字符串

信息安全、法学　梅骏逸

有错误请及时指出🙏
有错误请及时指出🙏
有错误请及时指出🙏

基本操作

```c
char* strcpy(char* dest, const char* src);
// 拷贝 `src` 中的字符（包括结束符）到`dest` 中.
char* strncpy(char* dest, const char* src, size_t count);
/* 拷贝 `count` 个 `src` 中的字符（包括结束符）到 `dest`中.
   若 `count` 个字符拷贝完后仍未到 `src` 结尾，那就没有结束符
   若拷贝完 `src`，仍未达到 `count` 个字符，多个 `'\0'` 会被添加到后面 */
char* strcat(char* dest, const char* src);
// 将 `src` 加到 `dest` 尾部。
char* strncat(char* dest, const char *src, size_t count)
// 将 `src` 中 `count` 个字符添加到 `dest` 尾部。
size_t strlen(const char* str); //字符串长度。
int strcmp(const char* lhs, const char* rhs);
// 若没有不同返回 `0`，有不同就是正数或负数。
int strncmp(const char *lhs, const char *rhs, size_t count);
char* strchr(const char* str, int ch);
// `ch` 第一次出现的地址
char* strrchr(const char* str, int ch);
// `ch` 最后一次出现的地址
char* strstr (char* str1, char* str2);
// `str1` 中第一次出现子串 `str2` 的地址
```

```c
isalpha()
// 检查是否为字母字符
isupper()
// 检查是否为大写字母字符
islower()
// 检查是否为小写字母字符
isdigit()
// 检查是否为数字
isxdigit()
// 检查是否为十六进制数字表示的有效字
isspace()
// 检查是否为空格类型字符
iscntrl()
// 检查是否为控制字符
ispunct()
// 检查是否为标点符号
isalnum()
// 检查是否为字母和数字
isprint()
// 检查是否是可打印字符
isgraph()
// 检查是否是图形字符
// 等效于 `isalnum() || ispunct()`
```

小写转大写

输入样例
GoodMorning

解题思路
遍历字符串中每一个字符，对每一个字符 a[i] 执行 a[i] = a[i] - ’a’ + ’A’;

输出样例
GOODMORNING

大写	ASCII	小写	ASCII
A	65	a	97
B	66	b	98
C	67	c	99
……	……	……	……
W	87	w	119
X	88	x	120
Y	89	y	121
Z	90	z	122

```cpp
#include <cstdio>
#include <cstring>
#include <iostream>

using namespace std;

int main() {
  char a[20];
  cin >> a;
  int length = strlen(a);

  for (int i = 0; i < length; i++) {
    if (a[i] >= 'a' && a[i] <= 'z') {
      a[i] = a[i] - 'a' + 'A';
    }
  }

  cout << a << endl;
  return 0;
}
```

统计单词数量

问题描述
输入一个字符串，统计其中的单词数量。

输入样例
```
Hello    world  Nankai   test
```

输出样例
4

解题思路
- 遇到新单词起始字母，单词数加一
- 上一个字符是空格，当前字符是字母表明遇到新单词。
- `flag==false` → 上一个字符不是字母
- `flag==True` → 上一个字符是字母

`isalpha(str[i]) && !flag` 的作用：当前字符是字母且不属于已经统计过的单词时，计数器 `cnt` 加一。

```cpp
#include <iostream>
#include <cstdio>
#include <cstring>

using namespace std;

int main() {
  char str[100001];
  gets(str);
  int length = strlen(str);
  bool flag = false;
  int cnt = 0;

  for (int i = 0; i < length; i++) {
    if (isalpha(str[i]) && !flag) { flag = true; cnt++; }
    else if (str[i] == ' ') { flag = false; }
  }

  cout << cnt << endl;
  return 0;
}
```

最长的单词

问题描述
从键盘输入一个字符串，找到字符串中最长的一个单词并输出(单词以空格分隔，可能有多个空格)，如果有多个最长单词，输出最前面的一个。

输入样例

```
This is    examination
```

输出样例

```
examination
```

解题思路
遍历字符串中的单词，记录最长单词的起始位置和长度用于输出。

```cpp
#include <iostream>
#include <cstdio>
#include <cstring>

using namespace std;

int main() {
  char str[100001];
  gets(str);

  int curr_len, curr_start, max_start;
  int idx = 0;
  int max_len = 0;
  int length = strlen(str);

  while (idx < length) {
    while (str[idx] == ' ' && idx < length) idx++;
    curr_start = idx;
    while (str[idx] != ' ' && idx < length) idx++;
    curr_len = idx - curr_start;
    if (curr_len > max_len) { max_len = curr_len; max_start = curr_start; }
  }

  for (int i = max_start; i < max_start + max_len; i++) { putchar(str[i]); }
  return 0;
}
```

