# 编译课程设计高难度题目检查要求及中间代码格式1.0

1. 高难度题目检查要求
2. 选择高难度题目的同学，要求在期末检查现场，在教师或者助教的监督下，只能使用个人已经提交的代码，调通优化测试程序，并按照以下要求生成中间代码、目标代码以及运行结果，并保存在文件中提交。

* 学号\_姓名\_优化前中间代码.TXT
* 学号\_姓名\_优化后中间代码.TXT
* 学号\_姓名\_优化前目标代码.TXT
* 学号\_姓名\_优化后目标代码.TXT
* 学号\_姓名\_优化前运行结果.TXT
* 学号\_姓名\_优化后运行结果.TXT

文件提交后不允许再修改。中间代码和目标代码格式要求见后文。

1. 在期末检查现场，在教师或者助教的监督下，只能使用个人已经提交的代码，调通优化测试程序后，在指定MIPS仿真器上运行测试程序，统计运行效率（仿真器统计的执行指令数量、类别及时钟周期等），根据统计结果进行竞速排名。注意：只有编译后运行正确的程序才有竞速排名资格。
2. 选择高难度题目的同学，要求在期末检查现场回答教师或者助教提出的问题（题目将做成答题卡形式，由学生随机盲选一个），问题主要关注中间代码生成、优化、目标码生成等问题。
3. 申优同学需提交申优文档。
4. 目标代码格式

按照所选体系结构汇编器或仿真器要求生成。

1. 中间代码格式

原则上按照中缀表达式格式输出中间代码，即，形如x = y op z，其中x为结果，y为左操作数，z为右操作数，op为操作符。以下根据基本语法现象举例说明。

* 1. 函数声明

源码形如：

int foo( int a, int b, int c, int d)

中间代码：

int foo()

para int a

para int b

para int c

para int d

* 1. 函数调用

源码形如：

i = tar(x,y)

中间代码：

push x

push y

call tar

i = RET

* 1. 函数返回

源码形如：

return (x)

中间代码：

ret x

* 1. 变量声明

源码形如：

int i, j;

中间代码（符号表信息输出，程序中可不生成真正的中间代码）：

var int i

var int j

* 1. 常数声明

源码形如：

const int c = 10

中间代码（符号表信息输出，程序中可不生成真正的中间代码）：

const int c = 10

* 1. 表达式

源码形如：

x = a \* (b + c)

中间代码（可优化）：

t1 = b + c

t2 = a \* t1

x = t2

* 1. 条件判断

源码形如：

x == y

中间代码：

x == y

* 1. 条件或无条件跳转

中间代码：

GOTO LABEL1 //无条件跳转到LABEL1

BNZ LABEL1 //满足条件跳转到LABEL1

BZ LABEL1 //不满足条件跳转到LABEL1

* 1. 带标号语句

源码形如：

Label\_1: x = a + b

中间代码：

Label\_1 :

x = a + b

* 1. 数组赋值或取值

源码形如：

a[i] = b \* c[j]

中间代码：

t1 = c[j]

t2 = b \* t1

a[i] = t2

* 1. 其他本文档未涉及到的语法现象，或者程序员自行定义的四元式操作，原则上均按照“x = y op z”形式的中缀表达式进行表达。