**一、基本概念**

串（字符串）：是由**零个**或多个**字符**组成的有限序列，是一类特殊的线性表

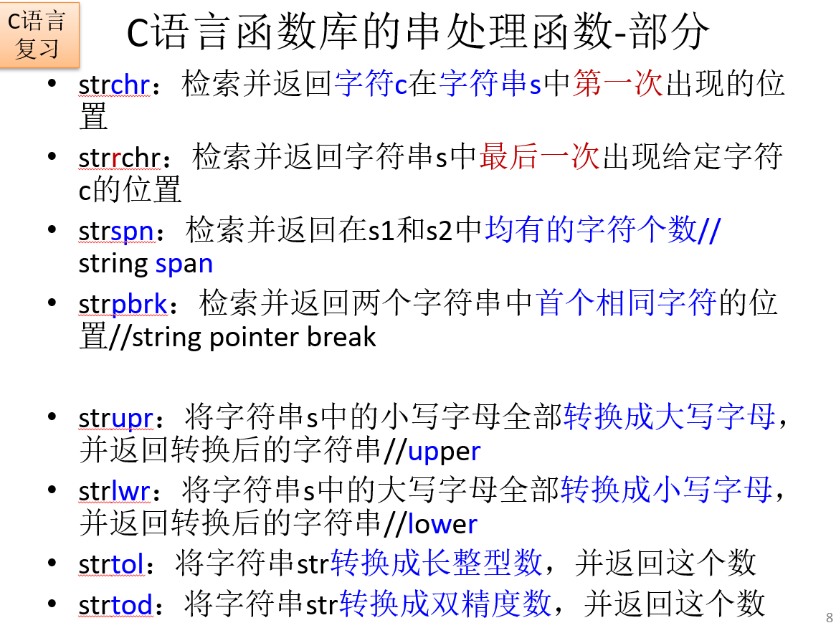
s = “a1a2… an”, n>=0

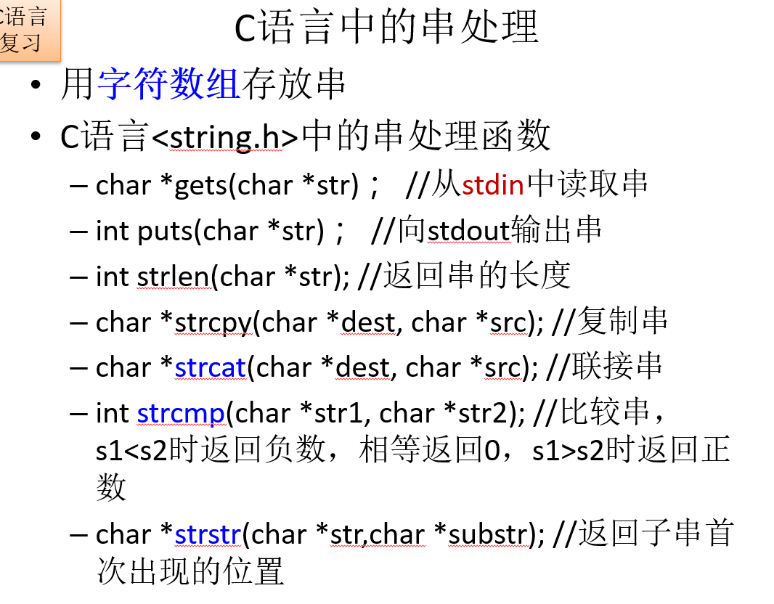
空串：n=0

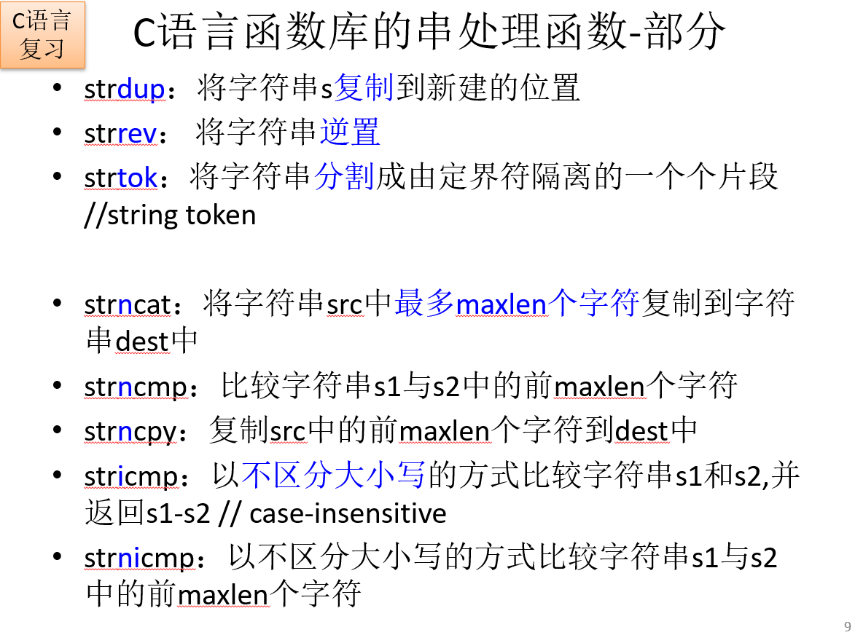
主串、子串

串的前缀：串中最靠前的若干个字符

串的后缀：串中最靠后的若干个字符

真前缀、真后缀





**二、串的具体实现**

**2.1 串：顺序串/定长顺序存储**

压缩存储：1个32位的内存单元存储4个字符

非压缩储存：1个32位的内存单元存储1个字符

**数据结构：SString[MAXSTRLEN+1]字符数组**

**2.2 串：顺序串/堆分配存储**

typedef struct {

char \*ch; //若是非空串，则按串长度+1分配存储区

int strsize; //存储空间大小，包含串的结束符

int length; //当前已写入的串长

} HString;

**2.3 块链存储**

**LString块链串：**

Chunk \*head; //头指针，指向首个字符

Chunk \*tail; //尾指针

int curLen; //结点个数

