



中國人民大學

RENMIN UNIVERSITY OF CHINA

信息学院

SCHOOL OF INFORMATION

程序设计荣誉课程

0. 基础1——课程概览

授课教师：游伟 副教授、孙亚辉 副教授

授课时间：周二14:00 – 15:30，周四10:00 – 11:30（教学三楼3304）

上机时间：周二18:00 – 21:00（理工配楼二层机房）

课程主页：<https://www.youwei.site/course/programming>

2023年图灵实验班选拔试卷第6题

六. (20分) 简易程序语言.

有一门简易程序语言 (T 语言), 支持下列 10 种基本操作:

- $n \leftarrow \text{INPUT}$: 读入用户输入并存放在变量 n 中
- $\text{OUTPUT} \leftarrow z$: 将变量 z 的值输出到屏幕
- $z \leftarrow c$: 将变量 z 的值设置为 c (即 $z=c$), c 是一个整数或者另一个变量的值
- $z \leftarrow \text{ADD}(x, y)$: 将变量 x 和变量 y 的值相加, 结果存放于变量 z 中 (即 $z=x+y$)
- $z \leftarrow \text{MUL}(x, y)$: 将变量 x 和变量 y 的值相乘, 结果存放于变量 z 中 (即 $z=x*y$)
- $z \leftarrow \text{NEG}(x)$: 将变量 x 的负数存放于变量 z 中 (即 $z=-x$)
- $z \leftarrow \text{INV}(x)$: 将变量 x 的倒数存放于变量 z 中 (即 $z=1/x$)
- $\text{STORE}(i, x)$: 将变量 x 的值存放在数列 $\{a_n\}$ 的第 i 项中 (即 $a_i=x$)
- $z \leftarrow \text{LOAD}(i)$: 从数列 $\{a_n\}$ 中取出第 i 项的值, 存放于变量 z 中 (即 $z=a_i$)
- $\text{FOR } i : b \text{ to } e \{ \}$: 循环执行 $\{ \}$ 内的操作, i 的初始值设置为 b , 每次循环将变量 i 的值加 1, 直至 i 的值为 e 时停止循环, 总共循环 $e-b+1$ 次; 如果 $e < b$, 则不执行任何操作

注: 变量名可用任意小写字母表示, 只能使用 T 语言支持的 10 种基本操作作答.

【材料 1】

$$\text{斐波拉契数列的递推公式是: } a_n = \begin{cases} 0, & n = 0 \\ 1, & n = 1 \\ a_{n-1} + a_{n-2}, & n \geq 2 \end{cases}$$

使用 T 语言实现计算上述数列第 n 项数值的功能, 代码如下 (//后面的文字是对当前行代码的注释说明):

```
n ← INPUT // 读取用户输入 n
STORE(0, 0) // 设置 a_0 的值为 0
STORE(1, 1) // 设置 a_1 的值为 1
FOR i : 2 to n // 循环 n-1 次, 第一轮循环时变量 i 的值为 2
{
  x ← LOAD(i-1) // 设置变量 x 的值为 a_{i-1}
  y ← LOAD(i-2) // 设置变量 y 的值为 a_{i-2}
  z ← ADD(x, y) // 设置变量 z 的值为 x + y (即 a_{i-1} + a_{i-2})
  STORE(i, z) // 设置 a_i 的值为 z (即 a_{i-1} + a_{i-2})
} // 每次循环后, 变量 i 的值自动加 1
z ← LOAD(n) // 设置变量 z 的值为 a_n
OUTPUT ← z // 输出变量 z 的值
```

【材料 2】

将给定的一个数列 $\{a_n\}$: $a_1, a_2, a_3, a_4, a_5, a_6, \dots$ 按照一定规则依顺序用括号将它分组, 则可以得到以组为单位的序列. 如在上述数列中, 将 a_1 作为第 1 组, 将 a_2, a_3 作为第 2 组, 将 a_4, a_5, a_6 作为第 3 组, \dots 依次类推, 第 m 组有 m 个元素, 即可得到以组为单位的序列: $(a_1), (a_2, a_3), (a_4, a_5, a_6), \dots$ 通常称此数列为分组数列.

一般地, 数列 $\{a_n\}$ 的分组数列用如下的形式表示: $(a_1, a_2, \dots, a_y), (a_{y+1}, a_{y+2}, \dots, a_r), (a_{r+1}, a_{r+2}, \dots, a_s), \dots$, 其中第 1 个括号称为第 1 组, 第 2 个括号称为第 2 组, 第 3 个括号称为第 3 组, \dots , 第 m 个括号称为第 m 组, 而数列 $\{a_n\}$ 称为这个分组数列的原数列. 若某一个元素在分组数列的第 m 个组中, 且从第 m 个括号的左端起是第 m 个数, 则称该元素为第 m 组中的第 m 个元素.

【请回答以下 3 个问题】:

(1) 下列 T 语言代码运行完成后, 形成数列 $\{a_n\}$, 请写出该数列的递推公式, 并求出 a_{10} 和 a_{500} 的值.

```
n ← INPUT
s ← 0
FOR i : 1 to n
{
  u ← MUL(2, i)
  v ← NEG(s)
  x ← ADD(u, v)

  w ← ADD(u, -1)
  y ← INV(w)
  z ← MUL(x, y)

  s ← u
  STORE(i, z)
}
z ← LOAD(n)
OUTPUT ← z
```

(2) 用 T 语言实现求数列 $\left\{\frac{a_n}{2n+1}\right\}$ 的前 n 项和 S_n 的功能 (输入 n 的值, 输出 S_n 的值).

