

内 容 简 介

计算机程序设计能力考试 (PAT) 是客观评测程序与算法设计实现能力的标准化考试, 已有数百家企业将该考试的成绩作为衡量计算机技术岗位应聘人员的基础标准; 同时, 多所高等院校也将之作为招收研究生的重要参考指标。

本书全面介绍 PAT 的规则设置、各等级的考试大纲与评分标准等, 并简要介绍考试大纲涉及的相关知识点, 最后给出一套模拟样题的详细解析, 以帮助有兴趣参加考试的读者充分了解该考试, 为报考做好准备。

本书从最基础的程序设计开始介绍知识点, 可以作为“程序设计基础”“数据结构”“算法设计与分析”这三门课程的补充读物, 主要面向报考 PAT 乙级、甲级、顶级的高校在校生及在职人员。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。举报: 010-62782989, beiqinquan@tup.tsinghua.edu.cn。

图书在版编目 (CIP) 数据

计算机程序设计能力考试 (PAT) 备考通 / 陈越, 戴龙翱主编. —北京: 清华大学出版社, 2022.5

ISBN 978-7-302-60331-3

I. ①计… II. ①陈… ②戴… III. ①程序设计—资格考试—自学参考资料 IV. ①TP311.1

中国版本图书馆 CIP 数据核字 (2022) 第 043511 号

责任编辑: 郭 赛

封面设计:

责任校对: 焦丽丽

责任印制:

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座

邮 编: 100084

社总机: 010-83470000

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:

经 销: 全国新华书店

开 本: 170mm×230mm

印 张: 11.75

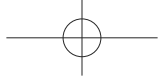
字 数: 163 千字

版 次: 2022 年 6 月第 1 版

印 次: 2022 年 6 月第 1 次印刷

定 价: 00.00 元

产品编号: 091027-01



前言

计算机程序设计能力考试 (PAT) 是客观评测程序与算法设计实现能力的标准化考试, 于 2011 年由浙江大学计算机科学与技术学院创办, 已经走过了十年历程, 从只有零星几家企业加盟、每场考试不足百人, 发展到成为数百家企业衡量计算机人才的基础标准、为数万名考生提供服务, PAT 正稳步走向成熟。

PAT 创办的初衷是破除“唯名校出身论”的职场招聘歧视, 旨在为广大学子提供一个公平、公正的求职起步平台。PAT 的联盟企业均承诺为达到分数线要求的考生提供招聘时的优先机会。

作为 PAT 及其支撑平台“拼题 A”的创始团队成员, 我们觉得有义务为广大考生提供一份备考指南, 从而向读者全面介绍 PAT, 为备考助力。

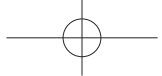
本书第 1 章介绍 PAT 的发展历史和规则设置, 并简介 PAT 的支撑平台“拼题 A”; 第 2 章介绍 PAT 考试大纲; 第 3 章简要介绍考试大纲中涉及的知识点; 第 4 章解析一套模拟样题; 附录 A 列出系统使用中的常见问题与解答; 附录 B 列出使用 C++ 语言参考的考生的重要工具 STL 的功能和简单介绍。

本书将在“拼题 A”系统平台为读者提供丰富的学习资源。

PAT 全部真题可在“固定题目集”和“教育超市”的“PAT 真题模拟卷”栏目中找到, 读者可自行注册练习;

本书全部源代码可从“固定题目集”的“《PAT 备考通》题目集”栏目中下载;

本书的模拟样题收录在“固定题目集”的“《PAT 备考通》题目集”中, 读者注册账户后, 可在该题目集右上角的“读者验证码”文本框内输入封



底刮刮卡中的序列码，单击“开始答题”按钮即可进行自由练习。

本书主要章节由陈越编写，第 3 章由陈越和戴龙翱共同编写，附录 B 由戴龙翱编写。

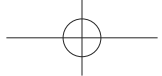
由于作者水平有限，书中难免存在不当之处，欢迎读者批评指正。

编者

2022 年 2 月

清华大学出版社

清华大学出版社



目 录

第1章 计算机程序设计能力考试概述

1

- 1.1 PAT的历史与组织结构 1
- 1.2 PAT的设置与规则 4
- 1.3 考试系统简介 8

第2章 计算机程序设计能力考试大纲解读

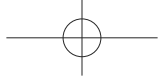
23

- 2.1 基础级考试大纲解读 23
- 2.2 乙级考试大纲解读 28
- 2.3 甲级考试大纲解读 29
- 2.4 顶级考试大纲解读 32

第3章 计算机程序设计能力考试知识点扼要

35

- 3.1 程序设计基础 35
- 3.2 排序、时空复杂度、查找与映射 46
- 3.3 链表的操作 58
- 3.4 树的应用 61
- 3.5 图的表示与相关算法 67
- 3.6 经典算法 77
- 3.7 高级数据结构 81

