

# Project 验收细则

Project 的具体要求分为代码风格、功能测试、课堂展示、实验报告四个部分，本文档主要说明代码风格和功能测试。

代码风格，指源代码的书写风格和组织管理，体现在命名，注释，缩进，布局，版本控制等。功能测试指是否能够完成指定任务，以及运行结果是否符合预期。

验收的大致流程是：

- 1. 检视代码风格；
- 2. 进行功能测试。

下面介绍具体规则。

## 1 代码风格

必须使用 `git` 进行版本控制。版本控制应当贯穿实验的整个开发过程，完成实验后再 `git init` 的、只有零星 `commit` 的不视为使用了版本控制。注意，任何 `commit` 中出现了抄袭的代码，均判定为抄袭。

代码风格的考察重点在于可读性。可读性是指其他人是否能够通过代码快速理解编码者的意图。有意义的命名、合理的模块设计、适当的注释、较强的一致性可以增强可读性。一致性是指编码时采用的格式细节是否始终保持不变，强一致性会降低阅读的难度，增加可读性。建议自行搜索你所使用的语言的编码规范作为参考。

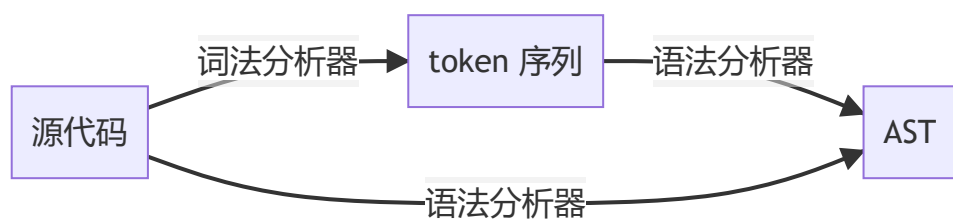
### 1.1 评价标准

得分	标准
D	没有上交代码或者被判抄袭。
C	没有参加验收，但上交了代码。
B	没有使用 <code>git</code> 或者代码难以理解。
B+	使用了 <code>git</code> 且代码较为混乱。
A	使用了 <code>git</code> 且代码基本可读。

得分	标准
A+	使用了 git 且代码风格良好。

## 2 功能测试

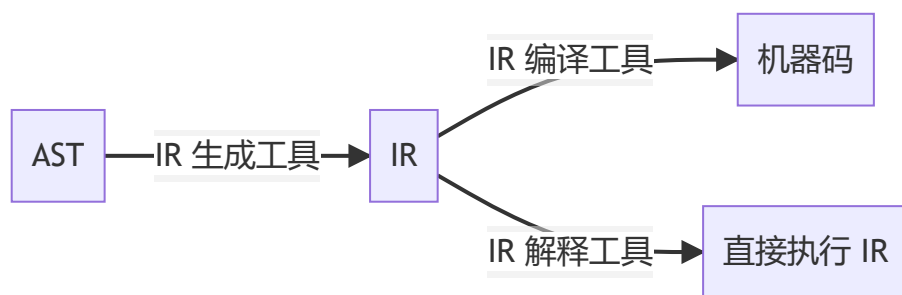
### 2.1 基本要求



实现一个编译器，一般是从词法分析器和语法分析器开始。上图展示了可选的技术路径：

1. 使用词法分析器（手写的或工具生成的）将源代码转换成 token 序列，然后使用语法分析器（手写的或工具生成的）将 token 序列转换成 AST。
2. 使用语法分析器（手写的或工具生成的）直接将源代码转换为 AST。

手写一个词法分析器非常简单，但是手写语法分析器可能就困难一些。常用的分析器生成工具有 lex、yacc 等。某些工具有非常强大的功能，生成出来的分析器能够提供更多的错误信息或者具有更强的健壮性，你也可以选择它们。

