

字符串运算器

要求

1. 不得使用字符串类库函数；
2. 字符串的输入输出需要自主编写函数，使用scanf("%c", &x)、printf("%c", x)实现，不可调用scanf("%s", s)、gets()等函数；
3. 需要完成两个版本，分别用数组和指针完成相关操作；
4. 字符串需要以'\0'作为结尾；
5. 程序可循环输入，直至输入0结束程序，每次输入前均显示菜单；
6. 每次输入一个字符串之后自动分配一个编号，编号从1开始连续分配；
7. 采用二维字符数组存放输入字符串和操作结果；
8. 每个涉及到输入编号或者下标的操作都要进行输入检查，如果超出了范围需要打印信息；
9. 其余细节参见参考样例。

菜单

0. 退出程序
1. 输入字符串
2. 显示全部字符串
3. 输入编号显示对应的字符串
4. 输入编号求串长
5. 输入编号s、t，将子串t连接到主串s之后，子串t不变；
6. 输入编号s、t，输出两个字符串的字典序比较结果；
7. 输入编号s，复制字符串s并加入到二维数组中；
8. 输入编号s、t、下标pos，将子串t插入到主串s的pos下标之后，子串t不变；
9. 输入编号s、下标pos，求主串s中从pos位置开始连续n个字符构成的子串sub，输出sub并加入到二维数组中；
10. 输入编号s、t，输出子串t在主串s中出现的位置下标。

函数原型

请勿在以下参考函数原型上增加新的参数，比如字符串的长度

- 使用数组实现：

```

void Input(char s[]);
void Print(char s[]);
int Length(char s[]);
void Connect(char s[], char t[]); // 串连接
int Compare(char s[], char t[]); // 串比较
void Copy(char s[], char t[]); // 串复制
void Insert (char s[], int pos, char t[]) // 串插入
void SubString(char s[], int pos, int n, char sub[]) // 求子串
int Search(char s[], char t[]) // 串查找

```

- 使用指针实现：

```

void Input(char *s);
void Print(char *s);
int Length(char *s);
void Connect(char *s, char *t); // 串连接
int Compare(char *s, char *t); // 串比较
void Copy(char *s, char *t); // 串复制
void Insert (char *s, int pos, char *t) // 串插入
void SubString(char *s, int pos, int n, char *sub) // 求子串
int Search(char *s, char *t) // 串查找

```

选做部分

串查找的优化：目前程序中找到一个相同的子串就返回，修改为找到所有的相同子串。此外，当前使用的朴素算法效率很低，请自己学习更高效的串匹配算法并实现，例如KMP算法。