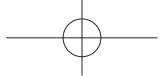


第4章 计算机程序设计能力考试模拟样题及详解 91

- 4.1 简单题 91
- 4.2 进阶题 99
- 4.3 提升题 115
- 4.4 难题 150

附录A “拼题A”系统使用常见问题与解答 162

附录B STL简介 168



第 1 章 计算机程序设计能力考试概述

计算机程序设计能力考试（Programming Ability Test, PAT；官网为 <https://www.patest.cn/>）采用分级式在线考试及自动评测方法，客观地考核考生的算法设计与程序设计的实现能力。目前，中国已有数百家企业将 PAT 成绩作为衡量应聘人员的程序设计能力的标准。浙江大学等多所高等院校已将 PAT 成绩作为推荐免试研究生的上机考试以及考研上机复试的直接替代成绩或重要参考指标。

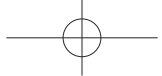
本书将为读者全面介绍 PAT 的相关信息，并简要回顾考试大纲涵盖的知识内容，最后给出一套模拟试题及其详细题解，以帮助读者，特别是第一次参加 PAT 的考生对 PAT 有更深入的了解，从而更好地为考试做好准备。

本章将从组织结构、考试规则、考试系统三方面介绍 PAT 的相关信息。

1.1 PAT 的历史与组织结构

进入 21 世纪后，随着信息技术的飞速发展，在计算机相关领域，无论是科研机构还是企业都对人才产生了巨大需求。近年来，各种关于应届生的调研报告都显示，就业市场中高科技行业的企业数量最多，特别是互联网、电子商务等新兴行业的人才需求增幅较大。

计算机相关专业在校生已达到“百万大军”，企业在招聘时也面临一定的选拔困难，特别是对基础程序设计与算法能力的考查，需要企业付出较



大的人力成本，但仍然难以保证招聘质量。

PAT 作为一个能够客观地考核程序设计与算法设计能力的第三方考试应运而生。PAT 的目标是成为 IT 界的标准化能力测试，即 IT 界的 TOEFL，为学生就业和企业选拔人才提供公平、有效的沟通平台。

2011 年，浙江大学计算机科学与技术学院创办了国内第一个标准化的计算机程序设计能力考试。同年 8 月，学院成立了 PAT 考试中心，推出了首场考试并颁发了学院认证的证书。

在经历了艰难的三年后，为进一步扩大 PAT 的影响力，在浙江大学本科生院的帮助下，PAT 考试中心联系全国高等学校教学研究中心谋求合作。全国高等学校教学研究中心是教育部高等教育司与高等教育出版社于 1999 年共建的单位，其主要职责包括：面向全国高等院校组织和开展高等教育教学、教材建设和现代教学技术及手段的研究；承担教学研究成果的鉴定和推广应用；组织编写教学刊物、简报、信息资料，通过图书、光盘、网络等多种形式向社会各界提供服务；承办教育部高等教育司委托的有关工作，其业务工作由教育部高等教育司指导。双方经过探讨，对考试的宗旨和目标达成高度一致的认同，并于 2014 年 11 月达成协议，共同运营 PAT 品牌。自 2015 年起，PAT 证书由全国高等学校教学研究中心和浙江大学共同认证。

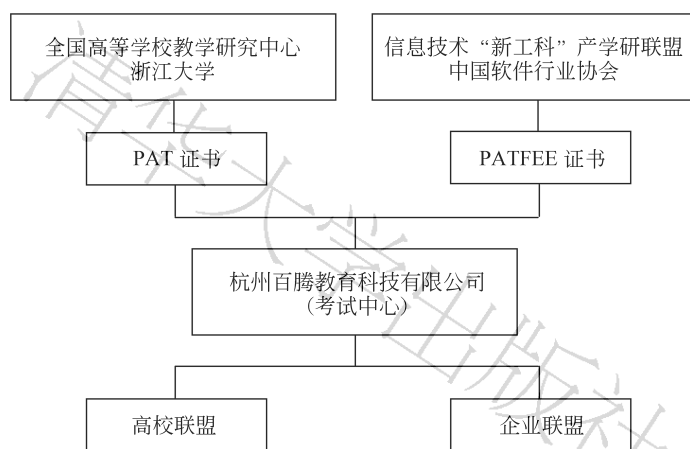
此后，随着考生规模的扩大，在线考试系统的压力不断增加。为长久保障系统的正常运行，PAT 创始人团队于 2016 年年底创办了杭州百腾教育科技有限公司（以下简称“百腾教育”），建立了一支强大、稳定的技术队伍，以保障 PAT 的可持续发展；考试的质量监管和证书颁发仍然由全国高等学校教学研究中心和浙江大学负责。