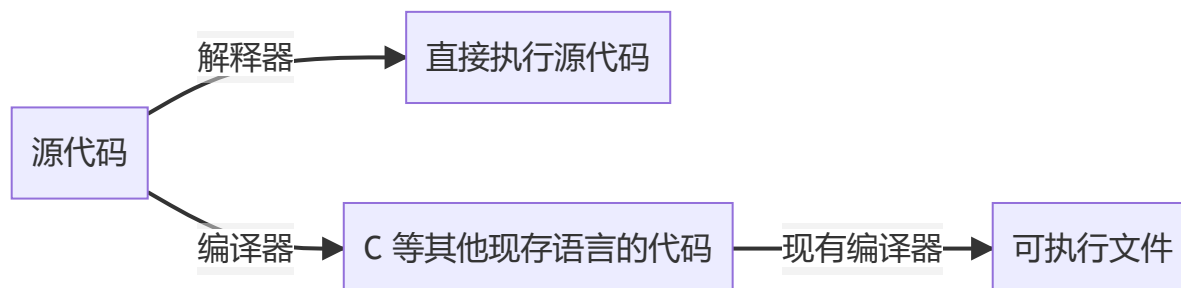


从 AST 出发，选择就非常多样了。以下提供几种可能的思路：

1. 完全使用 LLVM 等框架，使用 LLVM 将 AST 转换为 IR，然后使用 LLVM 将 IR 转换为机器码来运行。这种方法的缺点在于需要学习新的知识，查看一些文档（大部分是英文的）；优点是掌握 LLVM 之后工作量大幅降低了。
2. 自己将 AST 转换为 LLVM 标准的 IR，然后使用 LLVM 将 IR 转化为机器码或者使用 LLVM 执行 IR。这种方法的优点是不需要看 LLVM 的文档了，有关 IR 的文档短得多；缺点在于自己写的生成器可能附带大量 bug。

3. 自己将 AST 转换为 IR，然后另外实现一个 IR 解释器来运行这个 IR。这种方法的优点是不需要阅读任何文档，缺点是调试时间大幅增加。

以下是严禁使用的技术路线：



本课程要求实现的必须是一个编译器，而不是解释器；另外，不允许将源代码编译成一种高级语言，然后调用已有的编译器来编译它。本课程要求生成中间代码，至于中间代码之后的部分不做要求。你可以将其转化为机器码，也可以写一个解释器来运行它（在不考虑性能的情况下，使用 Python 写一个 IR 解释器是很简单的）。

总而言之，你的编译器必须生成一个能以某种方式执行的“结果”，这个结果必须能在现代操作系统的常见架构上运行。

## 2.2 语法定义

可以自己设计一种新语言，也可以实现现有语言的子集，也可以混合你喜欢的多种语言。但是，设计出的语言必须是现代高级语言，不能体现出任何与底层硬件或者体系架构相关的设计，比如这个[反例](#)以及相似的更多[晦涩编程语言](#)。

## 2.3 测试点

需要使用设计出的语言解决以下问题。所有问题的输入中只会使用 ASCII 码表中的可打印字符，并且本文中的整数是指匹配正则表达式  $-(0|[1-9][0-9]*)$  的字符串。

### 2.3.1 快速排序

以下是排序问题的定义：

输入：第一行是一个整数  $N$ ，满足  $0 \leq N \leq 10^4$ 。接下来是  $N$  个整数，每个整数独占一行。整数  $x_i$  满足  $-10^4 < x_i < 10^4$  但不保证  $i \neq j$  时  $x_i \neq x_j$ 。

输出：要求将输入的  $N$  个整数按非降序排序。计算结束后，需要将  $N$  个整数依次输出，每个数字独占一行。注意最后一个数字输出后必须换行。

要求必须使用[快速排序](#)算法解决该问题，并且必须使用递归形式的实现。

例一

输入：

0

输出：

例二

输入：

4  
3  
2  
1  
1

输出：

1  
1  
2  
3

例三

输入：

3  
1  
0  
-1

输出：

-1  
0  
1

2.3.2 矩阵乘法

以下是矩阵乘法问题的定义：

输入：两个矩阵  $A$  和  $B$  。每个矩阵输入的第一行是单个空格隔开的两个整数  $M$  和  $N$  ， 满足  $1 \leq M, N \leq 25$  ， 分别代表了矩阵的行数和列数。接下来  $M$  行， 每行是  $N$  个整数， 整数之间使用一个以上空格隔开， 并满足整数的位数（负号算一位）加上空格的数量等于 6 。 整数全部在区间  $(-1000, 1000)$  中。不保证  $A$  和  $B$  之间可以进行乘法操作。