(3) 将正奇数集合 $\{1, 3, 5, \dots\}$ 从小到大按第 i 组有 $2i-1$ 个奇数进行分组: $\{1\}, \{3, 5, 7\}, \{9, 11, 13, 15, 17\}$. 请完成: ①. 计算第 10 组第 15 个元素的值; ②. 计算 1991 位于第几组第几个元素; ③. 用 T 语言实现求第 n 组第 m 个元素的值 (输入 n 和 m 的值, 输出对应元素的值). 例如: 输入 3 和 5, 输出 17, 代表第 3 组第 5 个元素的值是 17.

教学团队

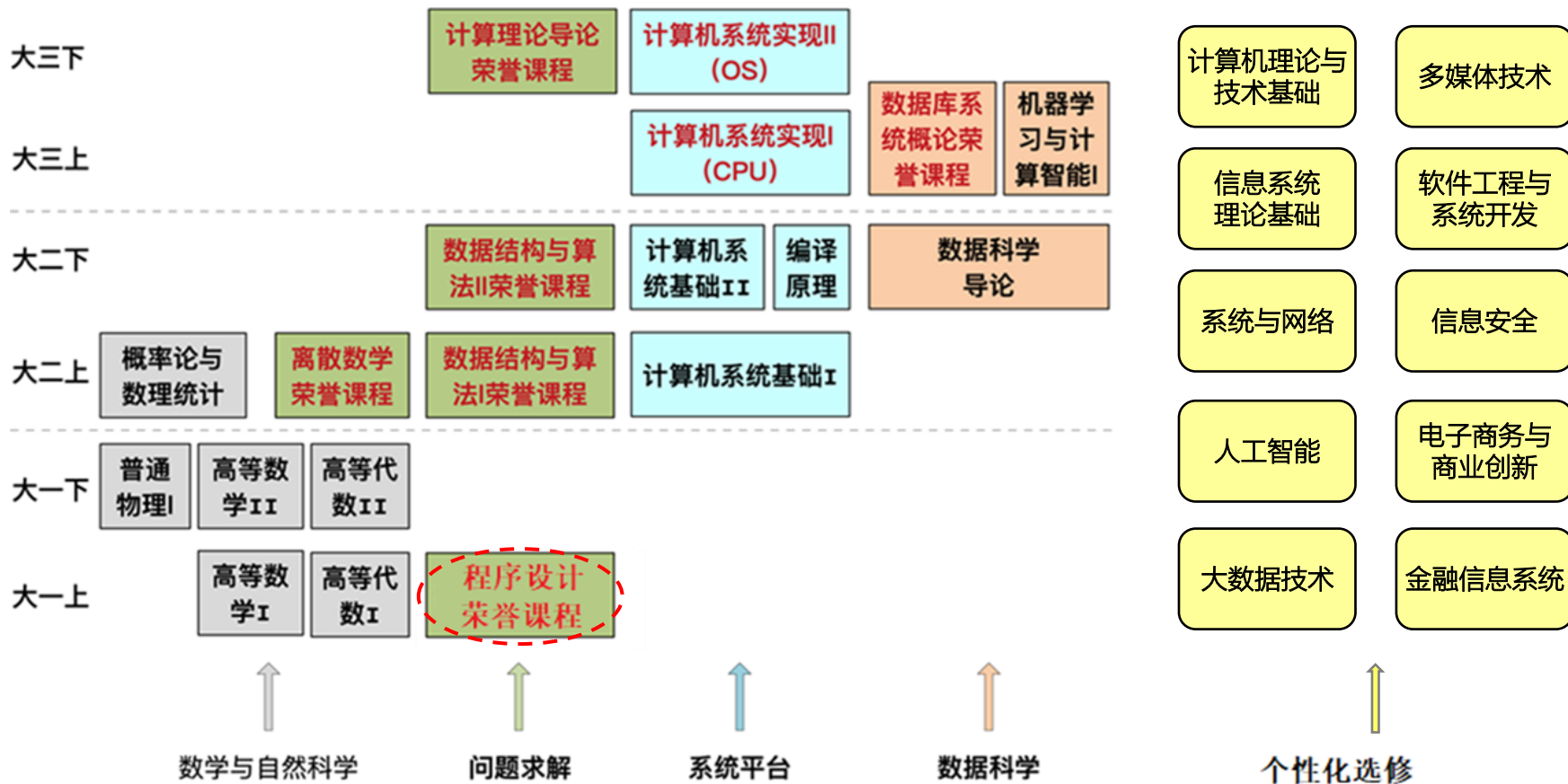
■ 教师：游伟 副教授、孙亚辉 副教授

- 邮箱: youwei@ruc.edu.cn、yahuisun@ruc.edu.cn
- 办公室: 游泳馆配楼406、信息楼126B

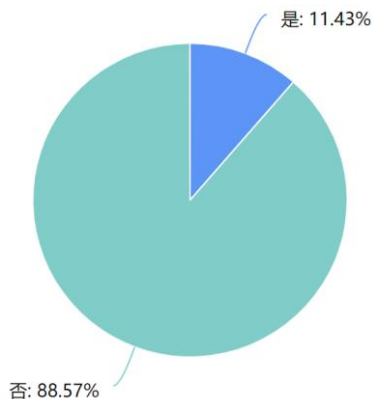
■ 助教：

- 字阳（2024级硕士研究生）ziyang12315@ruc.edu.cn
- 卢朗毅（2024级硕士研究生）lulangyi285711@163.com
- 李家裕（2024级硕士研究生）1309975419@qq.com
- 冯宵瑶（2021级理科试验班）2021201516@ruc.edu.cn

