



```
1 //第一题:用栈实现字符串逆序
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <string.h>
5
6 #define MAX_SIZE 100
7
8 typedef struct {
9     int data[MAX_SIZE];
10    int top;
11 } Stack;
12
13 void initStack(Stack* stack) {
14     stack->top = -1;
15 }
16
17 int isFull(Stack* stack) {
18     return stack->top == MAX_SIZE - 1;
19 }
20
21 int isEmpty(Stack* stack) {
22     return stack->top == -1;
23 }
24
25 int push(Stack* stack, int value) {
26     if (isFull(stack)) {
27         return -1;
28     }
29     stack->data[++(stack->top)] = value;
30     return 0;
31 }
32
33 int pop(Stack* stack, int* value) {
34     if (isEmpty(stack)) {
35         return -1;
36     }
37     *value = stack->data[(stack->top)--];
38     return 0;
39 }
40
41 char peek(Stack* stack, char value) {
42     if (isEmpty(stack)) {
43         return '\0';
44     }
45     return stack->data[stack->top];
46 }
47
48 int main(){
49     Stack s;
50     char input[100];
51
52     initStack(&s);
53
54     fgets(input, sizeof(input), stdin);
55
56     int len = strlen(input);
57     if (len > 0 && input[len - 1] == '\n') {
58         input[len - 1] = '\0';
59         len--;
60     }
61
62     for(int i = 0; i < len; i++){
63         push(&s, input[i]);
64     }
65
66     while(!isEmpty(&s)){
67         char ch;
68         pop(&s, (int*)&ch);
69         putchar(ch);
70     }
71
72     return 0;
73 }
74
75 //第二题:冰雹猜想
76 int countSteps(int n){
77     if(n == 1) return 0;
78
79     if (n % 2 == 0) {
80         return 1 + countSteps(n / 2);
81     }else {
82         return 1 + countSteps(3*n + 1);
83     }
84 }
85
86 int main(){
87     int n;
88     scanf("%d", &n);
89
90     int steps = countSteps(n);
91     printf("%d\n", steps);
92     return 0;
93 }
```