/\*翻转字符串的某一段字符\*/

void Reverse1(char \*str, int i, int j) {

char ch;

while (i < j) {

ch = str[i];

str[i] = str[j];

str[j] = ch;

++i;

--j;

}

}

/\*abcdefg ----> cdefgab\*/

/\*ba gfedc , cdefgab\*/

bool Reverse(char \*str, int n) {

if (str == NULL)

return false;

int len = strlen(str);

if (n > len || n <= 0)

return false;

Reverse1(str, 0, n - 1);

Reverse1(str, n, len - 1);

Reverse1(str, 0, len - 1);

}

/\*abcdefg ----> cdefgba\*/

/\*ab gfedc , cdefgba\*/

bool Reverse(char \*str, int n) {

if (str == NULL)

return false;

int len = strlen(str);

if (n > len || n <= 0)

return false;

Reverse1(str, n, len - 1);

Reverse1(str, 0, len - 1);

}

void Swap(char &ch1, char &ch2) {

char ch;

ch = ch1;

ch1 = ch2;

ch2 = ch;

}

/\* “I love my students.” 🡪 “students. my love I” \*/

bool Reserve(char \*str) {

if (str == NULL)

return false;

char \*pch;//临时保存一个单词后的单词的首地址

/\*翻转整个句子\*/

int len = strlen(str);

char \*p = str;

char \*q = str + len - 1;

while (p < q) {

Swap(\*p, \*q);

++p;

--q;

}

/\*逐个翻转句子中的每个单词\*/

p = str;

q = str;

while (\*q != ' '&& \*q != '\0') {

++q;

if (\*q == ' ') {

pch = q + 1;

--q;

while (p < q) {

Swap(\*p, \*q);

++p;

--q;

}

/\*获取下一个要翻转的单词的首地址\*/

p = pch;

q = pch;

}

/\*句子的最后一个单词，翻转后结束函数\*/

if (\*q == '\0') {

--q;

while (p < q) {

Swap(\*p, \*q);

++p;

--q;

}

return true;

}

}

}