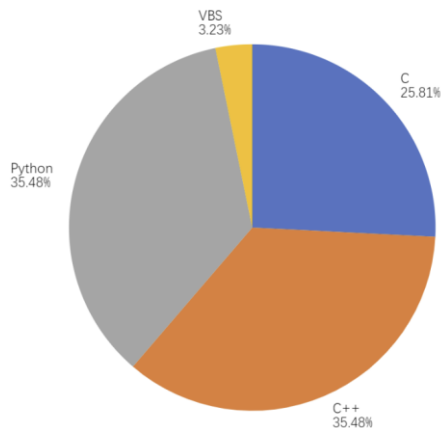
课程定位



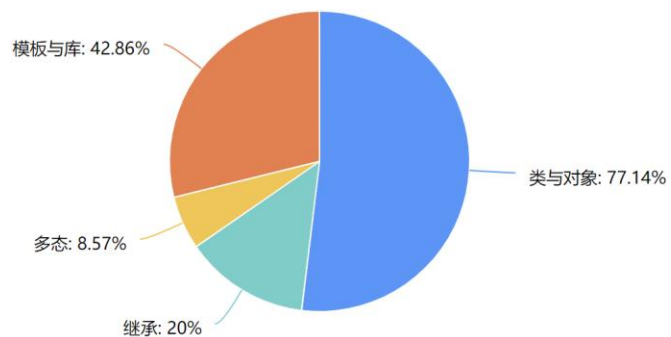
基础调查



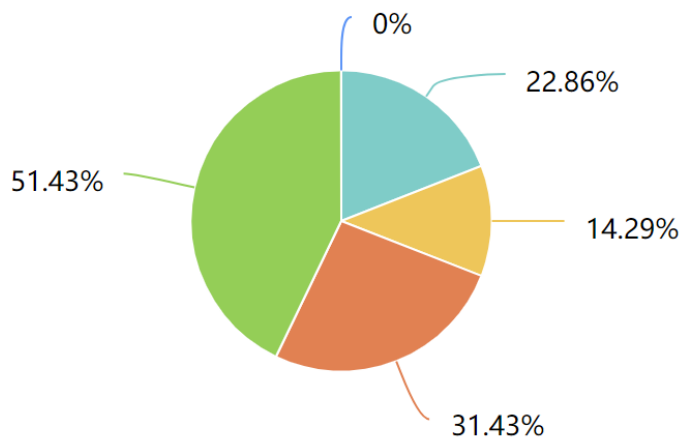
是否参加过信息学竞赛



使用的编程语言



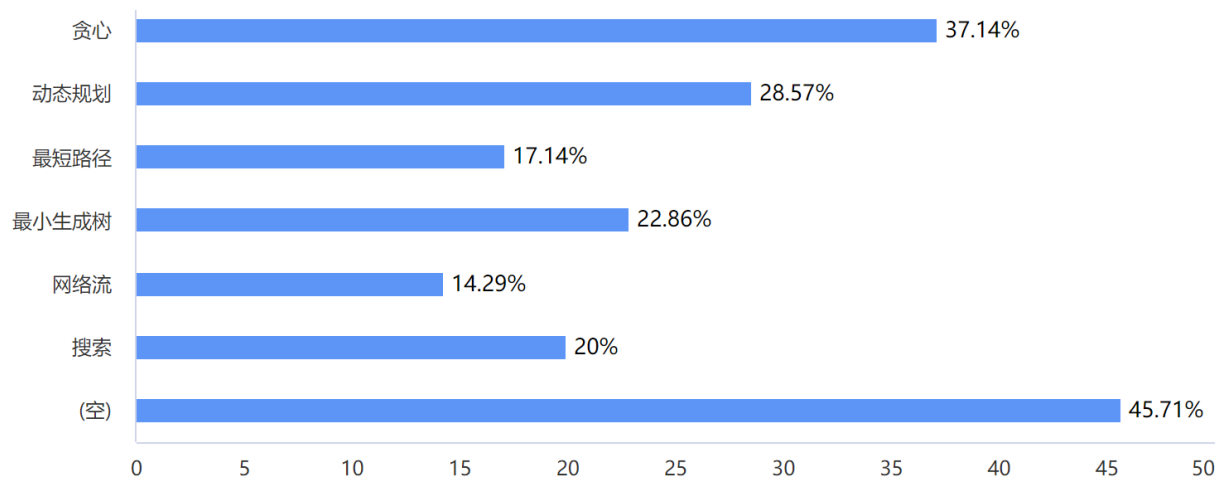
面向对象基础



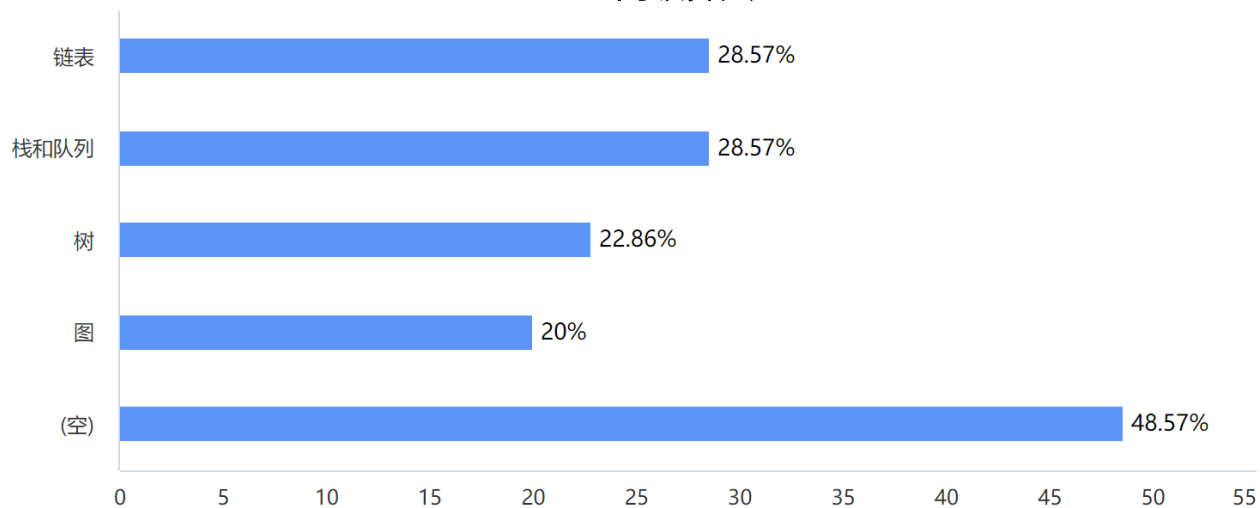
程序设计综合能力

- 编写过较大规模的程序或大型程序的组件
- 编写过简单实际应用
- 编写过程序解决某个具体实际问题 (如统计班级成员信息等)
- 编写过试题程序 (如在线评测试题)
