代码：

#include "stdio.h"

#include "stdlib.h"

# define MAXSIZE 100

//顺序串的头文件

typedef struct{

char str[MAXSIZE];

int length;

}seqstring;

//字符串的创建

seqstring \*create()

{

int m = 0;

seqstring \*p;

p = (seqstring \*)malloc(sizeof(seqstring));

printf("请输入将要输入的字符串长度:\n");

scanf("%d",&m);

p->length = m;

printf("请输入字符串:\n");

scanf("%s",p->str);

return p;

}

//顺序串的插入

seqstring \*strinsert(seqstring \*S,int i,seqstring \*T)

{

int k;

if(i < 1 || i > S->length + 1 || S->length + T->length > MAXSIZE -1)/\*非法判断\*/

printf("字符串过长，无法插入！\n");

else

{

//从第i个元素开始后移

for(k = S->length; k >= i - 1;k --)

S->str[T->length + k] = S->str[k];

//将T写入S中的第i个字符串开始的位置

for(k = 0;k < T->length; k ++)

S->str[i + k - 1] = T->str[k];

S->length = S->length + T->length;

S->str[S->length] = '\0';

}

return S;

}

//字符串的删除

seqstring \*strdelete(seqstring \*S,int i,int len)

{

int k;

if(i < 1 || i > S->length || i + len - 1 > S->length) /\*非法情况的处理\*/

printf("无法插入字符串！");

else

{

for(k = i + len - 1; k < S->length; k ++)

//s中从下标为i + len - 1开始的元素前移

S->str[k - len] = S->str[k];

S->length = S->length - len;

S->str[S->length] = '\0';

}

return S;

}

//字符串的连接

seqstring \*strconcat(seqstring \*S1,seqstring \*S2)

{

int i;

seqstring \*r;

//处理字符串数组空间不够使用的情况

if(S1->length + S2->length > MAXSIZE - 1)

{

printf("无法插入字符串！\n");

return (NULL);

}

else

{

r = (seqstring \*) malloc(sizeof(seqstring));

//将s1复制到r字符串的前端

for(i = 0;i < S1->length;i ++)

r->str[i] = S1->str[i];

//将s2复制到r字符串的后端

for(i = 0;i < S2->length;i ++)

r->str[S1->length + i] = S2->str[i];

r->length = S1->length + S2->length;

r->str[r->length] = '\0';

}

return (r);

}

//求给定字符串的子串

seqstring \*substring(seqstring \*S,int i,int len)

{

int k;

seqstring \*r;

//处理非法情况

if(i < 1 || i > S->length || i + len - 1 > S->length)

{

printf("错误！\n");

return (NULL);

}

else

{

r = (seqstring \*) malloc(sizeof(seqstring));

for(k = 0;k < len;k ++)

r->str[k] = S->str[i + k - 1];

r->length =len;

r->str[r->length] = '\0';

}

return (r);

}

void display(seqstring \*p)

{

if(!p)

{

printf("该顺序储存的字符串为空！");

exit(0);

}

printf("%s",p->str);

printf("\n");

}

int strcompare(seqstring \*st1,seqstring \*st2)

{

int i = 0;

while(st1->str[i] == st2->str[i])

i ++;

if(i == st1->length && i == st2->length)

return 0;

//是在st1->str[i] ！= st2->str[i]的情况下

else if(st1->str[i] > st2->str[i] || st1->length > st2->length)

return 1;

else

return -1;

}

int main()

{

int m = 0,n = 0;

int i = 0,j = 0;

seqstring \*st1,\*st2,\*st3;

st1 = create();

printf("创建的字符串为：\n");

display(st1);

printf("\n");

printf("输入想要插入的字符串");

printf("\n");

st2 = create();

printf("请输入想要插入的位置：\n");

scanf("%d",&m);

st1 = strinsert(st1,m,st2);

printf("在字符串st1的%d位置上插入st2后的字符串为：\n",m);

display(st1);

printf("\n");

printf("请输入想要删除的位置以及字符串长度:\n");

scanf("%d%d",&i,&j);

st1 = strdelete(st1,i,j);

printf("删除%d位置上%d长度的字符串后的字符串:\n",i,j);

display(st1);

printf("\n");

printf("输入一个想要连接的字符串\n");

st3 = create();

st1 = strconcat(st1,st3);

printf("将st3连接到st1后面后的字符串为：\n");

display(st1);

printf("\n");

printf("请输入想要截取的字符串位置以及长度:\n");

i = 0,j = 0;

scanf("%d%d",&i,&j);

st1 = substring(st1,i,j);

printf("截取字符串第%d位置上长度为%d的子串:\n",i,j);

display(st1);

printf("\n");

printf("请输入想要跟该字符串比较的字符串\n");

st3 = create();

printf("结果：\n");

display(st1);

display(st3);

if(strcompare(st1,st3) == 0)

printf("两个字符串相等\n");

else if(strcompare(st1,st3) == 1)

printf("第一个字符串大于第二个字符串\n");

else if(strcompare(st1,st3) == -1)

printf("第一个字符串小于第二个字符串\n");

else

printf("error");

return 0;

}

结果演示：

