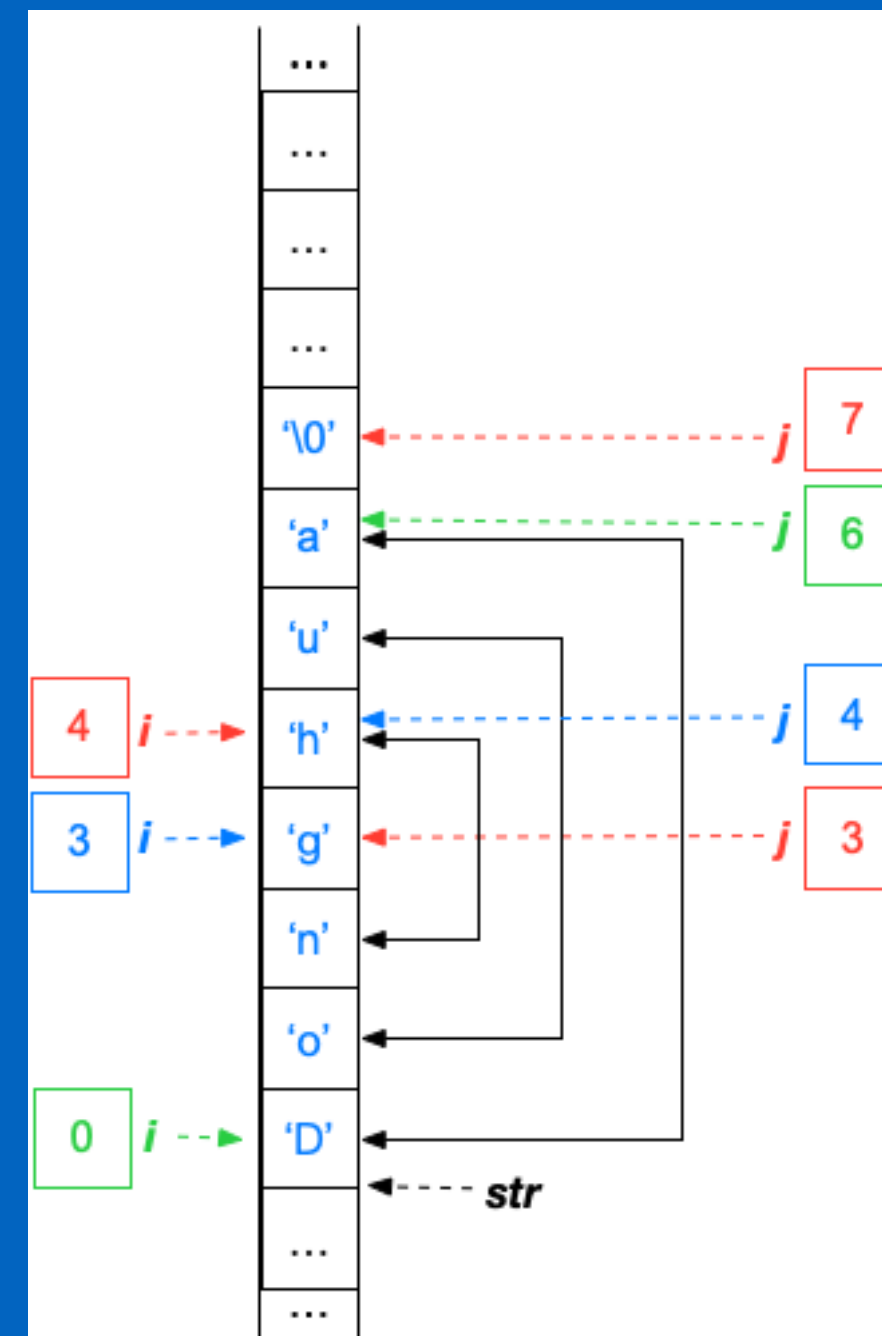
    return 0;
}
```


字符串倒置

```
#include <stdio.h>
#define SIZE 100
int main (){
    char str[SIZE];

    int i, j = 0, t;
    printf("Enter a string:");
    gets(str);
    // 定位到 '\0'
    while( str[j] != '\0' )
        j++;
    // 倒置
    for (i = 0, j = j-1; i < j; i++, j--){
        t = str[i];
        str[i] = str[j];
        str[j] = t;
    }
    puts(str);

    return 0;
}
```



删除字符串中的字符

```
#include <stdio.h>
#define SIZE 100
int main (){
    char str[SIZE],target;
    int i,j;