2017 年 11 月，为响应国家重大战略，面向以新技术、新业态、新模式、新产业为代表的新经济的发展需求，建立创新卓越人才培养体系，助力信息产业的超越式发展，信息技术“新工科”产学研联盟（以下简称“新工科联盟”）应运而生。新工科联盟由中国软件行业协会与国内部分高校、

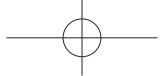
科研院所、相关企业联合组建，并得到了教育部高等教育司、工信部信息化和软件服务业司和人事教育司的大力支持。新工科联盟一经创立便开展了一系列工作，其中一项重要工作是针对高校信息类专业学生和信息产业从业人员开展信息技术能力评测工作，以期逐步形成信息产业创新人才的评价体系，为新一代信息产业的人才培养、人才评估、人才聘用等搭建平台。

新工科联盟信息技术能力测评的宗旨与 PAT 可谓不谋而合，因此，PAT 很自然地于 2018 年成为首个加入通用能力评测的考试，并有了另一个名称：“新工科”程序设计能力考试（Programming Ability Test for Emerging Engineering, PATFEE）。自 2018 年秋季起，考生可同时收到新工科联盟和中国软件行业协会联合颁发的 PATFEE 证书，以及全国高等学校教学研究中心和浙江大学联合颁发的 PAT 证书。

目前，PAT 的组织架构如图 1.1 所示。运营考试的核心机构为百腾教育，设立有考试中心，同时接受新工科联盟、中国软件行业协会、全国高等学校教学研究中心和浙江大学的指导，颁发 PAT 和 PATFEE 证书。同时，考试中心组织了高校联盟，负责线下考试的条件保障；组织了企业联盟，为成绩优秀的考生提供求职绿色通道。



❖ 图 1.1 PAT 的组织架构 ❖



高校联盟由遍布全国的高等院校和大型培训机构组成。联盟院校均与百腾教育签订了合作协议, 承诺为考试提供统一配置的机房和网络保障, 至少提供 C、C++、Java 三种编程语言的编译工具, 并配备监考人员。

企业联盟由国内外提供 IT 岗位的多家公司组成。联盟企业均与百腾教育签订了合作协议, 承诺将 PAT 成绩作为衡量应聘人员程序设计能力的标准。在考生同意的前提下, 联盟企业可免费获得每场考试的考生数据包, 包含考生的手机号码、电子邮箱、成绩、排名、提交的全部代码等。大部分企业都划定了明确的分数线, 承诺为达标的考生提供求职介绍信, 甚至直接向考生发出面试邀请。

1.2 PAT 的设置与规则

作为标准化考试, PAT 有较为稳定的难度等级标准与评分规则、报名与考试流程、证书申领等环节, 下面为读者进行详细介绍。

1. 难度等级

PAT 考查的是考生的算法设计与程序设计的实现能力, 根据考查的能力从低到高分以下四个等级。

- **基础级 (Fundamental Level)**: 考查初级编程能力。
- **乙级 (Basic Level)**: 考查基础编程和初级算法的综合应用能力。
- **甲级 (Advanced Level)**: 考查基础数据结构和基础算法的综合应用能力。
- **顶级 (Top Level)**: 考查高级数据结构和经典算法的综合应用能力。

其中, 基础级又细分为青铜、白银、黄金、白金、钻石五个段位。各等级的题量、考试时长、总分、试题语言如表 1.1 所示。

考试主服务器提供 30 余种编程语言的编译 / 解释器, 考试时间上限根据 C 语言标准答案运行时间的 3~5 倍设定。

表 1.1 PAT 各等级的考试信息

级 别	段 位	题 量	考试时长/分钟	总 分	试 题 语 言
基础级	青铜	5	30	30	中文
	白银	6	60	50	
	黄金	6	90	60	
	白金	6	120	80	
	钻石	7	150	100	
乙级	大师	5	180	100	中文
甲级	王者	4	180	100	英文
顶级	大圣	3	180	100	英文