输出：如果  $A$  和  $B$  的维度不满足乘法的要求（即  $N_A \neq M_B$  ）， 则输出 `Incompatible Dimensions` ， 该信息独占一行（输出后需要换行）； 否则， 需要计算：

$$C = \begin{bmatrix} c_{0,0} & \dots & c_{0,N_C-1} \\ c_{1,0} & \dots & c_{1,N_C-1} \\ \dots & \dots & \dots \\ c_{M_C-1,0} & \dots & c_{M_C-1,N_C-1} \end{bmatrix} = AB$$
$$= \begin{bmatrix} a_{0,0} & \dots & a_{0,N_A-1} \\ a_{1,0} & \dots & a_{1,N_A-1} \\ \dots & \dots & \dots \\ a_{M_A-1,0} & \dots & a_{M_A-1,N_A-1} \end{bmatrix} \begin{bmatrix} b_{0,0} & \dots & b_{0,N_B-1} \\ b_{1,0} & \dots & b_{1,N_B-1} \\ \dots & \dots & \dots \\ b_{M_B-1,0} & \dots & b_{M_B-1,N_B-1} \end{bmatrix}$$

其中：

$$c_{i,j} = \sum_{k=0}^{N_A-1} a_{i,k} b_{k,j}$$

计算完成后需要输出结果。结果分为  $M_C$  行输出， 每一行有  $N_C$  个整数， 每个整数前有数个空格， 并满足整数的位数（负号算一位）加上空格的数量等于 10 。

例一

输入：

```
2 2
  1   2
  3   4
3 1
  1
  2
  3
```

输出：

```
Incompatible Dimensions
```

例二

输入：

```
2 2
  1 2
  3 4
2 3
  1 2 0
  0 3 4
```

输出：

```
1      8      8
3     18     16
```

例三

输入：

```
4 1
 123
 345
 567
 789
1 4
-123 345 -567 789
```

输出：

```
-15129    42435    -69741    97047
-42435    119025   -195615   272205
-69741    195615   -321489   447363
-97047    272205   -447363   622521
```

2.3.3 选课助手

选课助手问题的定义如下：

输入：输入是一个培养方案，每行一门课程，空行代表输入结束。每门课程是一个四元组  $C =$  (课程名称, 学分, 前置课程, 成绩)。每个字段可能包含任意长度的字符串，但不会有前置和后置的空白符。各个字段之间使用 | 隔开。

- 课程名称 是不包含 , 、 ; 和 | 的长度大于 0 小于 5 的字符串。
- 学分 是非负整数, 大于 0 小于 5。

- **前置课程** 是修读本门课程时必须已经获得学分的课程。 ， 相当于逻辑操作 and ， ； 相当于逻辑操作 or 。 ， 的优先级高于 ； 。 A,B;C,D 的含义是 A 并且 B ， 或者 C 并且 D 。使用类似 C 语言的方式描述就是 A && B || C && D 。注意前置课程不一定存在于培养方案中。
- **成绩** 是能够匹配正则表达式 [ABCDF]? 的字符串。当本字段为空时，代表还未修读本课程；当成绩为 F 时，代表曾经修读但是挂科，没有获得学分；其他情况代表成功获得学分。

输出：需要计算以下内容。

- GPA ，使用四分制，其中 A 等于 4 ， B 等于 3 ， C 等于 2 ， D 等于 1 ， F 等于 0 ，结果保留一位小数，格式匹配正则表达式 [0-3]\.[0-9]|4\.0 。计算公式为： $GPA = (\sum \text{成绩} \times \text{学分}) / \text{尝试学分}$ 。
- 尝试学分，已经获得成绩的课程的总学分。包括获得 F 成绩的课程。
- 已修学分，已经获得的总学分。成绩为 F 的课程没有获得学分。
- 剩余学分，培养方案中还有多少学分没有修读，包括成绩为 F 的课程。
- 推荐课程，满足前置课程条件，可以修读的课程但还没有获得学分的课程。必须按照课程出现在输入中的先后顺序进行输出。

