第四章 串

一、选择题

1.下列关于串的叙述中，正确的是（ ）

(A)一个串的字符个数即该串的长度 (B)一个串的长度至少是 1

(C)空串是由一个空格字符组成的串 (D)两个串 S1 和 S2 若长度相同，则这两个串相等

2.字符串"abaaabab"的 nextval 值为( )

(A)(0,1,01,1,0,4,1,0,1) (B)(0,1,0,0,0,0,2,1,0,1)

(C)(0,1,0,1,0,0,0，1,1) (D)(0,1,0,1,0,1,0,1,1)

3.字符串满足下式,其中 head 和 tail 的定义同广义表类似,如 head(‘xyz’)= ‘x’ ,tail(‘xyz’)= ‘yz’ ,则 s=( )。concat(head(tail(s)),head(tail(tail(s))))= ‘dc’。

(A)abcd (B)acbd (C)acdb (D)adcb

4.串是一种特殊的线性表,其特殊性表现在（ ）

(A)可以顺序存储 (B)数据元素是一个字符

(C)可以链式存储 (D)数据元素可以是多个字符

5．设串 S1=‘ABCDEFG’,s2=‘PQRST’,函数 CONCAT（X，Y）返回 X 和 Y 串的连接串，SUBSTR（S，I，J）返回串 S 从序号 I 开始的 J 个字符组成的字串，LENGTH（S）返回串 S 的长度，则 CONCAT（SUBSTR（S1，2，LENGTH（S2)），SUBSTR（S1，LENGTH（S2），2））的结果串是（ ）

（A）BCDEF (B) BCDEFG (C)BCPQRST (D)BCDEFEF

二、算法设计

1.分别在顺序存储和一般链接存储两种方式下,用 C 语言写出实现把串 s1 复制到串 s2 的串复制函数 strcpy(s1,s2)。

2.在一般链接存储(一个结点存放一个字符)方式下,写出采用简单算法实现串的模式匹配的 C 语言函数 int L_index(t,p)。

参考答案：

一、选择题

1. A 2.B 3. D 4. D 5. D

二、算法设计

1.

顺序存储：

```
#include "string.h"
#define MAXN 100
char s[MAXN];
int S_strlen(char s[])
{
int i;
for(i=0;s[i]!='\0';i++);
return(i);
}
void S_strcpy(char s1[],char s2[]) //4.3 题
{
int i;
for(i=0;s1[i]!='\0';i++)
s2[i]=s1[i];
```

```
s2[i]='\0';
}
```

一般链接存储：

```
#include "stdio.h"
typedef struct node
{
char data;
struct node *link;
}NODE;
NODE *L_strcpy(NODE *s1)
{
NODE *s2,*t1,*t2,*s;
if(s1==NULL) return(NULL);
else
{
t1=s1;
t2=(NODE *)malloc(sizeof(NODE));
s2=t2;
while(t1!=NULL)
{
s=(NODE *)malloc(sizeof(NODE));
s->data=t1->data;
```

```
t2->link=s;
t2=s;
t1=t1->link;
}
t2->link=NULL;
s=s2;
s2=s2->link;
free(s);
return(s2);
}
}
```

2.

```
#include "stdio.h"
typedef struct node
{
char data;
struct node *link;
}NODE;
int L_index(NODE *t,NODE *p)
{
NODE *t1,*p1,*t2;
?int i;
```

```
t1=t;i=1;
while(t1!=NULL)
{
p1=p;
t2=t1->link;
while(p1->data==t1->data&&p1!=NULL)
{
p1=p1->link;
t1=t1->link;
}
if(p1==NULL) return(i);
i++;
t1=t2;
}
return(0);
}
```