- 无

基础调查

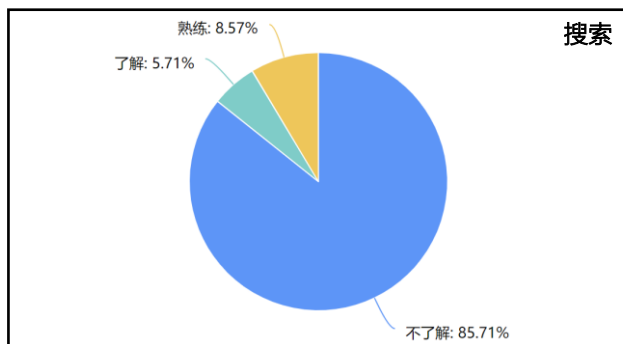
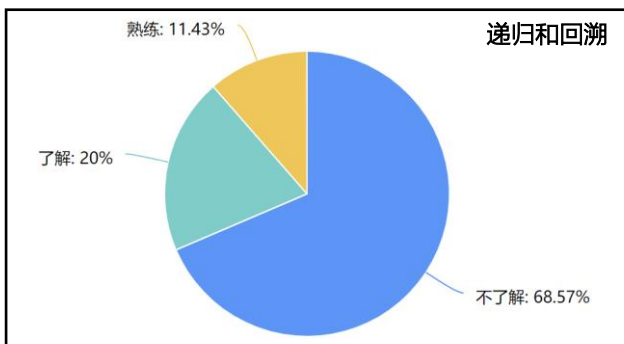
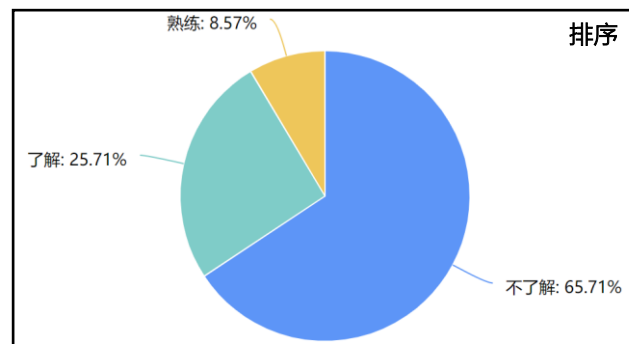
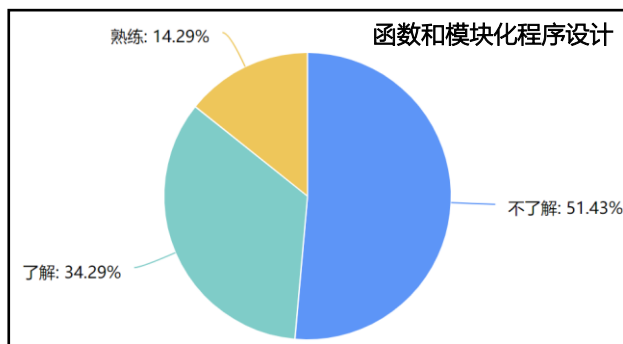
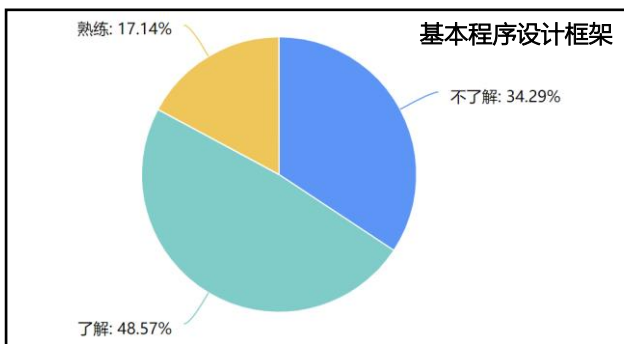
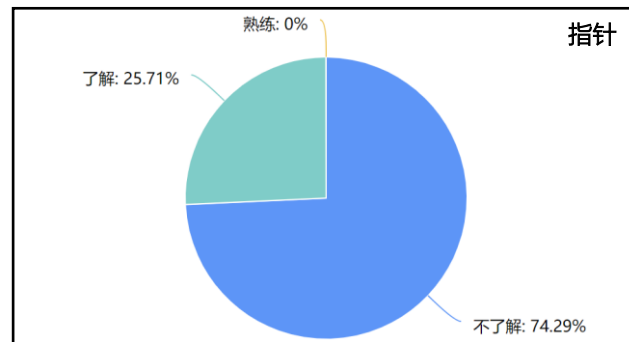
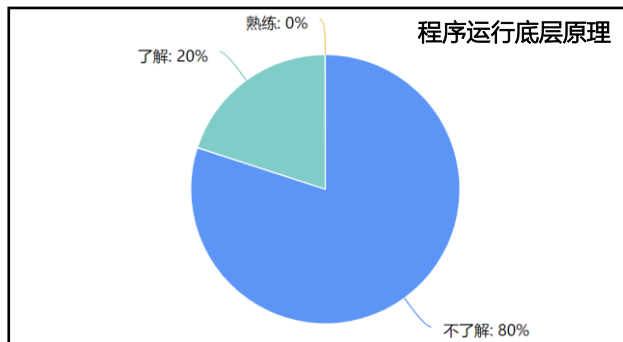
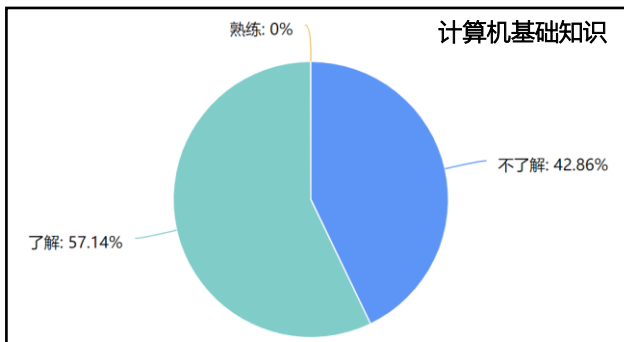