提示点：

1. 如果前置课程不存在于该培养方案中，应该当作是未修读的课程。假如培养方案只有一条  $x|2|y|$  ， y 不在培养方案中，当作没有修读的课程，所以 x 不满足前置课程条件，不进行推荐。由于 y 的前置课程是未知的，所以也不需要推荐 y 。
2. 注意有空的培养方案，此时输入是一个换行。
3. 随机测试案例的生成过程大致如下：在 100 门课（课的名称是类似 c2 、 c98 的编号式命名）之间随机添加依赖关系，保证没有环路，保证每门课的前置课程不会超过 7 组（由 ； 分隔的部分是一组），每一组中不会超过 7 门。
4. 不保证某课程所依赖的课程一定出现在它的前面。举个例子，可能先有  $x|2|y|$  ，再有  $y|3||$  。

输出的具体细节必须严格按照以下示例中的格式。

## 例一

输入：

```
c0|3||
c1|4|c0|
c2|3|c0|
c3|4|c1|
c4|4|c2|
c5|3|c3|
c6|3|c2|
```

输出：

```
GPA: 0.0
Hours Attempted: 0
Hours Completed: 0
Credits Remaining: 24
```

```
Possible Courses to Take Next
c0
```

## 例二

假如某学生已经修读了培养方案中的所有课程，那么可能的输出是：

```
GPA: 3.5
Hours Attempted: 129
Hours Completed: 129
Credits Remaining: 0
```

```
Possible Courses to Take Next
None - Congratulations!
```

注意！只有在未修学分等于 0 时，才输出 None - Congratulations! 。如果未修学分不为 0，但是没有可以推荐的课程，那么应该输出：

```
GPA: 3.5
Hours Attempted: 129
Hours Completed: 129
Credits Remaining: 8
```

```
Possible Courses to Take Next
```