2. 评分规则

PAT 的每题均要求考生严格按照输入 / 输出的要求提交程序以解决问题。程序必须经过若干测试用例的测试，每个测试用例均分配一定的分数。每题的得分为通过的测试用例的得分之和；整场考试的总得分为各题得分之和。考生可多次提交程序，取最高分；提交错误不扣分。

考生名次根据总得分决定，相同分数对应并列名次。

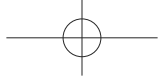
PAT 不设合格标准，凡参加考试且非零分的考生均有成绩，可获得统一颁发的证书。证书中将展示本次考试的“考试分数 / 满分”和“排名 / 考生总数”这两个指标，以供用人单位参考。

3. 报名与考试流程

PAT 根据级别的不同有不同的报名流程，同时根据线上、线下考场的不同要求又有不同的考试流程。

(1) 一般报名流程

PAT 的乙级、甲级、顶级考试由考试中心统一安排考试时间和地点。由于线下考场多为高校机房，因此每次考试的时间需要与各考点（联盟高校）的计算中心进行协调，并不固定。一般而言，会在春、秋、冬季各安排一次考试。其中，春季考试一般安排在二三月份，秋季考试安排在八九月份，



冬季考试安排在 11~12 月份。注意：冬季一般不安排顶级考试。

参加乙级、甲级、顶级考试的考生需要在报名开启后访问报名网站 (<https://www.patest.cn/reg>) 注册自己的真实身份。需要特别注意的是：管理员将随时删除用伪造的身份证号码注册的账号，并且在注册后，考生将无法修改自己的姓名和身份证号码，这是为了避免出现倒卖热门考点考位的情况。

报名者需要上传本人的 1 寸近照照片，照片将出现在准考证和 PAT 证书上。注册但没有上传照片的用户将无法报名考试。

此外，为方便考生向用人单位推荐自己，考生可以在报名时选择允许哪些联盟企业查看自己的哪些信息，并且可以在个人信息中填写 1000 字的简历作为自我介绍。不填写简历的考生通常会被认为“无求职需要”，企业一般不会主动联系。在考试结束前更新简历都是有效的。在考试结束后，系统将根据考生的意愿将考生信息打包发送给相应企业的人力资源部门。

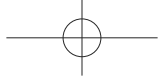
报名成功后，考生将在报名系统中看到自己的准考证，确认信息无误后，即可在考试当天携带准考证、身份证和笔前往指定考场参加考试。

(2) 线下考试流程

在线下考场参加考试的考生必须提前 30 分钟到达指定考场，并按监考指令就座。入座后，首先确认考试用机屏幕上显示的考生信息，随后将身份证放置在桌角待查。迟到 15 分钟者禁止入场；开考 15 分钟内禁止退场，违者按作弊论处。

监考员宣布考试开始后，考生可单击屏幕上的考试题目集，进入考试页面。通过“题目列表”查看题目内容；可使用本地 IDE 等辅助工具编写调试程序，将调试好的代码复制粘贴到对应题目下方的代码提交框中，选择合适的编译器 / 解释器并单击“提交”按钮；可以在“提交列表”中查看自己的提交结果；还可以通过“排名”列表查看自己的实时成绩。

提前交卷的考生必须先向监考员提出申请，得到允许后，须填写提前交卷声明，提交后方可离开考场。



(3) 线上考试流程

在线上考场参加考试的考生须自备考试环境，包括考试用机、双摄像头（一个为普通摄像头，另一个为手机摄像头），并自行安装监考客户端（详情请参阅官网首页“PAT 简介”中的最新版《考生须知》）。注意：参加线上考试的考生务必严格按照指定流程完成自测，并在确认自己的机器和网络条件符合考试要求后再确认报名。

参加线上考试的考生须提前 60 分钟到达自备考场，完成登录申请，就座并将身份证、草稿纸放在桌角待查，静待考试开始。同样，迟到 15 分钟的登录申请将被拒绝；开考 15 分钟内禁止退场，违者按作弊论处。

