



1.

//strcpy()会把尾巴的\0随之拷贝

void mystrcpy(char \*dest,char \*sour) {

int i = 0;

while (\*(sour + i)) {

//printf("%c\n", \*(sour+i));

dest[i] = sour[i];

i++;

}

dest[i] = '\0';

}

//简化一下，小于返回-1，等于0，大于1

int mystrcmp(const char \*arr1,const char \*arr2) {

int i = 0;

while (arr1[i] || arr2[i]) {

if (arr1[i] < arr2[i]) {

return -1;

}

else if(arr1[i]>arr2[i]) {

return 1;

}

i++;

}

return 0;

}

void mystrcat(char \*des,const char \*src) {

char\* ptr;

ptr = des;

while (\*ptr) {

ptr++;

}

while (\*src) {

\*ptr++ = \*src++;

}

\*ptr = '\0';

}

int mystrlen(const char\* str) {

int length = 0;

while (\*str) {

length++;

str++;

}

return length;

}

2.

//处理思想，记录当前字符为单词但下一个是空格的位置，count++，并特殊处理末尾有可能的空格

void word\_count(const char\* str) {

int count = 0;

while (\*str !='\n') {

if (\*str != ' ' && \*(str+1) == ' ') {

count++;

}

str++;

}

if (\*--str != ' ') {

count++;

}

printf("%d\n", count);

}

3.

void word\_output(const char\* str) {

while (\*str != '\n') {

if (\*str != ' ') {

printf("%c", \*str);

}

else if (\*str == ' ' && \*(str + 1) != ' ') {

printf("\n");

}

str++;

}

}

4.

//字符串反转，单词不反转

void sent\_reverse(char\* str) {

int length = strlen(str);//包括了最后的\n在内

char\* p = &str[length - 2];

char\* ptr = str;

while (ptr < p) {

char temp = \*ptr;

\*ptr = \*p;

\*p = temp;

p--; ptr++;

}

ptr = str;

while (\*str != '\n') {

if (\*str != ' ' && ( \* (str + 1) == ' ' || \*(str+1) == '\n')) {

p = str+1;

while (ptr <str && \*ptr !=' ') {

char temp = \*ptr;

\*ptr = \*str;

\*str = temp;

str--; ptr++;

}

str = p;

ptr = p;

}

else if (\*str != ' ' && \*(str-1) == ' ') {

ptr = str;

str++;

}

else {

str++;

}

}

}