## 例三

输入：



c53|2||B  
c40|5|c28,c26,c9,c25,c23,c38,c17;c30,c6,c9,c10,c28,c18;c2,c8,c3;c38,c18,c27;c18,c23,c38,c20,c39,c0,c32|B  
c34|4|c17;c20,c16,c21,c9,c25;c3,c1,c23,c31,c2,c20|C  
c18|3|c7,c3,c2,c13,c0,c15,c1;c6,c10,c14;c13;c17,c8,c0,c5,c13;c11,c2,c12;c1,c8,c7,c6|F  
c33|4|c31,c4,c12,c15,c20,c28;c24,c28,c21,c12;c24;c25,c31,c22,c14;c13,c25,c23,c31,c15;c3,c2,c28,c5,c30,c16|C  
c47|5|c11,c0;c20,c11,c32;c22,c39,c34;c5,c42,c3|C  
c15|2||C  
c81|1|c33,c6,c34,c16,c10,c72;c1,c59,c18,c58,c11;c30,c26,c45,c40;c11,c67,c60,c74,c59|A  
c12|5|c10,c2,c3,c5,c1,c0,c7|  
c67|1|c50,c39,c51,c63,c40|C  
c80|4|c24,c43|B  
c87|1|c9,c42,c39,c57;c72,c27,c2,c42,c13,c63;c21,c57;c41|  
c0|5||B  
c49|2||A  
c85|5|c51,c70,c59,c69,c45,c29;c61,c53,c32,c13,c24,c18,c58;c8,c20,c22,c36;c8;c40,c42,c62,c43,c19|  
c98|4|c65,c3,c96,c33,c46|F  
c64|5|c7,c13,c51,c12;c44,c10;c44,c5,c57,c42,c53;c43,c44,c46,c29,c5,c21,c49;c56,c29,c44,c10,c55,c22,c9;c38,c39,c8,  
c55|1||A  
c82|2|c32,c76,c26,c49,c44,c80|B  
c23|4|c9,c1;c18,c5,c12,c7,c16,c1,c3;c0,c9;c22,c17,c1;c22,c12,c10|  
c63|3|c40,c29;c36,c29,c47,c23,c48,c13,c32;c55,c39;c16,c45|D  
c89|1||A  
c39|1|c17,c26,c6,c9,c29,c3;c2,c28,c15,c38,c13,c29;c18,c1,c38,c33,c24;c31,c17,c22,c21|D  
c90|2||  
c59|2|c1,c41,c5;c55,c44,c36,c39,c46,c4;c29,c52,c7,c40,c6,c26|D  
c25|4|c4,c2;c18,c7,c2,c6,c20,c9,c22;c3,c21,c8,c13,c9|D  
c75|3|c41,c45;c6,c4,c22,c3|D  
c66|2|c21,c56,c30,c62,c33,c38,c9;c57,c28,c16,c53,c46;c12,c11,c59,c0,c34;c0;c13,c6,c20,c64,c63;c18,c50,c59,c9,c53,  
c38|5|c5,c33,c29,c35,c31;c20,c9,c16,c30;c11,c29,c24,c2|C  
c2|1||A  
c74|2|c48,c4,c60,c42,c29,c1;c21;c54,c24,c45,c73,c49;c65,c4|F  
c48|3||C  
c65|2|c9,c47,c57,c6,c19|  
c21|3|c17,c7,c18,c19|C  
c27|4|c7,c8,c25,c2;c20,c3,c19,c8,c7;c22,c23,c0,c18,c9;c24,c13,c2;c7;c26,c22,c0,c13,c2,c15,c5|D  
c61|3|c6,c55,c25;c49,c29,c45,c15,c14;c3,c19,c0,c37,c26,c27,c53|  
c72|3||C  
c45|3||  
c91|3|c37;c54,c34;c45,c57,c38,c83,c49,c71,c54;c3,c0,c77,c71,c1;c4,c56,c22,c40,c13,c57;c16|C  
c73|5|c42,c62,c22,c14,c9;c17,c35,c2|D  
c3|1||C  
c42|2|c29,c28,c34;c37,c26,c27,c8,c9;c24,c19,c31,c21,c8,c0,c32;c34,c25,c38|C  
c36|4|c28,c16,c15,c25,c33|A  
c96|2|c73,c38,c50;c26,c84,c5,c61,c12,c20;c44;c65,c36|B  
c88|5|c0,c35,c40,c9,c39;c80,c51,c14,c72,c77,c8,c57;c67,c38,c63,c31,c0|F  
c22|2|c16,c9,c4,c5;c6,c15,c19|  
c41|4|c21,c14,c32,c8,c1;c31,c5,c17,c36,c26;c38,c13,c31;c32,c13,c27,c0;c6,c36|D  
c35|5|c6,c29,c9,c32,c22;c23,c9,c22,c30,c17,c20,c19;c24,c19,c27,c10,c32,c26;c12,c30,c5;c26,c34,c27,c0;c21,c13,c26,  
c46|2|c21,c4,c7,c34;c5;c35;c18,c17,c11;c18,c14,c35,c1,c25,c19|  
c92|1|c41,c52,c81,c66,c9,c0,c68|D