考试开始后，监考员将通过监考系统向考生发出指令，要求查验身份证件。考生须及时做出应答，并遵照指令配合检查。其他流程与线下考试相同。

提前交卷的考生必须先通过监考系统的呼叫功能向监考员提出申请，得到允许后，须填写提前交卷声明，提交后方可关闭监考客户端，退出考试。

(4) 基础级考试报名流程

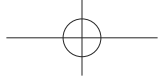
基础级考试是考试中心于 2018 年推出的面向机构的能力认证考试，考生必须向机构提出申请，由机构统一向 PAT 考试中心报名。所谓机构，可以是高校、企业，也可以是教育培训机构等。考试标准、考试命题、考试监督以及证书颁发均由考试中心统一负责，以保证考试的标准化和评价的客观性。

希望举办 PAT 基础级考试的机构可以自行决定考试段位、时间、地点，并至少提前一个月向考试中心提交考试申请。考试中心将审核机构考场的环境和配置，在确保其符合考试要求后，方可就本次考试的所有安排与机构签订工作协议。

机构需要自行收集考生信息并填写考生信息表，在考试前一周将完整的考生信息表发送到考试中心，以完成机构报名工作。

4. 证书申领

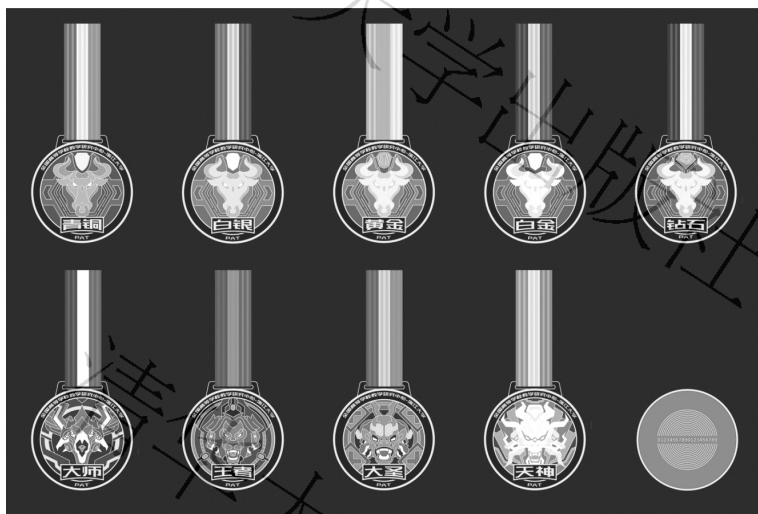
考试中心将在每场考试结束后的 24 小时内完成代码查重和监考录屏检



查工作。凡参加考试、无作弊行为且非零分的考生均有成绩,可获得统一颁发的证书。考生可以在考试结束后的 48 小时后登录报名网站,即可在“我的报名”中查询到自己的证书信息。如果需要申领纸质证书,则可以填写快递信息并提交申请,考试中心将通过快递向考生邮寄证书。

考试中心还提供官方证书的查验功能,在查验网站 (<https://www.patest.cn/certificates>) 中输入证书编号即可查验其真实性。

此外,成绩优秀的考生可获得相应段位的精美徽章,徽章背面将刻有考生的姓名和证书编号,作为 PAT 优秀考生独有的尊享标志(如图 1.2 所示)。只有在某个段位获得优秀成绩的考生才有资格购买实体徽章,每个证书编号仅对应其有资格购买的徽章各一块,不可重复购买。具体规则可在官网的“徽章商城”中查阅。

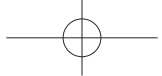


❖ 图 1.2 PAT 徽章 ❖



1.3 考试系统简介

PAT 自 2018 年起采用了“拼题 A”系统作为考试平台。下面对考试平台的背景和资源进行简要介绍,并给出平台目前支持的编程语言的样例程



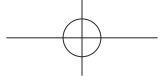
序，以供读者参考。更多“常见问题与解答”请参阅本书的附录 A。

1. 什么是“拼题 A”

“拼题 A” (<https://pintia.cn>) 是浙江大学“国家级程序设计系列课程”教学团队与网易、百腾教育于 2015 年 9 月合作推出的面向高校和社会的程序自动评测、开放式教学辅助平台。系统创建之初的全称为“程序设计类教学辅助平台”(Programming Teaching Assistant, PTA)。发展到今天,“拼题 A”已不限于计算机程序设计类课程的辅助教学,而是将功能延伸到多个学科的近百门课程。