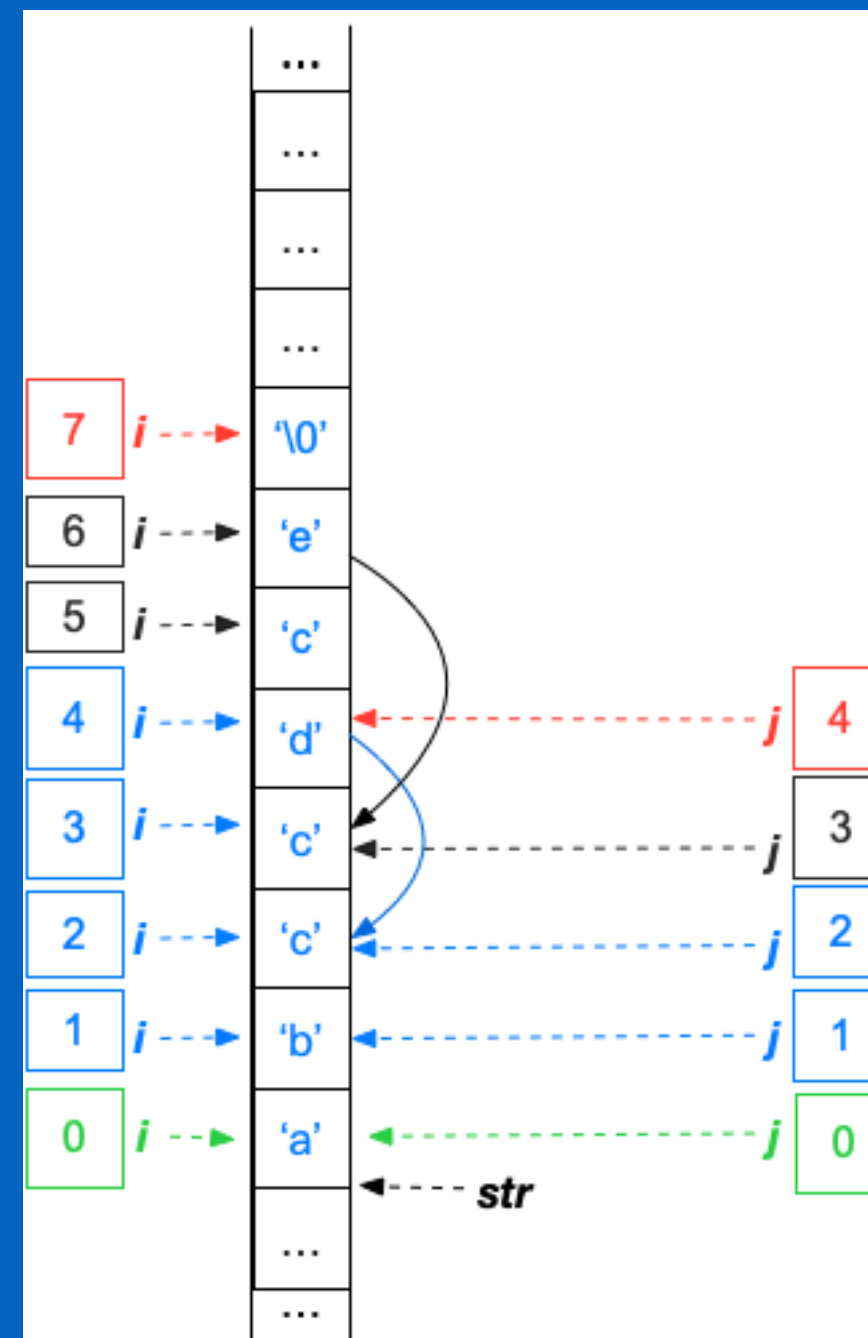
    printf("Enter a string:");// abccdce
    gets(str);
    printf("Enter a character:"); // c
    scanf("%c", &target);

    for(i = 0, j = 0; str[i] != '\0'; i++){
        if(str[i] != target){
            str[j++] = str[i];
        }
    }

    str[j]='\0';

