函数与递归

可能是最后一课

Garen-Wang

2018 年 6 月 24 日

目录

函数介绍

函数介绍

函数 (function)，又名子程序。在 C 语言中，子程序的作用是由一个主函数和若干个函数构成。由主函数调用其他函数，其他函数也可以互相调用。同一个函数可以被一个或多个函数调用任意多次。

函数介绍

函数 (function)，又名子程序。在 C 语言中，子程序的作用是由一个主函数和若干个函数构成。由主函数调用其他函数，其他函数也可以互相调用。同一个函数可以被一个或多个函数调用任意多次。
函数的基本语法构成由：返回类型 + 函数名 + 参数 + 大括号

函数介绍

函数 (function)，又名子程序。在 C 语言中，子程序的作用是由一个主函数和若干个函数构成。由主函数调用其他函数，其他函数也可以互相调用。同一个函数可以被一个或多个函数调用任意多次。
函数的基本语法构成由：返回类型 + 函数名 + 参数 + 大括号
函数的实现意义是用来进行专一化处理问题，而不是一个 main 函数写满了所有基础操作。

函数介绍

函数 (function)，又名子程序。在 C 语言中，子程序的作用是由一个主函数和若干个函数构成。由主函数调用其他函数，其他函数也可以互相调用。同一个函数可以被一个或多个函数调用任意多次。
函数的基本语法构成由：返回类型 + 函数名 + 参数 + 大括号
函数的实现意义是用来进行专一化处理问题，而不是一个 main 函数写满了所有基础操作。
使用函数可以更简单明了地解决问题，有时还能办到一些神奇的事情。

常见函数应用

常见函数应用

函数的应用太广泛了，所以下面的只是一些题目的操作。

常见函数应用

函数的应用太广泛了，所以下面的只是一些题目的操作。

- 维护进制转换操作
- 判断一个字符串是不是回文串
- 求出一个函数（数学的）的值
- 交换两个数
- 求两个数的最大公约数

常见函数应用

函数的应用太广泛了，所以下面的只是一些题目的操作。

- 维护进制转换操作
- 判断一个字符串是不是回文串
- 求出一个函数（数学的）的值
- 交换两个数
- 求两个数的最大公约数

等等等等。。。

P1149 火柴棒等式

P1149 火柴棒等式

题目要求用 n 根火柴摆出类似 $A+B=C$ 的等式，询问等式种类。注意数字可以是多位数。$n \leq 24$。

P1149 火柴棒等式

题目要求用 n 根火柴摆出类似 $A+B=C$ 的等式，询问等式种类。注意数字可以是多位数。$n \leq 24$。
加号和等号都需要 2 根火柴，所以先减掉 4，剩下的就是枚举三个数了。

P1149 火柴棒等式

题目要求用 n 根火柴摆出类似 $A+B=C$ 的等式，询问等式种类。注意数字可以是多位数。$n \leq 24$。
加号和等号都需要 2 根火柴，所以先减掉 4，剩下的就是枚举三个数了。而我们也不用枚举三个数，因为加法的性质，只要确定两个，第三个就出来了。

P1149 火柴棒等式

题目要求用 n 根火柴摆出类似 $A+B=C$ 的等式，询问等式种类。注意数字可以是多位数。$n \leq 24$。
加号和等号都需要 2 根火柴，所以先减掉 4，剩下的就是枚举三个数了。而我们也不用枚举三个数，因为加法的性质，只要确定两个，第三个就出来了。
所以最重要的是判断一个数由几根火柴构成。而实现这个功能，可以使用函数封装起来。

P1149 火柴棒等式

题目要求用 n 根火柴摆出类似 $A+B=C$ 的等式，询问等式种类。注意数字可以是多位数。$n \leq 24$。
加号和等号都需要 2 根火柴，所以先减掉 4，剩下的就是枚举三个数了。而我们也不用枚举三个数，因为加法的性质，只要确定两个，第三个就出来了。
所以最重要的是判断一个数由几根火柴构成。而实现这个功能，可以使用函数封装起来。
最高不超过 1111，使用火柴数最少并且数字最大的数就是它了。

P1149 火柴棒等式

题目要求用 n 根火柴摆出类似 $A+B=C$ 的等式，询问等式种类。注意数字可以是多位数。$n \leq 24$。
加号和等号都需要 2 根火柴，所以先减掉 4，剩下的就是枚举三个数了。而我们也不用枚举三个数，因为加法的性质，只要确定两个，第三个就出来了。
所以最重要的是判断一个数由几根火柴构成。而实现这个功能，可以使用函数封装起来。
最高不超过 1111，使用火柴数最少并且数字最大的数就是它了。
所以一个 $O(1111^2)$ 的双重 for 循环就能够解决问题。