“拼题 A”是以浙江大学近 20 年来在程序设计能力培养、精品课程与资源共享课程建设、在线开放课程建设、教学团队建设等方面的工作为基础,针对“程序设计”“数据结构”等程序设计类课程的实践性特征,重点结合新兴的在线课程建设需求和不同层次的学校在程序设计类课程方面的实践教学需求而设计的。“拼题 A”致力于打造面向学校与企业的优质教学资源共建、共享平台,开创互联网时代多校多企共同参与、共同建设、共同利用教育资源的新模式,相关的探索与实践工作获得 2016 年浙江省高等教育教学成果奖一等奖,并且是 2018 年全国高等教育教学成果一等奖的重要内容之一,近年来更是成为教育部首批“新工科”研究与实践项目(2019 年)、科技部科技创新 2030-“新一代人工智能”重大项目(2020 年)和国家自然科学基金重点项目(2020 年)研究的支撑平台。在 2020 年新冠肺炎疫情防控期间,“拼题 A”凭借丰富的教学资源与稳定支撑百万师生日常教学的强大表现,被教育部产学合作协同育人项目专家组推荐为“在新冠肺炎疫情防控期间支持高校在线教学的产学合作教学资源”。

“拼题 A”系统可以支持 20 万人同时在线进行作业、测验、考试、竞赛等教学活动,其高可用分布式评测架构已获国家发明专利;题库支持判断、选择(单选、多选)、填空(程序填空、函数填空)、编程(常规编程、多文件编程、数据库编程)、主观等题目类型,并为教师用户提供智能推荐;支持日常教学、大规模互评、ICPC 规则、企业招聘等多种判题模式;提供



代码 / 文本查重、随机组卷、线上 / 线下监考等高级功能, 为教学质量提供全方位的保障; 提供人工与智能混合答疑系统, 可以为学习者提供即时帮助。

2. 考生常用操作

考生在报名参加 PAT 之前, 必须首先访问“拼题 A”系统主站注册账号, 并通过注册邮箱中的链接激活账号。这个账号既可用于考试报名, 也可用于日常练习。

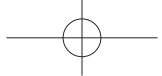
考生在使用激活的账号登录主站后, 会看到首页的“固定题目集”中列出的各种公开练习题目集。选择一个题目集, 单击标题即可进入题目集内页, 并可以做以下操作。

- **查看题目:** 单击左栏的“题目列表”即可查看题目内容。
- **提交答案:** 可使用本地 IDE 等辅助工具编写调试程序, 将调试好的代码复制粘贴到对应题目下方的代码提交框中, 选择合适的编译器 / 解释器并提交。
- **查看反馈:** 提交后, 系统将自动弹窗显示本次提交的判题结果, 包括每组测试数据是否通过、运行的时间、使用的内存量等; 如果未正确通过, 则会显示错误类型 (如“答案错误”“运行超时”等); 如果出题者给出了测试数据的提示信息, 则考生也可以在这个页面看到, 当然考试中不会显示提示信息; 另外, 在弹窗关闭后, 考生仍可以随时单击左栏的“提交列表”, 以查看自己的全部提交详情。
- **查看实时成绩:** 单击左栏的“排名”, 考生即可随时看到自己的总分和名次。

3. 考生练习资源

“拼题 A”系统为考生提供了包含 PAT 全部真题的练习集, 其中, 3 年前的真题分布在以下 3 个题目集中供考生免费练习。

- **PAT (Basic Level) Practice (中文):** 乙级真题题库。



- PAT (Advanced Level) Practice: 甲级真题题库。
- PAT (Top Level) Practice: 顶级真题题库。

3 年内的真题暂放在“教育超市”的“PAT 真题模拟卷”栏目下，并且提供了“时光机”功能，使考生宛如置身当年的考试现场，可以参与当时真实考生的排名，以检验自己的实力。

此外，“团体程序设计天梯赛 - 练习集”也是一个很好的补充练习集，该项赛事的题目覆盖了从 PAT 基础级到顶级的全部难度，题干均用中文描述，对初学者更为友好。

“拼题 A”系统同时还为各大出版社的教材提供线上练习的支持。部分以教材名称命名的题目集（如“浙大版《数据结构（第2版）》题目集”）即对应该教材的配套编程类习题。此类教材题目集要求用户购买对应教材，输入教材封底的读者码后才可以开始线上练习。另外，部分选择、判断、填空类的习题则放在“教育超市”的“教材配套习题”栏目下。目前，“拼题 A”系统支持 C 语言、Python 语言、数据结构三类课程的教材，覆盖了 PAT 考查的大部分基础知识。