c9|5|c1,c4,c6,c3;c0,c6,c1,c8;c7,c4,c1;c6,c3,c5;c8,c1,c6,c5,c0;c2,c4,c0,c5,c7,c1|  
c11|5||C  
c94|4||D  
c29|4|c16|  
c93|3|c11,c5,c16,c43,c49,c48;c36,c48,c45,c83,c27,c84;c60,c83,c17,c0,c48,c27,c59;c12,c56|B  
c57|2|c42,c44,c34;c41,c26,c3,c17;c43,c15,c33,c2,c39|A  
c70|3|c58,c59,c0,c49,c54;c43,c3,c39,c2,c11,c13;c53,c18,c50,c47|D  
c78|2||  
c56|3|c31,c24,c43,c0,c9,c21,c12;c23,c16,c41,c26,c19,c45;c32,c11,c13,c28,c2,c44,c24;c15,c43,c27,c45,c38,c2;c22,c23|  
c58|2|c1,c13;c44,c16,c40,c51,c28|F  
c8|3|c6,c0,c3,c7,c1,c2;c3,c7,c1;c6,c1;c5,c2|A  
c7|4|c5,c6;c4,c3,c0,c5,c1,c2;c1,c5;c0,c3,c2,c5,c6;c5;c1|  
c5|2||D  
c28|3|c23,c4|A  
c68|1|c23,c36,c51;c13,c44,c60|F  
c13|5|c3,c7,c1,c12,c9,c6,c4;c7,c2;c10,c5,c0,c8|C  
c1|1||B  
c37|5|c30|B  
c71|2|c5,c40,c61,c60,c9,c37,c35;c32,c30,c4,c49,c0,c12;c18,c43,c47;c58,c17,c59,c56,c66,c69,c29|  
c6|5||  
c77|1|c47,c3,c50;c49,c37,c34|  
c19|2|c2,c15,c4;c13,c10,c15,c14|A  
c32|2||F  
c26|1|c0,c25,c13,c15,c12,c9,c17|F  
c43|2|c11,c22,c34,c25;c41,c25,c0|F  
c44|1|c0,c31,c30,c26,c11,c7,c17;c7,c16|D  
c95|3|c73,c75,c51,c69,c21,c22;c68,c75,c55,c6;c52;c48,c32,c80,c57,c68,c5,c62;c68,c50,c93,c77,c40|A  
c20|2||D  
c97|5|c61,c67|F  
c16|3|c0,c13,c9,c8|D  
c62|4|c44,c14,c15,c27,c18,c47,c53;c13,c51,c33;c56,c54,c46,c16,c32;c17,c23,c27,c7,c1,c32;c61,c51,c56,c44,c40,c9;c2|  
c86|2|c32,c10,c63,c26;c32,c11,c67,c85,c42,c58,c34;c27,c46;c31,c9,c24,c76,c79,c49,c39|F  
c52|1|c10;c43,c34,c26,c50,c12;c27,c28,c36,c38,c14;c6,c8,c49,c39|  
c31|1|c26,c23|C  
c17|4|c8,c4,c3,c10,c15;c0,c5,c10,c6;c5,c16,c2,c10,c7,c9|A  
c30|1||D  
c4|1||B  
c99|4|c96,c86,c41,c7,c70;c67,c9,c37,c40,c3,c65,c92;c34,c22,c57,c61,c37;c9,c6,c22,c1,c82,c81;c71|  
c84|5|c33,c1,c30,c74;c34;c69,c65,c15;c7;c43,c69,c0,c44;c55|B  
c79|4|c48,c65,c59;c16,c25,c66|F  
c51|3|c21,c17,c48,c39,c11,c9;c26,c21,c8,c36,c39;c28;c41,c50,c23,c42,c21,c8,c3;c9|D  
c76|1|c46,c66,c41,c26,c70,c69;c74,c69,c49,c59,c17,c68,c30;c17,c29,c3,c33,c25,c21,c65;c72,c75,c50,c0;c49,c20,c23,c2|  
c50|2|c34,c40,c6;c6,c23,c26,c34,c21,c36,c19;c27,c1,c49|B  
c83|2|c22,c36,c72;c15,c79,c36;c45,c6,c46,c72,c23,c31;c60,c31,c48,c56;c78,c38,c57,c54;c61,c15,c29,c46,c17|C  
c69|1|c14,c54;c1,c61,c0;c51,c22,c25,c49,c46,c13;c44,c20,c29,c67,c11;c32,c61,c42,c68,c67,c2,c62;c42,c31,c51,c24|D  
c10|1|c6,c9,c8;c1,c0,c6,c8,c3;c8,c7,c1,c3,c4,c6|A  
c14|5|c2,c6,c8,c4,c3,c9;c4,c6,c12,c5,c1;c11|D  
c24|3|c18,c9,c11,c7,c10,c8;c21;c18,c21,c2,c12;c21,c17,c12|  
c54|4|c8;c34;c28,c53,c24;c13,c30|C  
c60|2|c48,c52;c47,c34,c53;c0,c40,c55,c22;c42;c5,c19;c42,c23,c2,c6|B

