

实验报告5 参考答案

1 串的基本操作和线性表有很大差别,其主要区别是什么?请分别简单解释。

答:串的基本操作与线性表有很大区别。在线性表的基本操作中,大多以“单个元素”作为操作对象,例如在线性表中查找某个元素,求取某个元素,在某个位置上插入或删除一个元素;而在串的基本操作中,通常以“串的整体”作为操作对象,例如在串中查找某个子串,求取一个字串,在串的某个位置上插入一个字串以及删除一个字串。

2. 设 $s = \text{'I AM A STUDENT'}$, $t = \text{'GOOD'}$, $q = \text{'WORKER'}$, 求: (1)
 $\text{StrLength}(s), \text{StrLength}(t)$

答: $\text{StrLength}(s) = 14$; $\text{StrLength}(t) = 4$.

(2) $\text{SubString}(s, 8, 7)$, $\text{SubString}(t, 2, 1)$

答: $\text{SubString}(s, 8, 7) = \text{'STUDENT'}$, $\text{SubString}(s, 8, 7)$ 实现的操作是: 在串 s 中取出第 8 个字符后长度为 7 的子。

$\text{SubString}(t, 2, 1) = \text{'O'}$ $\text{SubString}(t, 2, 1)$ 实现的操作是: 在串 t 中取出第 2 个字符后长度为 1 的子串。

(3) $\text{Index}(s, \text{'A'}, 1), \text{Index}(s, t, 3)$

答: $\text{Index}(s, \text{'A'}, 1) = 3$; $\text{Index}(s, \text{'A'}, 1)$ 实现的操作是: 在主串 s 中的第 1 个字符后第一次出现 A 的位置。

Index(s,t,3)=0,Index(s,t,3) 实现的操作是:在主串 s 中的第 3 个字符后第一次出现 t 这个子串的位置,由于 s 中没有子串 t,因此返回值为 0。

(4) Replace(s, 'STUDENT', q)

答:Replace(s, 'STUDENT', q)='I□AM□A□WORKER',实现的操作是:用 q='WORKER'替换主串 s 中'STUDENT'的子串。

(5) Concat(SubString(s,6,2),Concat(t,SubString(s,7,8)))

答: Concat(SubString(s,6,2),Concat(t,SubString(s,7,8)))='A□GOOD□STUDENT'

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

```
#define MAXSIZE 255
typedef unsigned char SString[MAXSIZE+1];
```

```
int StrAssign(SString T, char *chars){    // 生成串
```

```
    int i;
    if(strlen(chars)>MAXSIZE)
        return 0;
    T[0]= strlen(chars);
    for(i=1;i<=T[0];i++)
        T[i]=*(chars+i-1);
    return 1;
}
```

```
int StrCopy(SString T, SString S){    // 拷贝串
```

```
    int i;
```

```

    S[0]=T[0];
    for(i=1;i<=T[0];i++)
        S[i]=T[i];
    return 1;
}

```

```

int SubString(SSString T, SString S, int pos, int len){ // 求子串

```

```

    int i ;
    if(pos<1||pos>T[0]||len>T[0]-pos)
        return 0;
    S[0]=len;
    for(i=1;i<=len;i++)
        S[i]=T[pos+i-1];
    return 1;
}

```

```

int ClearString(SSString T){ //清空串

```

```

    int i;
    if(!T[0])
        return 0;
    for(i=1;i<=T[0];i++)
        T[i]='\0';
    T[0]=0;
    return 1;
}

```

```

int StrDelete(SSString T, int pos, int len){

```

```

    int i,j;
    j=T[0];
    if(pos<1||pos>T[0]||len>T[0]-pos)
        return 0;
    for(i=1;i<T[0]-pos-len+1;i++)
        T[pos+i-1]=T[pos+len+i-1];
    for(i=1;i<=len;i++)
        T[j-i+1]='\0';
    T[0]=T[0]-len;

    return 1;

}

```

```

void Strprint(SSString T){

    int i;
    for(i=1;i<=T[0];i++)
        printf("%c",T[i]);
    printf("\n");
}

int main(){

    int pos,len,a=1,b;
    SString S,T,C;

    char chars[255];
    printf("1.生成串\n");
    printf("2.清空串\n");
    printf("3.拷贝串\n");
    printf("4.求子串\n");
    printf("5.删除子串\n");
    printf("0.退出操作\n");

    while(a!=0){
        printf("input 0--5: ");
        scanf("%d",&a);

        switch(a){
            case 1:
                printf("请输入字符串:");
                scanf("%s",chars);
                b=StrAssign(T,chars);
                if(!b){
                    printf("输入的字符串过长\n");
                    exit(0);
                }
                printf("字符串 S 为: \n");
                Strprint(T);
                break;

            case 2:
                ClearString(T);
                printf("现在的串为:");
                Strprint(T);

```

```

        printf("\n");
        break;
    case 3:
        StrCopy(T,S);
        printf("拷贝的串 S 为: ");
        Strprint(S);
        printf("\n");
    case 4 :
        printf("请输入子串的起始位置");
        scanf("%d",&pos);
        printf("请输入子串的长度");
        scanf("%d",&len);
        b=SubString(T,C,pos,len);
        if(!b){
            printf("输入的数据有错误! \n");
            exit(0);
        }
        printf("子串 C 为: ");
        Strprint(C);
        printf("\n");
        break;
    case 5 :
        printf("请输入子串的起始位置");
        scanf("%d",&pos);
        printf("请输入子串的长度");
        scanf("%d",&len);
        b=StrDelete(T,pos,len);
        if(!b){
            printf("输入的数据有错误! \n");
            exit(0);
        }
        printf("删除后的子串 T 为: ");
        Strprint(T);
        printf("\n");
        break;
    case 0:
        exit(0);
    }
}
}

```