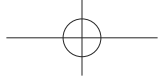
4. 系统支持的编程语言及样例程序

“拼题 A”系统目前支持 30 余种编译器 / 解释器，并欢迎用户随时提出添加最新工具的需求。

在 PAT 中，考生可以使用系统提供的任意一种语言答题，并且可以用不同的语言解决不同的问题，对编程语言的选择是在提交代码前做出的。

另一方面，PAT 对考生的编程准确性和速度都有较高的要求，而善用工具会为考生带来很大的助力。例如，C++ 中的一套功能全面、强大的标准模板库（Standard Template Library, STL）就提供了许多功能模块，考生如果能熟练巧妙地运用它，便可以为编写 C++ 程序提供不少便利。关于 STL 的简介及其实用技巧，请参阅本书的附录 B。

下面给出每种编程语言的样例程序，它们均可用于解决这道简单的题目：读入两个整数 A 和 B，然后输出它们的和。



ASM (.asm)

```
; add1.asm
SECTION .data
    formatin: db "%d", 0
    formatout: db "%d", 10, 0 ; newline, nul terminator
    integer1: times 4 db 0 ; 32-bits integer = 4 bytes
    integer2: times 4 db 0 ;
SECTION .text
    global main
    extern scanf
    extern printf

main:

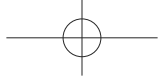
    push ebx ; save registers
    push ecx

    add esp, 4 ; remove parameters
    push integer1 ; address of integer1 (second
parameter)
    push formatin ; arguments are right to left (first
parameter)
    call scanf

    add esp, 4 ; remove parameters
    push integer2 ; address of integer2
    push formatin ; arguments are right to left
    call scanf

    add esp, 8 ; remove parameters

    mov ebx, dword [integer1]
```



```
mov ecx, dword [integer2]
add ebx, ecx ; add the values ; the addition
push ebx
push formatout
call printf ; call printf to display the sum
add esp, 8 ; remove parameters
pop ecx
pop ebx ; restore registers in reverse order
mov eax, 0 ; no error
ret
```

AWK (.awk)

```
{ print $1 + $2 }
```

Bash (.sh)

```
while read i; do
    echo ${i/ /+})
done
```

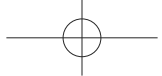
C (.clang .c)

```
#include <stdio.h>

int main()
{
    int a, b;

    while(scanf("%d %d", &a, &b) != EOF)
        printf("%d\n", a + b);

    return 0;
}
```



C++ (.clang .cpp)

```
#include <iostream>

using namespace std;

int main()
{
    int a, b;

    while (cin >> a >> b)
        cout << a+b << endl;

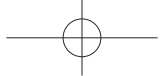
    return 0;
}
```

Common Lisp (.cl)

```
(loop for n = (read t nil nil)
      while n
      do (format t "~d~C" (+ n (read)) #\linefeed))
```

C# (.cs)

```
public class Program {
    public static void Main() {
        string line;
        while ((line = System.Console.ReadLine ()) != null) {
            string[] tokens = line.Split();
            System.Console.WriteLine(int.Parse(tokens[0]) +
                                     int.Parse(tokens[1]));
        }
    }
}
```



D (.d)

```
import std.stdio;

void main() {
    int a, b;
    while (readf(" %s %s", &a, &b)) {
        writeln(a + b);
    }
}
```

FORTRAN95 (.f95)

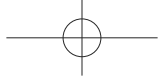
```
program a
integer :: i, j, stat
do
    read(*,*,iostat=stat) i, j
    if (stat /= 0) exit
    write(*, '(I0)') i+j
end do
end program a
```

Go (.go)

```
package main

import "fmt"

func main() {
    var a, b int
    for {
        n, _ := fmt.Scanf("%d %d", &a, &b)
        if (n != 2) { break }
        fmt.Println(a + b)
    }
}
```



Haskell (.hs)

```
main = interact $ unlines . map (show . sum . map read
    . words) . lines
```

Java (.java)

```
import java.util.Scanner;

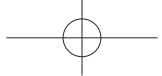
public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        while (in.hasNextInt()) {
            int a = in.nextInt();
            int b = in.nextInt();
            System.out.println(a + b);
        }
    }
}
```

JavaScript (.js)

```
var fs = require('fs');
var buf = '';

process.stdin.on('readable', function() {
    var chunk = process.stdin.read();
    if (chunk) buf += chunk.toString();
});

process.stdin.on('end', function() {
    buf.split('\n').forEach(function(line) {
        var tokens = line.split(' ').map(function(x) {
```



```
return parseInt(x); });  
    if (tokens.length !== 2) return;  
    console.log(tokens.reduce(function(a, b) { return  
a + b; }));  
    });  
});
```

