* 1. mystrcpy

void mystrcpy(char\* dest, const char\* src) {

int len;

len = strlen(src);

int i = 0;

while(i<=len)

{

dest[i] = src[i];

i++;

}

}



int main() //2 输出单词个数

{

char c[20],dest[20];

gets(c);

mystrcpy(dest, c);

printf("dest: %s", dest);

}

1.2 my\_strcmp();

int my\_strcmp(const char\* a, const char\* b)

{

int i,N=5;

for (i = 0; i < N; i++)

{

if (a[i] == b[i])continue;

else if (a[i] > b[i]) { printf("%d",1); return 0; }

else if (a[i] < b[i]) { printf("%d",-1); return 0; }

}

printf("%d", 0);

return 0;

}

int main()

{

char a[20], b[20]; // 不知道输入，那么数组的空间应该设置多大？

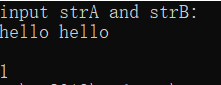
printf("input strA and strB:\n");

gets(a);

gets(b);

my\_strcmp(a, b);

}



1.3 mystrcat()

void mystrcat(char \*s1,const char \*s2)

{

if (s1 == NULL || s2 == NULL)

perror("error");

int i = 0, j = 0;

while (s1[i] != '\0')

i++;

while ((s1[i++] = s2[j++]) != '\0');

}

int main()

{

char s1[50],s2[50];

scanf("%s%s", &s1,&s2);

//rewind(stdin);

mystrcat(s1,s2);

printf("cat后 %s\n",s1);

return 0;

}



1.4 mystrlen()

int mystrlen(char\* s1) //1.4

{

int i = 0;

while (s1[i] != '\0') {

i++;

}

return i;

}

int main()

{

char s1[50];

scanf("%s", &s1);

printf("len is : %d\n", mystrlen(s1));

return 0;

}



2.1 输入一行字符串，输出该行单词个数。

int main()

{

char\* split = "\_ ";

char\* res = NULL;

char c[20];

int count = 0;

gets(c);

//char str[200] = "\_\_\_\_hello\_\_\_\_\_\_\_\_\_world\_ how\_\_\_are\_\_\_you\_\_\_\n";

res = strtok(c, split);

while (res)

{

//printf("%s\n", res);

res = strtok(NULL, split);

count++;

}

printf("%d", count);

}



3 输入一行字符串（单词和若干空格），输出该行单词（每个单词一行）

int main() //3. 输入一行字符串（单词和若干空格），输出该行单词（每个单词一行）

{

char\* split = "\_ ";

char\* res = NULL;

char c[20];

int count = 0;

gets(c);

res = strtok(c, split);

while (res)

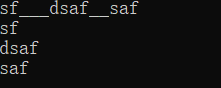
{

printf("%s\n", res);

res = strtok(NULL, split);

}

}



4 输入一行字符串，把字符串翻转

void inverse(char\* str)//字符串逆置

{

int len = strlen(str);

char\* p1 = str;

char\* p2 = str + len - 1;

while (p1 < p2)

{

char tmp = \*p1;

\*p1 = \*p2;

\*p2 = tmp;

p1++;

p2--;

}

}

int main()

{

char c[20]="I\_\_\_\_am\_\_student";

int s1[20] = { 0 };

int s2[20] = { 0 };

//gets(c);

inverse(c);

printf("现在的句子是： %s\n", c);

char\* split = "\_ ";

char\* res = NULL;

res = strtok(c, split);

int i = 0;

while (res)

{

inverse(res);

printf("%s ",res);

res = strtok(NULL, split);

}

return 0;

}