高级算法



数据结构

基础调查



- ### 调查结论
- 少数同学参加过信息学竞赛
 - 学过编程的同学主要用C/C++
 - 大多数同学有一定的程设习得能力
 - 部分同学掌握简单算法数据结构
 - 熟悉底层基础和原理的同学不多

底层基础不扎实带来的问题

■ 示例1：美链币智能合约漏洞，造成近10亿美元的经济损失

```
1. function batchTransfer(uint256 _value, uint256 _receivers[]) {
2.     uint cnt = _receivers.length; //cnt: 交易份额; _value: 交易单价; amount: 交易总额
3.     uint256 amount = uint256(cnt) * _value; 整数溢出漏洞
4.     require(cnt > 0 && cnt <= 20); //要求cnt∈(0,20]
5.     require(_value > 0 && balances[msg.sender] >= amount); //要求_value>0并且转出者账户余额≥交易总额
6.     balances[msg.sender] = balances[msg.sender].sub(amount); //转出者账户余额减少
7.     for (uint i = 0; i < cnt; i++) { //进行cnt次交易
8.         balances[_receivers[i]] = balances[_receivers[i]].add(_value); //转入者账户余额增加
9.     }
10.     .....
11.     return true;
12. }
```


底层基础不扎实带来的问题

■ 示例2：冲击波病毒利用微软RPC漏洞，感染超800万台计算机

```
1. error_status_t _RemoteActivation(..., WCHAR *pwszObjectName, ... ) {
2.     *pshr = GetServerPath(pwszObjectName, &pwszObjectName);
3.     ...
4. }

5. HRESULT GetServerPath(WCHAR *pwszPath, WCHAR **pwszServerPath ){
6.     WCHAR *pwszFinalPath = pwszPath;
7.     WCHAR wszMachineName[MAX_COMPUTERNAME_LENGTH_FQDN+1];
8.     hr = GetMachineName(pwszPath, wszMachineName);
9.     *pwszServerPath = pwszFinalPath;
10.}

11. HRESULT GetMachineName(WCHAR *pwszPath, WCHAR wszMachineName[MAX_COMPUTERNAME_LENGTH_FQDN+1])
12. {
13.     pwszServerName = wszMachineName;
14.     LPWSTR pwszTemp = pwszPath + 2;
15.     while ( *pwszTemp != L'\\' ) *pwszServerName++ = *pwszTemp++;
16.     ...
17. }
```

