**1．试设计一个算法：将子串T插入主串S第pos个位置上。**

解答：

void StrInsert(SString S, int pos, SString T){

//初始条件：串S和T存在

//若子串T完全插入主串S，则给出“完全插入”信息；若部分插入，则给出“部分插入”信息

if (pos < 1 || pos > S[0] + 1)

   Error(“Position Error!”);

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

{ // 完全插入

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

S[i+T[0]] = S[i]; //主串S给子串T腾地

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

S[i] = T[i-pos+1]; //子串T插入主串S原第pos个字符之前

S[0] = S[0] + T[0]; //修改主串S长度

cout<<”完全插入” ;

}

else

{ // 部分插入

for (i=MAXSTRLEN; i<=pos; i--) //主串S给子串T腾地

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

for (i=pos; i<pos+T[0]; i++) //子串T部分插入

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

S[0] = MAXSTRLEN; //修改主串S长度为最大串长

cout<<”部分插入” ;

}

}

1. **试设计一个算法：删除串S中第pos个字符开始的t个字符。**

解答：

void StrDelete(SString S, int pos, int t){

//在串S中删除自第pos个字符开始的t个字符

if(pos < 1 || pos > S.length || t < 0) {

Error(“Position Error!”);

}

if(pos+t-1 > S.length) //若t较大，则从第pos个字符起，删除到串尾

t=S.length-pos+1;

for(i=pos+t-1;i<S.length;i++)

S.ch[i++]=S.ch[i];

S.length=S.length-t;

}