递归

递归

递归 (recursion) 是函数的更深层应用，因为函数允许嵌套使用，甚至可以使用自己！

递归

递归 (recursion) 是函数的更深层应用，因为函数允许嵌套使用，甚至可以使用自己！
常见的递归应用方法有：

递归

递归 (recursion) 是函数的更深层应用，因为函数允许嵌套使用，甚至可以使用自己！
常见的递归应用方法有：

- dfs（深度优先搜索或遍历）
- bfs（宽度优先搜过或遍历）
- 回溯法
- 减小问题范围（递推的反过程）

递归

递归 (recursion) 是函数的更深层应用，因为函数允许嵌套使用，甚至可以使用自己！
常见的递归应用方法有：

- dfs（深度优先搜索或遍历）
- bfs（宽度优先搜过或遍历）
- 回溯法
- 减小问题范围（递推的反过程）

时间不够，只能够挑一个讲，我们讲回溯法。

回溯法

回溯法

回溯法是一种搜索，相比于传统 dfs 的不同之处是回溯了。

回溯法

回溯法是一种搜索，相比于传统 dfs 的不同之处是回溯了。
虽然回溯之后会适当减缓速度，但是也并不慢。

回溯法

回溯法是一种搜索，相比于传统 dfs 的不同之处是回溯了。
虽然回溯之后会适当减缓速度，但是也并不慢。
适用于全方位搜索答案的问题。

回溯法

回溯法是一种搜索，相比于传统 dfs 的不同之处是回溯了。
虽然回溯之后会适当减缓速度，但是也并不慢。
适用于全方位搜索答案的问题。
让我去网上找回溯法的框架。

回溯法

回溯法是一种搜索，相比于传统 dfs 的不同之处是回溯了。
虽然回溯之后会适当减缓速度，但是也并不慢。
适用于全方位搜索答案的问题。
让我去网上找回溯法的框架。
回溯法的经典例题是八皇后问题，时间关系我不讲。

P1036 选数

P1036 选数

题目要求你求出 n 个数取 k 个的排列情况中和为素数的种类数。

P1036 选数

题目要求你求出 n 个数取 k 个的排列情况中和为素数的种类数。
显然情况种类数共有 C_n^k 种。但是求出这个东西并没什么卵用。

P1036 选数

题目要求你求出 n 个数取 k 个的排列情况中和为素数的种类数。
显然情况种类数共有 C_n^k 种。但是求出这个东西并没什么卵用。
而我们需要做的是模拟所有选择，然后得到和，再判断是否为质数。

P1036 选数

题目要求你求出 n 个数取 k 个的排列情况中和为素数的种类数。
显然情况种类数共有 C_n^k 种。但是求出这个东西并没什么卵用。
而我们需要做的是模拟所有选择，然后得到和，再判断是否为质数。
这道题我来亲手写一写。。。

后面的话

后面的话

How time flies! 高一就快结束了！

后面的话

How time flies! 高一就快结束了！
这个团队在下学期起头应该不会再进行维护了，在复赛后会重新招入新高一的团队成员。

后面的话

How time flies! 高一就快结束了！
这个团队在下学期起头应该不会再进行维护了，在复赛后会重新招入新高一的团队成员。
希望大家在这一年能够学到东西，也希望大家多多包涵我的不足之处。

后面的话

How time flies! 高一就快结束了！
这个团队在下学期起头应该不会再进行维护了，在复赛后会重新招入新高一的团队成员。
希望大家在这一年能够学到东西，也希望大家多多包涵我的不足之处。
进入高二后，仅存的团队成员可能只会有 3 到 4 位。

后面的话

How time flies! 高一就快结束了！
这个团队在下学期起头应该不会再进行维护了，在复赛后会重新招入新高一的团队成员。
希望大家在这一年能够学到东西，也希望大家多多包涵我的不足之处。
进入高二后，仅存的团队成员可能只会有 3 到 4 位。
最希望大家的事情是能够在 10 月的初赛好好发挥，一个月后公款去广州！（更好的是拿奖了啊！）

后面的话

How time flies! 高一就快结束了！
这个团队在下学期起头应该不会再进行维护了，在复赛后会重新招入新高一的团队成员。
希望大家在这一年能够学到东西，也希望大家多多包涵我的不足之处。
进入高二后，仅存的团队成员可能只会有 3 到 4 位。
最希望大家的事情是能够在 10 月的初赛好好发挥，一个月后公款去广州！（更好的是拿奖了啊！）
我也要去好好准备去招待高一了。。。

吐槽

吐槽

我肝这个课件的时候，CE 了一个小时。。。

吐槽

我肝这个课件的时候，CE 了一个小时。。。
想写句话来吐槽又 CE 了。。。

吐槽

我肝这个课件的时候，CE 了一个小时。。。
想写句话来吐槽又 CE 了。。。
啊！伟大的 LaTeX！

谢谢！

all made by LaTeX