源数据来自远程网络包

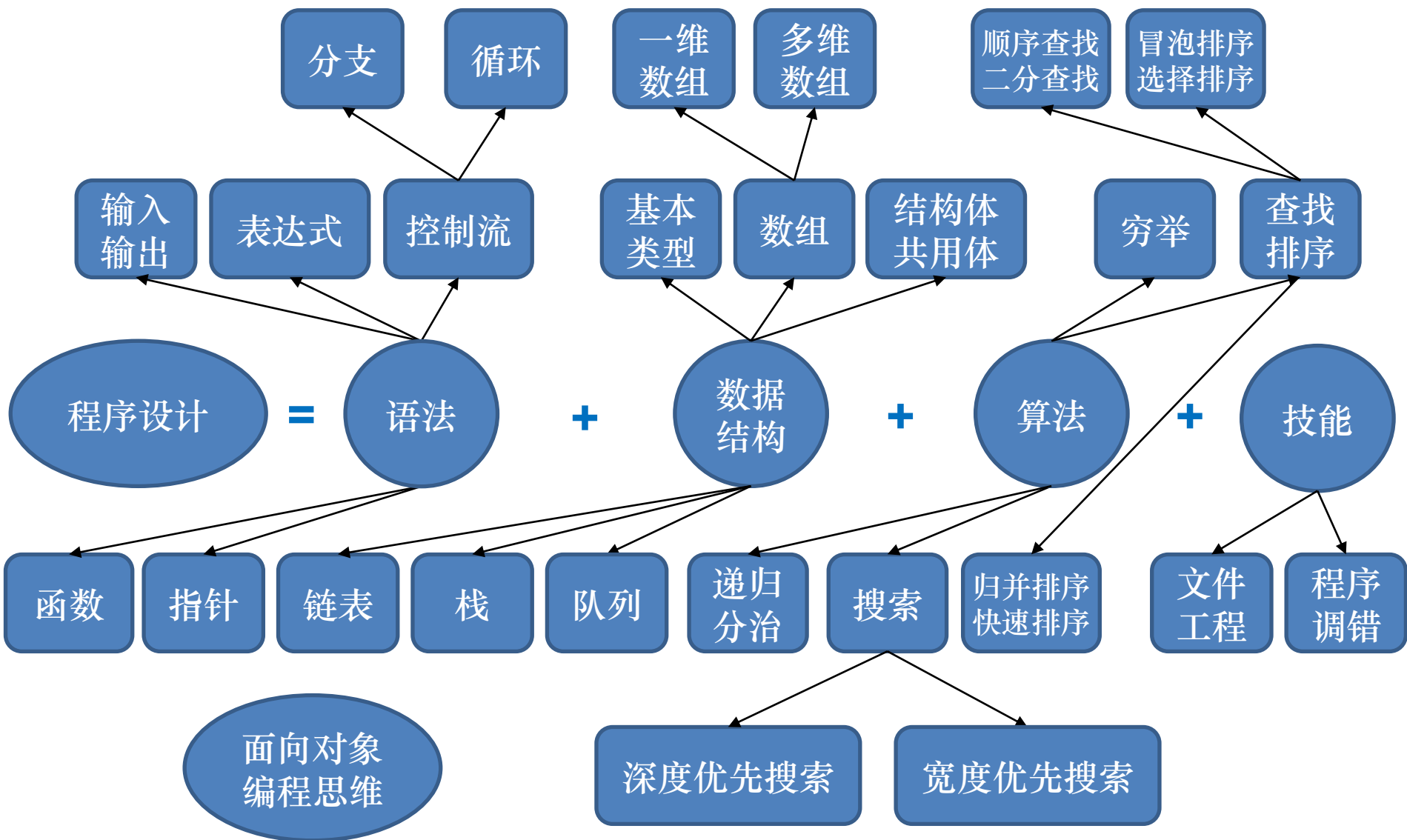
字符串复制操作
(可能会导致缓冲区溢出)

教学目标

- 建立程序设计基本思想（程序设计=语法+数据结构+算法+技能）
 - 使用程序设计的思想分析和解决问题
 - 基础算法（穷举、排序、递归、分治、搜索）
 - 简单数据结构（数组、链表、栈、队列）
- 掌握程序设计基本技能（写得了代码，查得出异常）
 - 编写代码
 - 调错修正
- 了解程序运行底层原理（知其然，知其所以然）
 - 计算机底层基础知识和基本概念
 - 程序编译运行的整体流程
- 开展小型工程项目实践（剖丁解牛，依葫芦画瓢）
 - 剖析真实工程项目，了解项目结构和编写规范
 - 动手编写工程项目，实现现实需求

荣誉内容

知识脉络



课程考核

- 期末成绩 (40%)
- 平时成绩 (60%)
 - 课程作业 (40%) : YOJ上机题, 课堂派笔试题
 - 期中考试 (40%) : 上机+笔试
 - 课程互动 (20%) : 课堂讨论互动、刷题量, 报告分享
- 附加成绩
 - 编程对抗赛
 - 附加题

编程对抗赛

- 同学们以小组为单位参赛，并出题、写解题报告

The screenshot shows a web browser window displaying a competition results page. The title is '2021-2022学年程序设计1荣誉课程团体赛'. Below the title is a table with columns for '学号' (Student ID), '姓名' (Name), '总分' (Total Score), and '答题情况' (Question Status). The '答题情况' column is further divided into 13 sub-columns labeled 686 through 699. The table lists 10 teams with their respective scores and performance in each sub-column.

学号	姓名	总分	答题情况												
			686	687	688	689	690	691	692	694	695	696	697	698	699
21002	潘俊达 +许嘉瑞 +叶伟堂	788	100	100	100	70	60	90	90	40	73	25	10	30	0
21005	贾昊睿 +赵宸+冯 锦程	681	100	100	100	80	60	100	25	40	60	16	0	0	0
21003	李天成 +刘博宇 +梁建伟	629	100	100	85	40	60	100	15	40	73	16	0	0	0
21009	邓皓元 +叶泽华 +田子航	613	100	100	100	60	30	90	20	0	57	16	10	30	0
21004	程云飞 +耿韵阳 +孙铸桐	565	100	100	75	30	0	90	0	40	80	0	10	40	0
21006	詹倚飞 +李沛康 +张启豪	551	100	100	100	10	20	100	0	40	77	4	0	0	0
21007	鲁露宁 +马博博 +冯肖斌	520	100	100	100	0	0	90	25	5	90	0	10	0	0
21010	群钰星 +廖有阳 +邓李俊 +邢致森	460	100	100	100	20	0	60	0	0	80	0	0	0	0



方设...安全...
遵守以下用户守则, 并承担起...
1. 遵守学校关于实验室...
用实验室的各类设备...
防...

