1. 回文串

解题思路：if str[i] !=str[len-1-i] return false;

1. 说反话

解题思路：二维数组存储

1. 换个格式输出整数

解题思路：

1. 个位数统计

解题思路： ‘R’ - ‘0’ -> 真正的数，同理 大小写转换 大 + ‘a’ -’A’ -> 小

1. 查验身份证

解题思路：字符数组存储，对前17位中有非数字字符的串直接输出

1. 写出这个数

解题思路：使用 sprintf() d-(%d)>s

1. 福尔摩斯的约会

解题思路：

1. 科学计数法

解题思路：关键是‘E’的位置，

指数为 0 需要特判

指数为负，需要计算出小数点后0的个数

指数为正，需要根据指数的大小判断小数点的位置或者没有小数点需要补0的个数问题。

1. 数字加密

解题思路：首先反转字符串，取两个较长的字符串，循环遍历，如果超出字符串A/B的长度，取做 0 ，然后计算。

1. Format A+B

解题思路：xx,xxx,xxx格式

先输出len%3, 如果mod>0 && len>3 输出‘，’，然后每三位输出一个‘，’，最后一个不输出，。

1. Spell It Right

解题思路：Sprintf()

1. Passworld

解题思路：使用结构体

注意字符要写对

1. Kuchiguse

解题思路：输入整数后，在使用 gets()前使用 getchar()接收回车

PAT不支持gets()

可以使用fgets()代替，但是注意，fgets函数包含回车，而gets()不包含。

1. Read Numbers in Chinese

解题思路：

#include<stdio.h>

#include<string.h>

int main()

{

char num[10][5] = {"ling","yi","er", "san","si","wu","liu","qi","ba","jiu"};

char wei[5][5] = {"Shi","Bai","Qian","Wan","Yi"};

char str[20];

scanf("%s", str);

int len = strlen(str);

int left = 0;

int right = len-1;

if(str[0] == '-'){

printf("Fu");

left++;

}

while(left+4<=right)

{

right -= 4;

}

while(left<len) // 每次处理一节

{

bool flag = false; // 表示是否有累积的0

bool isPrint = false; // 表示该节中是否输出过

while(left<=right) // 处理该节中的每一位

{

if(left>0&&str[left]=='0')

{

flag = true;

}else{ // 当前为 ！=0

if(flag == true){ // 存在累积的 0

printf(" ling");

flag = false;

}

// 只要不是首位，前输出空格

if(left >0)

{

printf(" ");

}

printf("%s", num[str[left] - '0']);

isPrint = true;

if(left != right){ // 本节中处了个位， 输出shibaiqian

printf(" %s", wei[right-left-1]);

}

}

left++;

}

if(isPrint == true && right !=len-1) // 只要不是个位节，就输出wanYI

{

printf(" %s", wei[(len-1-right)/4+2]);

}

right+=4;

}

return 0;

}