输出：

GPA: 1.9  
Hours Attempted: 220  
Hours Completed: 185  
Credits Remaining: 100

Possible Courses to Take Next

- c18
- c87
- c85
- c90
- c66
- c74
- c45
- c88
- c46
- c29
- c78
- c58
- c7
- c68
- c6
- c77
- c32
- c43
- c52
- c24

2.3.4 测试工具

以上三个问题通过位于 `tester` 文件夹中的测试工具进行测试，如果没有你的平台能够使用的测试工具，请及时联系助教。目前提供以下版本：

操作系统	体系架构
Linux	amd64
MacOS	amd64/arm64
Windows	amd64

假设你使用 Windows 操作系统和 amd64 架构的计算机，并且使用设计好的编译器将快速排序的源代码编译成了 `qs.exe`，那么可以在命令行中通过如下命令使用测试工具：

`./windows-amd64.exe ./qs.exe`

前一个路径是测试工具的路径，后一个路径是编译好的可执行文件的路径。如果得到如下输出：

```
fixed case 0 (size 0)...pass!
fixed case 1 (size 1)...pass!
fixed case 2 (size 2)...pass!
fixed case 3 (size 2)...pass!
fixed case 4 (size 3)...pass!
fixed case 5 (size 3)...pass!
fixed case 6 (size 3)...pass!
fixed case 7 (size 3)...pass!
fixed case 8 (size 3)...pass!
fixed case 9 (size 4)...pass!
fixed case 10 (size 9)...pass!
fixed case 11 (size 9)...pass!
fixed case 12 (size 10000)...pass!
fixed case 13 (size 10000)...pass!
fixed case 14 (size 4096)...pass!
randomly generated case 0 (size 10000)...pass!
randomly generated case 1 (size 10000)...pass!
randomly generated case 2 (size 10000)...pass!
randomly generated case 3 (size 10000)...pass!
randomly generated case 4 (size 10000)...pass!
randomly generated case 5 (size 10000)...pass!
randomly generated case 6 (size 10000)...pass!
randomly generated case 7 (size 10000)...pass!
randomly generated case 8 (size 10000)...pass!
randomly generated case 9 (size 10000)...pass!
-----
2021-06-05 13:29:47.868
```

那么说明通过了测试。注意测试工具中使用了随机生成的测试数据，因此多次测试的结果可能有所不同。建议多次测试保证能够成功通过测试，避免验收时抽奖。

如果你不能直接生成一个可执行文件，而是需要多个命令才能最终执行编译结果，那么你需要把所有命令写在一个脚本中，例如 `qs.sh` 或者 `qs.bat`（根据操作系统而不同），然后测试该脚本即可。

```
./darwin-arm64 ./qs.sh
```

使用 LLVM 时主函数必须返回 0，否则可能会出现如下错误：

```
panic: error happened while waiting for given program
error message:
exit status 1
...
```

## 2.4 进阶主题

除了上述基本功能外，你还可以探索一些进阶主题。根据实现难度不同，可以获得一些额外奖励。

- 复杂的数据类型，比如结构体、变长数组、字典等；
- 错误恢复；
- 宏展开；
- 函数式；
- 循环优化；
- 面向对象；
- .....

## 2.5 评价标准

得分	标准
D	没有上交代码或者被判抄袭。
C	没有参加验收，但上交了代码。
B	参加了验收，但是没有通过任何测试点。
B+	通过某个测试点。
A	通过两个测试点。
A+	通过全部测试点。

进阶主题可以使你在基础得分上获得一些额外奖励。换句话说，通过的测试点数量决定了得分档位（一个确定的区间），然后进阶主题可以使你更靠近这个区间的最大值。所以通过完成进阶主题获得分数的性价比是受到严格限制的。建议先完成测试点，有余力的同学可以试试进阶主题。

### 2.5.1 人数校正

1 人队伍：通过全部测试程序后，成功实现的进阶主题数量加一。