课程免听申请

- 对象：有一定编程基础，学过相关内容的同学
- 条件：
 - 参加编程能力提前考试，通过指定成绩线
 - 对教师指定的一道题目，撰写解题报告并做报告分享
- 规则：
 - 可根据课程安排，自行决定是否来听课
 - **需要**参加期中考试和期末考试
 - 可不提交课程作业和课程互动，将期中考试成绩作为整体平时成绩；也可选择提交课程作业和课程互动，正常计算平时成绩
 - 无论选择哪种计算方式，附加成绩都可以加入平时成绩。

附加题和对抗赛加分说明

■ 附加题

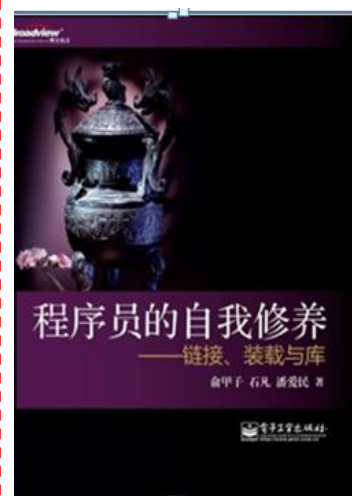
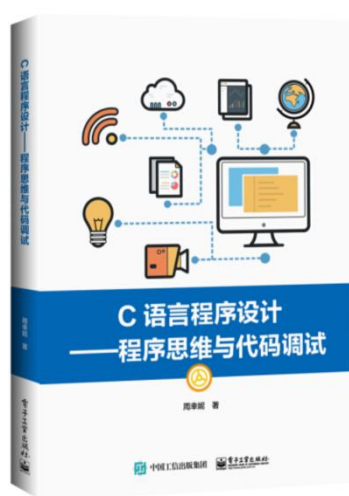
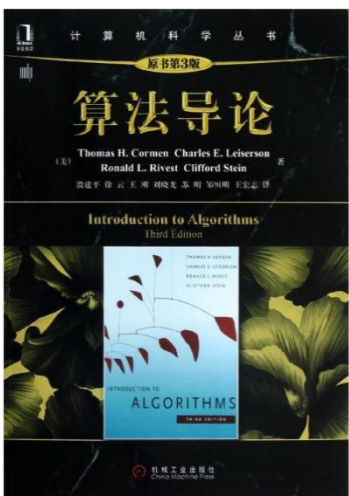
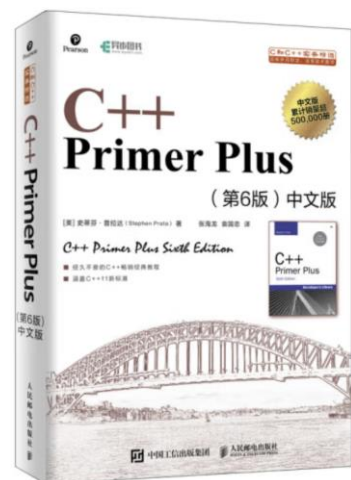
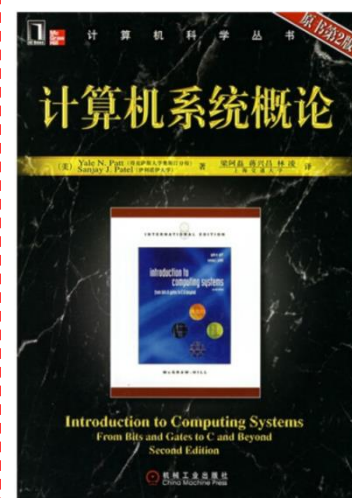
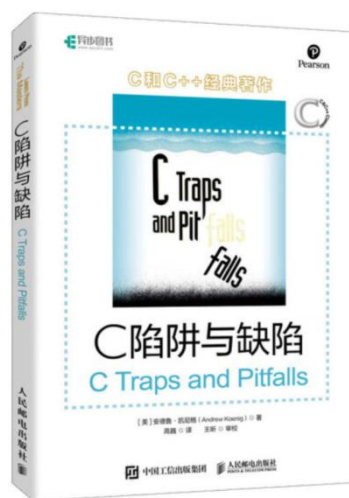
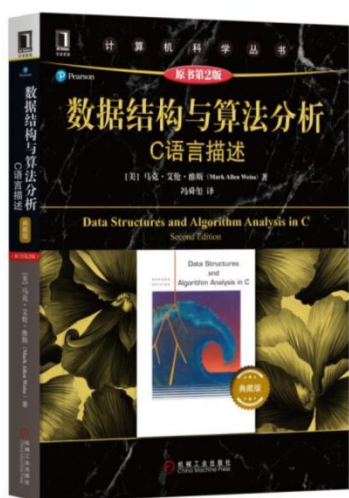
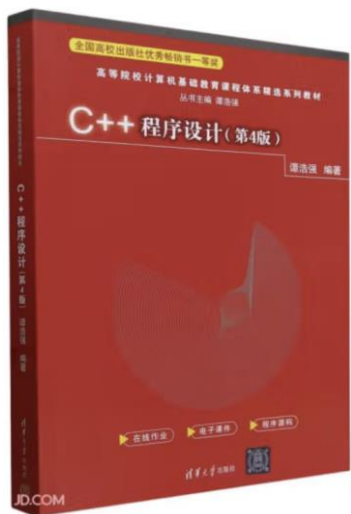
- 附加题，每题2分，按实际得分等比计算
- 直接加到平时成绩中