Kotlin (.kt)

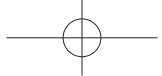
```
import java.util.Scanner  
  
fun main(args: Array<String>) {  
    val input = Scanner(System.`in`)  
    while (input.hasNextInt()) {  
        val a = input.nextInt()  
        val b = input.nextInt()  
        println(a + b)  
    }  
}
```

Lua (.jit.lua)

```
for a, b in io.read('*a'):gmatch('([%d-]+) ([%d-]+)')  
do  
    print(tonumber(a) + tonumber(b))  
end
```

Object-C (.m)

```
#import <Foundation/Foundation.h>  
  
int main (int argc, const char * argv[])  
{  
    int a, b;  
    while(scanf("%d %d", &a, &b) != EOF) {
```



```
    printf("%d\n", a + b);  
}  
return 0;  
}
```

OCaml (.ml)

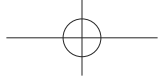
```
try  
  while true do  
    Scanf.scanf " %d %d" (fun a b -> Printf.printf  
"%d\n" (a + b))  
    done;  
    None  
  with  
    End_of_file -> None  
;;
```

Octave (.octave)

```
while (true)  
  a = scanf("%d", [1, 2]);  
  if (! feof(stdin() )  
    printf("%d\n", sum(a));  
  else  
    break  
  endif  
endwhile
```

PASCAL (.pas)

```
var  
  a, b: integer;  
begin  
  while not eof(input) do begin
```



```
    readln(a, b);  
    writeln(a + b);  
end;  
end.
```

Perl (.pl)

```
#!/usr/bin/perl -pla  
$_ = $F[0] + $F[1]
```

PHP (.php)

```
<?php  
while (fscanf(STDIN, "%d%d", $a, $b) == 2) {  
    print ($a + $b) . "\n";  
}
```

Python (.py)

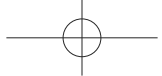
```
import sys  
  
for line in sys.stdin:  
    print(sum(map(int, line.split())))
```

Python (.3.py)

```
while True:  
    raw = input()  
    if raw == '':  
        break  
    print(sum(map(int, raw.split())))
```

R (.r)

```
num <- scan("stdin", what = integer(), quiet = TRUE)  
write(sum(num), file = "")
```



Racket (.rkt)

```
#lang racket

(define (f)
  (let ([n (read)])
    (if (eof-object? n)
        (void)
        (begin (displayln (+ (read) n))
                 (f)))))

(f)
```

Ruby (.rb)

```
STDIN.each_line do |line|
  puts line.split.map(&:to_i).inject(:+)
end
```

Rust (.rs)

```
use std::io;

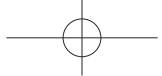
fn main() {
  let mut input = String::new();

  io::stdin()
    .read_line(&mut input)
    .unwrap();

  let mut s = input.trim().split(' ');

  let a_str = s.next().unwrap();
  let a: i32 = a_str.parse().unwrap();

  let b_str = s.next().unwrap();
```



```
let b: i32 = b_str.parse().unwrap();

println!("{}", a + b);
}
```

Scala (.scala)

```
import scala.io.StdIn

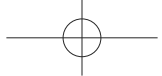
object Main {
  def main(args: Array[String]): Unit = {
    while(true) {
      val line = StdIn.readLine()
      if (line == null) {
        return
      }
      println(line.split(" ").map{ case i => i.toInt }.sum)
    }
  }
}
```

Swift (.swift)

```
import Foundation
while let line = readLine() {
  let words: [String] = line.components(separatedBy:
  NSCharacterSet.whitespaces)
  let inputs: [Int] = words.map { Int($0)! }
  print(inputs[0] + inputs[1])
}
```

Vala (.vala)

```
public static int main(string[] args) {
  int a = 0, b = 0;
```



```
while (stdin scanf("%d%d", &a, &b) == 2) {  
    stdout printf("%d\n", a + b);  
}  
return 0;  
}
```

Visual Basic (.vb)

Module Program

```
Sub Main()  
    Do  
        Dim line As String = Console.ReadLine()  
        If line Is Nothing Then Exit Do  
        Dim a() As String = line.Split(" "c)  
        Console.WriteLine(CInt(a(0)) + CInt(a(1)))  
    Loop  
End Sub  
  
End Module
```