# 任务05 串

# 任务05\_1 串的定长顺序存储

请根据要求，按照**C语言语法，**编写串的三个函数：StrAssign、StrInsert、PrintStr，完成规定的运行效果，并将这三个函数提交。

**预定义常量和类型**

#define Status int

#define OK 1

#define ERROR 0

#define INFEASIBLE -1

**定长顺序串存储结构**

#define MAXSTRLEN 255

typedef unsigned char SString[MAXSTRLEN + 1];

**三个函数的函数原型及功能如下：**

Status StrAssign(SString T,char \* chars) ：串赋值函数，完成将chars指向的字符串赋值给T指向的定长顺序串;

Status StrInsert(SString S, int pos, SString T)：将字符串T，插入到S所指向串的第pos个位置，若发生截断的话，返回INFEASIBLE，若pos位置不合法的话返回ERROR，若插入成功的话，返回OK;

void PrintStr(SString S):打印串中字符

**编写要求：**

1. 请用C语言完成函数编写，且符合C89语法规则（影响成绩评定）

2. 运行效果必须严格符合要求（影响成绩评定）

3. 需要提交StrAssign、StrInsert、PrintStr这三个函数

**main函数如下：**

int main(void)

{

SString S, T;

Status result;

StrAssign(S, "abcdefhijklmn");

StrAssign(T, "UVWXYZ");

result = StrInsert(S, 4, T);

if(OK == result)

{

printf("未发生截断，结果是：");

PrintStr(S);

}

else if(INFEASIBLE == result){

printf("发生截断!\n");

}

else {

printf("插入位置有误!\n");

}

return 0;

}

**运行效果：**

未发生截断，结果是：abcUVWXYZdefhijklmn

请按任意键继续. . .

**提交格式：**

StrAssign函数如下：

StrInsert函数如下：

PrintStr函数如下：

# 任务05\_2 串的堆分配存储

请根据要求，按照**C语言语法，**编写串的五个函数：StrAssign、StrCompare、SubString、Strfun、PrintStr，完成规定的运行效果，仅提交StrCompare、SubString、Strfun这三个函数即可。

**预定义常量和类型**

#define Status int

#define OK 1

#define ERROR 0

#define OVERFLOW -2

**堆分配串存储结构**

typedef struct {

char \*ch;

int length;

}HString;

**五个函数的函数原型及功能如下：**

Status StrAssign(HString \*T,char \*chars) ：串赋值函数，完成将chars指向的字符串赋值给T指向的定长顺序串;

int StrCompare(HString S,HString T)：串比较函数，S中的字符和T中字符比较，若S>T,返回大于0的数；若S=T，返回0；若S<T，返回小于0的数；

Status SubString(HString \*Sub,HString S,int pos,int len):求子串函数

void PrintStr(HString S)：将S中的字符打印出来

Status Strfun(HString \*S,HString T)：查找S中是否有与T相同的子串，有的话，删除，返回ERROR；没有的话，返回OK。

**编写要求：**

1. 请用C语言完成函数编写，且符合C89语法规则（影响成绩评定）

2. 运行效果必须严格符合要求（影响成绩评定）

3. 仅提交StrCompare、SubString、Strfun这三个函数

**main函数如下：**

int main(void)

{

HString S ,T;

S.ch = T.ch = NULL;

StrAssign(&S ,"abcFGdeFGHiGH");

PrintStr(S);

StrAssign(&T ,"FGH");

PrintStr(T);

Strfun(&S,T);

PrintStr(S);

return 0;

}

**运行效果：**

abcFGdeFGHiGH

FGH

abcFGdeiGH

请按任意键继续. . .

**提交格式：**

StrCompare函数如下：

SubString函数如下：

Strfun函数如下：

# 任务05\_3 模式匹配算法

设字符串S=' aabbaaabaaaacb '，P='abaaa'

（1）请以表格形式给出S和P的next数组。（表格形式如下：）

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S | a | a | b | b | a | a | a | b | a | a | a | a | c | b |
| next | 0 | 1 | 2 | 1 | 1 | 2 | 3 | 3 | 4 | 2 | 3 | 3 | 3 | 1 |
| P | a | b | a | a | a |  | | | | | | | | |
| next | 0 | 1 | 1 | 2 | 2 |

（2）若S做主串，P做模式串，请给出利用KMP算法给出匹配的过程。（本小题答案格式具体参考书上82页）

a a b b a a a b a a a a c b

0 1 2 1 1 2 3 3 4 2 3 3 3 1

a b a a a

0 1 1 2 2

#include <stdio.h>

#include <string.h>

#define Status int

#define OK 1

#define ERROR 0

#define INFEASIBLE -1

#define MAXSTRLEN 255

typedef unsigned char SString[MAXSTRLEN + 1];

Status StrAssign(SString T, char \* chars) {

int i;

if (strlen(chars) > MAXSTRLEN)

{

for (i = 1; i <= MAXSTRLEN; ++i)

{

T[i] = \*(chars + i - 1);

}

T[0] = MAXSTRLEN;

}

else

{

T[0] = strlen(chars);//此时T[0]保存的为int类型的数据

for (i = 1; i <= MAXSTRLEN; i++)

{

T[i] = \*(chars + i - 1);

}

return OK;

}

}

Status StrInsert(SString S, int pos, SString T)

{

int i;

if (S[0] + T[0] <= MAXSTRLEN)

{

for (i = S[0] + T[0]; i >= pos + T[0]; --i)

S[i] = S[i - T[0]];

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

S[pos + i - 1] = T[i];

S[0] = S[0] + T[0];

return OK;

}

else

{

for (i = MAXSTRLEN; i >= pos + T[0]; --i)

{

S[i] = S[i - T[0]];

}

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

{

S[pos + i - 1] = T[i];

}

S[0] = MAXSTRLEN;

return INFEASIBLE;

}

}

void PrintStr(SString S) {

int i;

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

{

printf("%c", S[i]);

}

printf("\n");

//////////////////////////////return OK;

}

int main(void)

{

SString S, T;

Status result;

StrAssign(S, "abcdefhijklmn");

StrAssign(T, "UVWXYZ");

result = StrInsert(S, 4, T);

if (OK == result)

{

printf("未发生截断，结果是：");

PrintStr(S);

}

else if (INFEASIBLE == result) {

printf("发生截断!\n");

}

else {

printf("插入位置有误!\n");

}

getchar();

return 0;

}