    puts(str); //abde

    return 0;
}
```



题金山寺 苏轼

潮随暗浪雪山倾
远捕渔舟钓月明
桥对寺门松径小
槛当泉眼石波清
迢迢绿树江天晓
霭霭红霞晚日晴
遥望四边云接水
碧峰千点数鸥轻

题金山寺 苏轼

潮随暗浪雪山倾
远捕渔舟钓月明
桥对寺门松径小
槛当泉眼石波清
迢迢绿树江天晓
霭霭红霞晚日晴
遥望四边云接水
碧峰千点数鸥轻

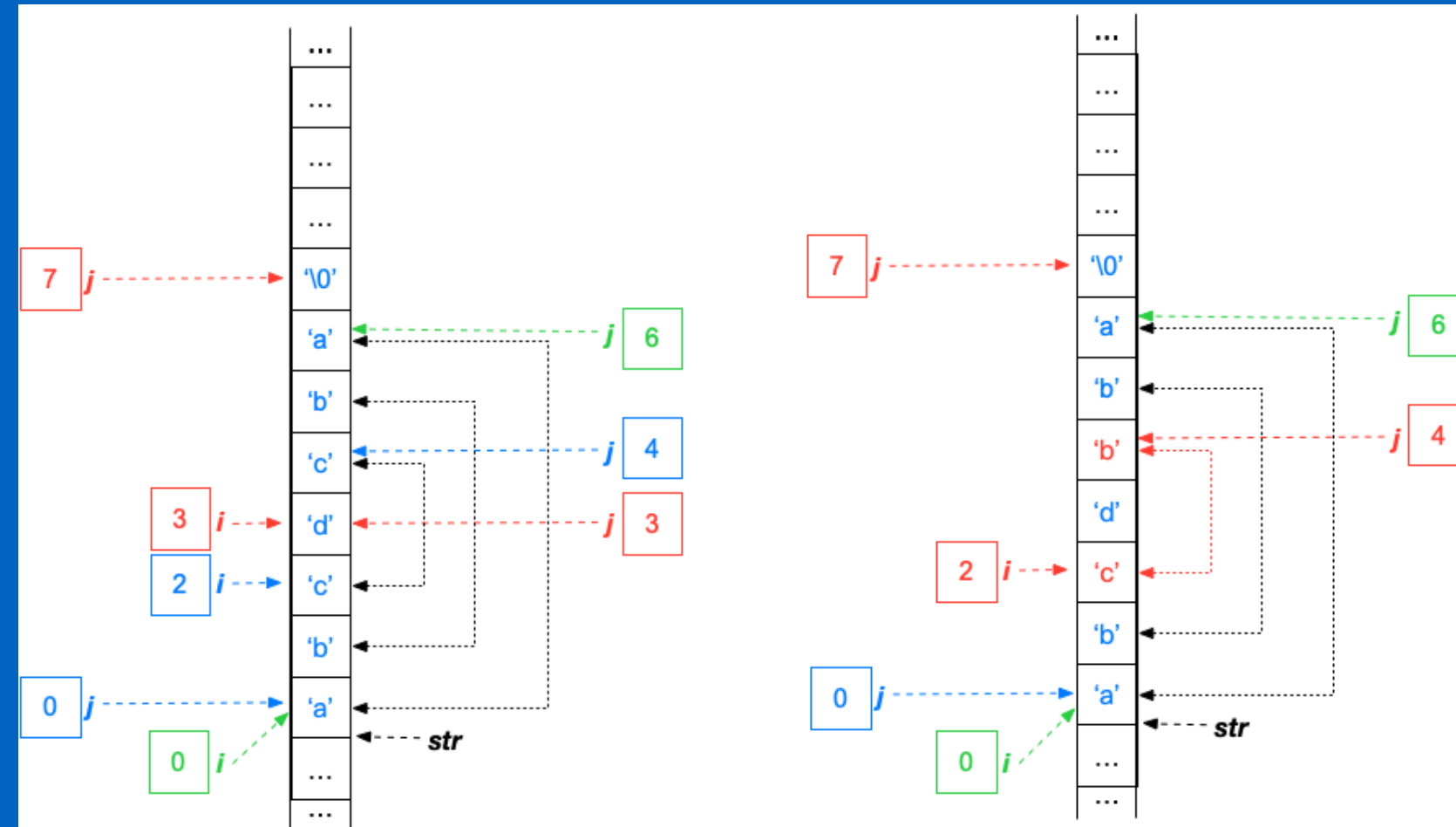
轻鸥数点千峰碧
水接云边四望遥
晴日晚霞红霭霭
晓天江树绿迢迢
清波石眼泉当槛
小径松门寺对桥
明月钓舟渔捕远
倾山雪浪暗随潮

判断回文

- rotator, Madam, I'm Adam, 上海自来水来自海上

