Competition，准备

# 一．第一至三个月：建立基础

学习编程基础：掌握一门编程语言，建议选择C++、Java或Python。学习基本语法、数据类型、流程控制等基础知识。

熟悉算法和数据结构：学习常见的数据结构（如数组、链表、栈、队列、树、图等）和基本算法（如排序、查找、递归、动态规划等），理解它们的原理和实现方式。

练习题目：刷题是提高编程能力的有效方法，可以选择一些在线平台（如LeetCode、Codeforces、AtCoder等）进行刷题练习，逐步提高解题速度和技巧。

# 二．第四至六个月：深入学习

深入算法和数据结构：学习高级数据结构（如堆、树状数组、并查集等）和常见算法（如贪心算法、回溯算法、图算法等），掌握其应用场景和解题思路。

阅读经典教材：阅读《算法导论》等经典教材，深入理解算法和数据结构的基本原理，提高对编程问题的抽象和解决能力。

参加训练营或课程：参加ACM竞赛的培训营或在线课程，系统学习竞赛相关知识和技巧，通过导师指导和实战训练提高编程水平。

# 三．第七至九个月：刷题和模拟比赛

刷题练习：继续刷题练习，注重解决复杂问题和提高编程效率，逐步提高ACM竞赛所需的编程能力和算法水平。

模拟比赛：参加在线模拟比赛，模拟真实竞赛环境，提高应对竞赛压力和解题速度。

分析错题：对做错的题目进行深入分析，找出解题思路和方法的不足之处，及时调整学习方法和策略。

# 四．第十至十二个月：备战比赛

强化训练：重点训练ACM竞赛常见题型和技巧，注重团队合作和问题解决能力的培养，提高在竞赛中的表现水平。

模拟比赛：继续参加模拟比赛，逐步适应竞赛的节奏和规则，提高在有限时间内的解题效率和正确率。

团队合作：如果是以团队形式参加ACM竞赛，加强团队协作和配合，培养团队意识和沟通能力，提高整体竞赛实力。

最后冲刺：在比赛前进行最后的复习和练习，巩固知识点，调整状态，为比赛做好充分准备。

制定这样的计划并严格执行，相信你能在一年后参加ACM竞赛并取得理想成绩。记得在学习的过程中保持耐心和持续努力，加油！

# 经典书籍

1.《算法》（Algorithm） by Robert Sedgewick and Kevin Wayne： 这本书是一本经典的算法教材，以清晰简洁的风格介绍了各种基本的数据结构和算法，适合初学者入门。

2.《数据结构与算法分析：C语言描述》（Data Structures and Algorithm Analysis in C） by Mark Allen Weiss： 这本书介绍了数据结构和算法的基本原理和常用实现方式，以C语言为例，适合想要通过实践来理解算法的初学者。

3.《数据结构与算法（Java版）》（Data Structures and Algorithms in Java） by Robert Lafore： 这本书使用Java语言来讲解数据结构和算法的基本概念和实现方法，适合Java初学者入门。

4.《数据结构与算法分析：Java语言描述》（Data Structures and Algorithm Analysis in Java） by Mark Allen Weiss： 这本书以Java语言为例，介绍了数据结构和算法的基本原理和分析方法，适合Java编程初学者。

5.《算法竞赛入门经典》（Introduction to Algorithms Competitions） by 刘汝佳： 这本书主要面向初学者，以清晰简洁的语言介绍了常见的算法和解题技巧，适合准备参加算法竞赛的初学者。