■ 对抗赛

- 出一道题（包括题面、测试数据，标准程序及注释），小组成员直接+2分
- 按最后得分排位加分：一等奖+4分，二等奖+3分，三等奖+2分，其他+1分
- 加分加到平时成绩中

- 注：加到平时成绩到达100分为止

参考书目



C++语言基础

算法与数据结构

程序错误与调试

程序运行底层原理

课程平台

- 课堂派 (<https://www.ketangpai.com/>)
- YOJ在线评测系统 (<http://yoj.ruc.edu.cn>)
- 其他资源
 - POJ: <https://poj.org/>
 - HDUOJ: <https://acm.hdu.edu.cn/>
 - UOJ: <https://uoj.ac/>
 - CodeForces: <http://codeforces.com/>
 - 洛谷: <https://www.luogu.com.cn/>

YOJ题库

■ 顺序执行与输入输出

- YOJ-4: 读入字符串并输出
- YOJ-6: 浮点数求和
- YOJ-7: 三角形面积
- YOJ-9: 按要求读入数据并输出
- YOJ-10: 温度转换
- YOJ-413: 输出字符的ascii码
- YOJ-446: 整数求和
- YOJ-447: 求几何平均数
- YOJ-448: 按要求读入并输出数据
- YOJ-449: 按要求输出一个整数

YOJ题库

■ 分支与循环

- YOJ-3: 比较两个整数的大小
- YOJ-5: 切换字符大小写
- YOJ-8: 计算学分绩点
- YOJ-495: 双十一购物
- YOJ-502: 公斤与磅的互换
- YOJ-171: 简单计算器
- YOJ-2: 输出Hello world加强版
- YOJ-162: ISBN号码
- YOJ-457: 字母金字塔
- YOJ-180: 计算cos的近似值
- YOJ-459: 求 π 的近似值
- YOJ-256: 计算约数
- YOJ-257: 级数求和
- YOJ-89: 奖金发放
- YOJ-636: 博饼游戏
- YOJ-504: 数字分类
- YOJ-667: 报数

YOJ题库

■ 穷举法

- YOJ-147: 教室排课
- YOJ-148: 傻大木买军火
- YOJ-134: 猜数字
- YOJ-298: 被整除的数
- YOJ-299: 质因数分数
- YOJ-296: 班会时间
- YOJ-414: 水仙花数
- YOJ-661: 移动火柴
- YOJ-492: 加法表达式

YOJ题库

■ 字符串操作

- YOJ-497: 最大跨距
- YOJ-144: 子串替换
- YOJ-159: 字符串编辑
- YOJ-160: 字符串移位包含

YOJ题库

■ 数组操作

- YOJ-483: 数列合并 (简单版)
- YOJ-496: 上升序列
- YOJ-503: 组成最小数
- YOJ-266: 矩阵旋转
- YOJ-498: 变换的矩阵
- YOJ-660: 矩阵局部替换

YOJ题库

■ 统计分析

- YOJ-306: 寻找众数
- YOJ-290: 购物车共同性
- YOJ-201: 银行存取款汇总
- YOJ-468: 统计元音字母频率

YOJ题库

■ 进制与计算

- YOJ-289: 大整数加减
- YOJ-337: 高低位交换
- YOJ-127: 超级细菌
- YOJ-129: 外星数字
- YOJ-152: 确定进制

YOJ题库

■ 排序

- YOJ-486: 碱基串排序
- YOJ-505: 计算中位数
- YOJ-506: 排值日表
- YOJ-668: 变位词
- YOJ-135: 身份证排序

YOJ题库

■ 搜索

- YOJ-113: n皇后
- YOJ-117: 摘桃子
- YOJ-122: 迷宫
- YOJ-124: 滑雪
- YOJ-126: travel
- YOJ-163: 最大岛屿
- YOJ-202: 分数问题
- YOJ-262: 24点 (整除版)
- YOJ-263: 24点 (非整除版)
- YOJ-268: 穷游一族
- YOJ-437: 寻找一卡通
- YOJ-513: 加法表达式 (进阶版)

YOJ题库

■ 面向对象编程

- YOJ-344. 三角形
- YOJ-350. 完善student类的构造函数
- YOJ-605. 约瑟夫问题（面向对象版本）
- YOJ-338. 重载函数getpower
- YOJ-346. 设计复数类
- YOJ-421. C++复数运算符重载（+与<<）
- YOJ-606. 数据共享保护与操作符重载
- YOJ-789. 数据共享保护与操作符重载2
- YOJ-426 矩形相加
- YOJ-607 命名空间-变量与函数
- YOJ-411 完善Professor类

YOJ题库

■ 面向对象编程

- YOJ-382 计算工资系统（继承和派生）
- YOJ-397 商品的销售总款和平均售价
- YOJ-401 Shape
- YOJ-618. string与IO (1)
- YOJ-619. string与IO (2)
- YOJ-620. string与IO (3)
- YOJ-621. string与IO (4)
- YOJ-399 三维数组类模板

YOJ题库

■ 面向对象编程

- YOJ-622 C++模板 (1)
- YOJ-623 C++模板 (2)
- YOJ-624 C++模板 (3)
- YOJ-625 C++模板 (4)
- YOJ-626. 模板与迭代器
- YOJ-627. 模板与链式迭代器
- YOJ-628. STL容器练习 (1)
- YOJ-629. STL容器练习 (2)
- YOJ-630. STL容器集合处理