```
#include <stdio.h>
#define SIZE 100
int main (){
    char str[SIZE];
    int i, j = 0, flag = 0;
    printf("Enter a string:");
    gets(str);
    //定位 '\0'
    while(str[j] != '\0')
        j++;
    //如果不是回文, 将flag置为1
    for(i = 0, j = j-1; i < j; i++, j--){
        if(str[i] != str[j]){
            flag = 1;
            break;
        }
    }
    if ( flag ) // flag == 1
        //if (i < j) 如果不用 flag
        printf( "This string is not a palindrome." );
    else
        printf( "This string is a palindrome." );

    return 0;
}
```



分别统计字符串中英文字母、空格、数字和其它字符的个数

```
#include <stdio.h>
#define SIZE 100
int main(){
    char str[SIZE];
    int space = 0, letter = 0, digit = 0,
        other = 0, i = 0;
    printf("Enter a string:");
    gets(str); //scanf("%s",str) ?
    while(str[i] != '\0'){
        if((str[i] <= 'Z' && str[i] >= 'A') || (str[i] <= 'z' && str[i] >= 'a'))
            letter++;
        else if (str[i] <= '9' && str[i] >= '0' )
            digit++;
        else if (str[i] == ' ')
            space++;
        else
            other++;
        i++;
    }
    printf("space=%3d,letter=%3d,digit=%3d,other=%3d", space, letter, digit, other);
    return 0;
}
```

字符串处理函数

- string.h
 - strlen()：计算字符串长度函数
 - strcat()：字符串连接函数
 - strcpy()：字符串拷贝函数
 - strcmp()：字符串比较函数

字符串 长度 与 连接

- strlen(str)

- 从 数组地址 开始
- 至 '\0' 结束

```
#include <stdio.h>
#include <string.h>
#define SIZE 100
int main (){
    char str[SIZE];
    int length;
    gets(str);
    length = strlen(str); //从 str 开始, 至 '\0' 结束
    printf("%d",length);

    return 0;
}
```

- strcat(str1,str2)

- 去掉str1中的'\0'
- 把str2中的字符串连接到str1之后

```
#include <stdio.h>
#include <string.h>
#define SIZE 100
int main (){
    char str1[SIZE] = "Dong", str2[] = "hua";
    strcat(str1, str2);
    printf("%s",str1);

    return 0;
}
```


字符串 连接 与 复制

- strcat(str1,str2)

- 去掉str1中的'\0'
- 把str2中的字符串连接到str1之后

```
#include <stdio.h>
#include <string.h>
#define SIZE 100
int main (){
    char str1[SIZE] = "Dong", str2[] = "hua";
    strcat(str1, str2);
    printf("%s",str1);

    return 0;
}
```

- strcpy(str1,str2)

- 把str2中的字符串copy到str1之后
- 连同str1中的'\0'

```
#include <stdio.h>
#include <string.h>
#define SIZE 100
int main (){
    char str1[SIZE] = "abc", str2[] = "Donghua";
    strcpy(str1, str2);
    //str1 = str2; 出错
    printf("%s",str1);

    return 0;
}
```

字符串 比较

- strcmp(str1,str2)
 - 从左向右逐个字符比较两个字符串 (ASCII码)
 - 直到遇到不同字符或'\0'为止
 - 字符串比较不能用“==”,必须用strcmp
- 返回值: int型整数
 - 若字符串1== 字符串2, 返回 0 (结果=0)
 - 若字符串1> 字符串2, 返回 1 (结果>0)
 - 若字符串1< 字符串2, 返回-1 (结果<0)

- 尝试密码

```
#include <stdio.h>
#include <string.h>
#define SIZE 100
int main (){
    char str1[SIZE],
           str2[] = "123456";
    while(1) {
        gets(str);
        if(strcmp(str1,str2))
            puts("Invalid password.\n");
        else
            break;
    }
    puts("Correct.\n");

    return 0;
}
```

```
strcmp("A","B"):    <0
strcmp("a","A"):    >0
strcmp("computer","compare"):    >0
```

找出三个字符串中的最大者

- "computer", "compare", "competition" ?

```
#include <stdio.h>
#include <string.h>
#define SIZE 100
int main (){
    char str1[SIZE],str2[SIZE],str3[SIZE],
        string[SIZE];
    printf("Enter 3 strings");
    gets(str1);
    gets(str2);
    gets(str3);
    if(strcmp(str1,str2) > 0)
        strcpy(string,str1);
    else
        strcpy(string,str2);
    if(strcmp(str3,string) > 0)
        strcpy(string,str3);
    printf("The largest string is:%s",string);
    return 0;
}
```

```
#include <stdio.h>
#include <string.h>
#define SIZE 100
int main (){
    char str[3][SIZE],string[SIZE];
    int i;
    printf("Enter 3 strings");
    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("The largest string is:%s",string);
    return 0;
}
```