字符串运算器

要求：

1. 不得使用字符串类库函数

2. 尽量用指针完成对字符串的操作

3. 采用两维字符数组存放若干个样本字符串和操作结果:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | b | a | b | y | \0 |  |  |  |  |  |
| 1 | g | i | r | l | \0 |  |  |  |  |  |
| 2 | b | o | y | \0 |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |
| 4 | m | a | t | h | e | r | \0 |  |  |  |
| 5 | f | a | t | h | e | r | \0 |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |
| 7 | d | a | u | g | h | t | e | r | \0 |  |
| 8 |  |  |  |  |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |  |  |  |  |

菜单：

1. 输入字符串

2. 显示字符串

3. 求串长

4. 串连接

5. 串比较

6. 串复制

7. 串插入

注：void Insert (char s[], int pos, char t[ ])

//将子串t插到主串s的pos下标之前

8. 求子串

注：void SubString(char s[ ], int pos, int n, char sub[ ])

//求主串s中从pos位置开始连续n个字符构成的子串sub

9. 串查找

注：int Search( char s[ ], char t[ ])

//返回子串t在主串s中出现的位置下标，若不存在返回-1

10. 串置换(选做)

注：void Replace(char s[ ], char v[ ], char t[ ])

//将主串s中出现的不重叠的v串用t串置换