

试数详细步骤

- 1、把自己当成计算机，计算机怎么执行，自己就怎么写；
- 2、不要把某一步漏下，计算机怎么执行，自己就怎么写；
- 3、写详细，格式写对；

```
1  /*
2     2020年12月14日
3     强制类型转换
4  */
5  #include <stdio.h>
6  int main(void)
7  {
8      int i;
9      float sum = 0;
10
11     for (i=1; i<=100; i++)
12     {
13         sum = sum + 1.0/i; //是ok的, 推荐使用
14         //sum = sum + (float)(1/i); 这样写是不对的: 因为(1/i)的值一直是0(两个整型)
15         //也可以这样写: sum = sum + 1/(float)(i); (float)(...)将后面括号内的值...强制转化为浮点型; 不推荐
16     }
17
18     printf("分数相加的值为: %f\n", sum); //float必须用%f输出
19
20     return 0;
21 }
```

- ①、 $i=1$ $1 \leq 100$ $sum=0+1/1.0$ $i=2$
- ②、 $i=2$ $2 \leq 100$ $sum=0+1/1.0+1/2.0$ $i=3$
- ③、 $i=3$ $3 \leq 100$ $sum=0+1/1.0+1/2.0+1/3.0$ $i=4$

如何看懂一个程序

一些常见的需要算法知识的 C 语言程序：

判断一个数字是否是素数

判断一个数字是否是回文数

编程实现求一个十进制数字的二进制形式

求一个数字的每位是奇数的数字取出来组合形成的新数字

求一个数字倒过来的数字

如何看懂一个程序,分三步：

1. 流程

2. 每个语句的功能

3. 试数

如何学习一些需要算法的程序【如何掌握一个程序】

1. 尝试自己去编程解决它

但要意识到大部分人都是自己无法解决的，这时不要气馁，也不要自卑，如果十五分钟还想不出来，此时我建议您就可以看答案了

2. 如果解决不了，就看答案

关键是把答案看懂，这个要花很大的精力，也是我们学习的重点，看懂一个程序要分三步：流程、每个语句的功能、试数

3. 看懂之后尝试自己去修改程序，并且知道修改之后程序的输出结果的含义

不建议看懂程序之后就立即自己敲程序

4. 照着答案去敲

5. 调试错误

6. 不看答案，自己独立把答案敲出来

7. 如果程序实在无法彻底理解，就把它背会，不过无法彻底理解的程序非常少，我自己几乎没有碰到过