字符串单词翻转

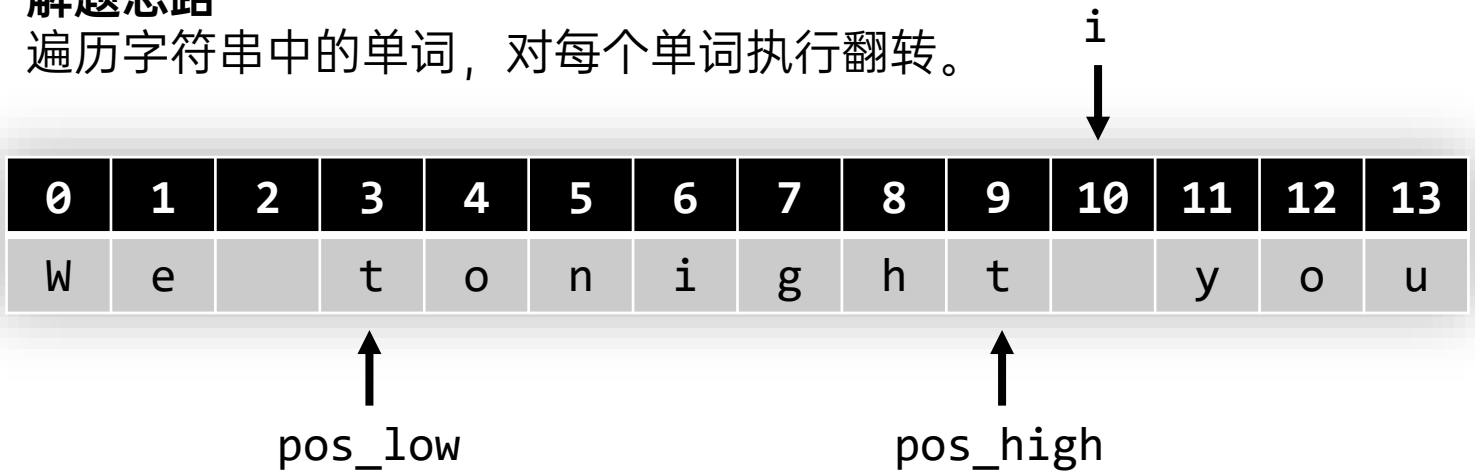
问题描述

从键盘输入一个字符串存入数组，将字符串反转（保持单词顺序不变，每个单词翻转），单词以空格隔开，输出的字符串保持空格位置不变。

输入样例	输出样例
We tonight you	eW thginot uoy

解题思路

遍历字符串中的单词，对每个单词执行翻转。




```cpp
#include <iostream>
#include <cstdio>
#include <cstring>

using namespace std;

int main() {
  char str[100001];
  gets(str);

  char tmp;
  int pos_low, pos_high;
  int length = strlen(str);
  int i = 0;

  while (i < length) {
    while (str[i] == ' ' && i < length) i++;
    pos_low = i;
    while (str[i] != ' ' && i < length) i++;
    pos_high = i - 1;

    while (pos_low < pos_high) {
      tmp = str[pos_low];
      str[pos_low] = str[pos_high];
      str[pos_high] = tmp;
      pos_low++;
      pos_high--;
    }
  }

  cout << str;
  return 0;
}
```

字符串右移

从键盘输入一个字符串，输入一个整数 m，将字符串循环右移 m 位（m 可能大于字符串长度）

0	1	2	3	4	5	6		7	8
d	i	o	p	H	e	g			
e	g	d	i	o	p	H		e	g

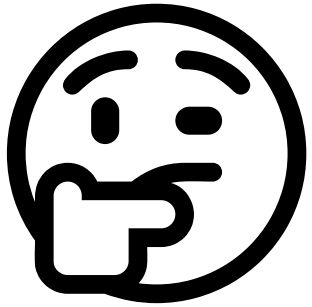
循环右移（0 为起始位）：i’ = (i + m) % n

1	2	3	4	5	6	7		8	9
d	i	o	p	H	e	g			
e	g	d	i	o	p	H		e	g

循环右移（1 为起始位）：i’ = (i + m - 1) % n + 1

```cpp
#include <iostream>
#include <cstdio>
#include <cstring>

using namespace std;

int main() {
  char a[10001];
  char b[10001];

  int n, i_, m;

  cin >> a;
  cin >> m;

  n = strlen(a);

  for (int i = 0; i < n; i++) {
    i_ = (i + m) % n;
    b[i_] = a[i];
  }
  b[n] = '\0';

  cout << b;

  return 0;
}
```

思考右移的更优方法
以及左移的有关方法



凯撒密码

问题描述

Julius Caesar 生活在充满危险和阴谋的年代。为了生存，他首次发明了密码，用于军队的消息传递。假设你是 Caesar 军团中的一名军官，需要把 Caesar 发送的消息破译出来、并提供给你的将军。消息加密的办法是：对消息原文中的每个字母，分别用该字母之后的第5个字母替换（例如：消息原文中的每个字母A 都分别替换成字母F），其他字符不变，并且消息原文的所有字母都是大写的。密码中的字母与原文中的字母对应关系如下

密码	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
原码	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U

输入 Caesar 发出的加密消息

输出 Caesar 发出的原始消息。

输入样例

```
NS BFW, JAJSYX TK NRUTWYFSHJ FWJ YMJ WJXZQY TK YWNANFQ HFZXJX
```

输出样例

```
IN WAR, EVENTS OF IMPORTANCE ARE THE RESULT OF TRIVIAL CAUSES
```

解密实现好了
可以试试看实现加密

进阶题：

[W 密码 (POJ 1107)]

```cpp
#include <cstdio>
#include <cstring>

int main() {
  char s[201];
  gets(s);
  int length = strlen(s);

  for (int i = 0; i < length; i++) {
    if (s[i] < 'A' || s[i] > 'Z') continue;
    int c = s[i] - 'A' + 1;
    c = (c + 21) % 26;
    s[i] = c + 'A' - 1;
  }

  printf("%s\n", s);
  return 0;
}
```

自选习题

洛谷字符串题单
OpenJudge 百练 2744 子串
洛谷 P1055 [NOIP2008 普及组] ISBN 号码
（以下可尝试）
洛谷 P1026 [NOIP2001 提高组] 统计单词个数
洛谷 P1054 [NOIP2005 提高组] 等价表达式
（以下可自学）
洛谷 P3805 【模板】Manacher 算法
洛谷 P3375 【模板】KMP字符串匹配

参考

cppreference
C++ Reference

谢谢！