

# Catelog

sis0:命令行编译 .....	2
sis1:基础 IO 操作 .....	3
sis2:UID 管理器 .....	4
sis3:学生信息管理器 .....	5
sis4:实现一个线段树泛型模板 .....	6
sis5:实现一个 trie(字典树)模板 .....	7
sis6:实现对网络最大流与最小费用流问题的 solution 类 .....	8
sis7:实现一个高性能的基础正则表达式匹配 solution 类 .....	9
sis8:实现一个瘟疫传播的可视化模拟 .....	10
sis9:实现一个较为复杂的线性代数计算器 .....	11

**sis0:命令行编译**

## sis1:基础 IO 操作

使用了 `Scanner` 实现了输入一个给定长度的差分数组，求出其原数组，即对数组进行逐项求和。

## sis2:UID 管理器

提供三种类型的 UID 生成、申请与维护。

1. 基于日期的 UID，例如 2022101149
2. 完全无序的类激活码，例如 AA2CA-STBS3-AED2P-RDGSSP
3. 按顺序发放的序列，可选固定位数，例如 00001,00002

可以保证所有 UID 不会重复。类中储存所有 UID 对应的引用。申请复杂度  $O(1)$ ，引用复杂度  $O(\log(n))$

测试用例的控制台规则：

中括号内为需要填入字符串

尖括号为可选参数，默认为列表第一个

<code>getUID &lt;type: "date"   "code"   "seq"&gt; [name]</code>	申请对应类型的新 UID，并与一个引用名称 <code>name</code> 绑定
<code>secUID &lt;type: "date"   "code"   "seq"&gt; [UID]</code>	查找 UID 对应的引用名称

# sis3:学生信息管理器

提供了三个类，其中 SchoolLib 为主类，提供了对学生以下操作：

- 1. 添加学生并自动创建学号
- 2. 添加课程并自动生成课程编号
- 3. 学生选课
- 4. 记录考试成绩，补考成绩
- 5. 与平时分加权，计算总评成绩
- 6. 计算绩点
- 7. 以绩点、学号、姓名字典序等对学生排序

保证所有操作线程安全，必要的地方进行了异常处理与保护性拷贝。

测试用例的控制台规则：

中括号内为需要填入字符串

尖括号为可选参数，默认为列表第一个

<code>addStu [student name]</code>	添加学生
<code>addCur [Course name] [teacher name] [credit]</code>	添加课程
<code>celCur [student UID] [Course name]</code>	选课
<code>showStu &lt;sort with:"UID"   "name"   "point"&gt;</code>	显示学生列表
<code>showCur &lt;sort with:"UID"   "Course name"   "teacher name"&gt;</code>	显示课程列表
<code>addScore [student UID] [Course name] [Normally score] [Exam score]</code>	添加成绩
<code>addResc [student UID] [Course name] [Exam score]</code>	添加补考成绩
<code>cacu</code>	计算总评成绩与绩点

## sis4:实现一个线段树泛型模板

线段树是一种较为复杂的数据结构，旨在快速解决区间数据批量修改与特征统计。

本类实现了一个可以批量地对数据进行线性空间内加和运算的线段树，统计内容为区间内的最大值，对于每个操作：

1. 修改单点：时间复杂度为 $O(\log n)$
2. 修改区间：均匀修改与查询后最坏时间复杂度为每点渐进 $O(\log(nm))$ ， $n$  为内容总数， $m$  为修改区间长度
3. 查询区间： $O(\log n)$

**sis5:实现一个 trie(字典树)模板**

## **sis6:实现对网络最大流与最小费用流问题的 solution 类**

最大流问题叙述略



## **sis7:实现一个高性能的基础正则表达式匹配 solution 类**

### 正则表达式

**sis8:实现一个瘟疫传播的可视化模拟**

sis9:实现一个较为复杂的线性代数计算器

myLinearEntire	线性空间元素	接口：具有线性性质的基础运算
myLinearSpace	线性空间	1. 基础运算 2. 基 3. 计算秩 4. 对应矩阵
myMatrix	矩阵	1. 基础运算 2. 行列式 3. 对角化
myPolynomial	多项式	1. 基础运算 2. 求值 3. 牛顿迭代法解特殊解
myRealNum	实数	1. 基础运算

实现的功能：控制台指令如下： 中括号内为需要填入字符串  
尖括号可为可选参数，默认为列表第一个

<code>addMat [columns] [rows] [digit(1,1)] ... [digit(c,r)]</code>	添加一个列数为 columns，行数为 rows 的矩阵
<code>addPol [rank] [digit 1] ... [digit r]</code>	添加一个阶数为 r 的多项式
<code>addLS [rank] [LE name 1] ... [LE name r]</code>	添加一个以 LE 1~r 为基的线性空间
<code>show &lt;scope : "all"   "Mat"   "Pol"   "LS"&gt;</code>	列出对应的所有元素
<code>cacuMat [Mat name] &lt;type : "Det"   "Eig"&gt;</code>	计算矩阵的行列式和对角化
<code>cacuPol [Pol name] [digit]</code>	计算多项式的值
<code>solvePol [Pol name] [digit]</code>	求多项式等于输入数字时的一个数值解
<code>op [LE name1] &lt;operator : "+"   "- "   "*"&gt; [LE name2]</code>	对线性空间元素进行基础运算