**结点Chunk：**

char ch[CHUNKSIZE];//每个结点中的数据域

struct Chunk \* next ;//每个结点中的指针域

**3. 模式匹配**

**暴力算法**

设主串的长度是n，模式串的长度是m

暴力匹配：时间复杂度O(m\*n)

**KMP算法**

时间复杂度O(n+m)

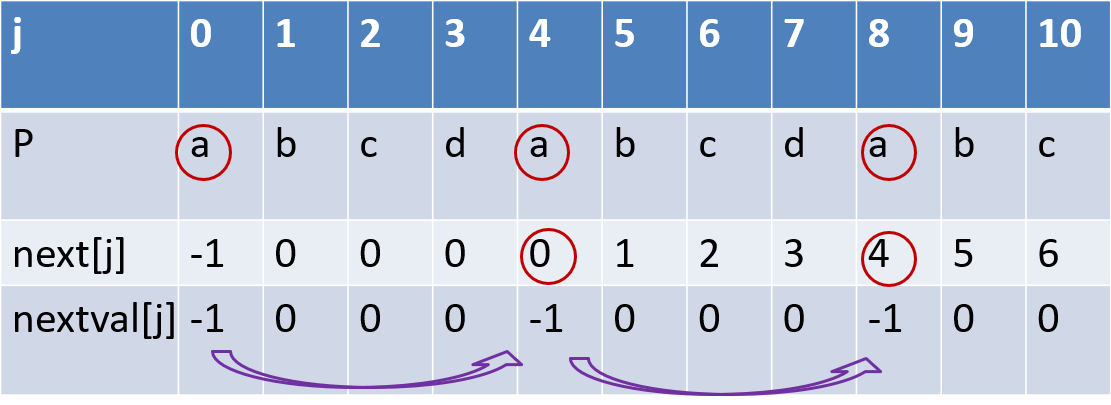
1.根据子串求出子串的next数组

next[0]= -1; next[1]= 0；

next=前缀能匹配的字符数量

**在p[j]处失配，k=p在p[0]~p[j-1]中自匹配子串的最大长度**

此处失配

****

优化：如果p[j] == p[ next[j] ]，那么将next[j]的值换为next[ next[j] ]

省去了跳转后仍然连续失配的情况

2.从头开始匹配，

若主串的第i个字符 与模式串的第j个的字符失配，那么，

* + 在next[j]== -1时，主串后（i++）移指针，模式串重头开始
  + 否则，(主串的指针不动)主串的第i个字符下一次将与模式串的next[j]个的字符匹配

若当前i、j所指字符匹配成功，i++，j++；

**4. 串操作应用举例**



aaa**a**caaab

aaa**b**—1

aaaa**c**aaab

**a**—2

aaaacaaab

aaab