ICS HW7 ANS

T1

1.

```
INPUT R0 ;x
AND R1, R1, #0
ADD R1, R1, #2 ;y
ADD R2, R1, R1
ADD R2, R2, R1
ADD R2, R2, #1 ;z
AND R3, R3, #0 ;i

LOOP
NOT R4, R2
ADD R4, R4, #1
ADD R4, R3, R4 ;i-z
BRzp LOOP_END

ADD R1, R3, #3 ;y = i + 3

ADD R3, R3, #0
BRn IF_BLOCK
;i>=0
ADD R0, R0, #1 ;x++
BR IF_END

IF_BLOCK;i<0
ADD R0, R0, #-1 ;x--

IF_END
OUTPUT R0

ADD R3, R3, #1 ;i++
BR LOOP

LOOP_END
OUTPUT R0
OUTPUT R1
OUTPUT R2
```

2.

```
INPUT R0 ;x
AND R2, R2, #0
ADD R2, R2, #7 ;z
AND R3, R3, #0 ;i

LOOP
NOT R4, R2
ADD R4, R4 ,#1
ADD R4, R3, R4 ;i-z
BRzp LOOP_END

OUTPUT R0

ADD R3, R3, #1 ;i++
BR LOOP

LOOP_END
ADD R0, R0, R2 ;x+=z
ADD R1, R3, #2 ;y
OUTPUT R0
OUTPUT R1
OUTPUT R2
```

3. 用常量替换已知表达式；删除死代码；删除永远为真或假的条件分支；将循环中不变的计算移到循环外。

T2

1.

(1) 1518

(2) 20

(3) 711

2.

(1) $A\ B + C\ D -*$

(2) $A\ B$ && $C\ D$!&& ||

(3) $A\ B\ C\ D\ E$ ^ $-\ F\ /*+G*H-$

3.

$$n-1=\sum_{i=1}^{m}(opt_i-1)$$

T3

1. 调用位置（返回地址）；所有被调用者保存寄存器；调用者的栈帧指针；局部变量；

2. 每次递归调用都占用栈内存，在递归深度过大时导致栈内存溢出；递推时手动栈保存在堆内存中，堆内存大小远大于栈内存，且手动栈只需保存必要信息。

T4

1.

(a) 42 42

(b) 5 42

2.(a)不需要拷贝数组中的所有元素，且不需要额外的内存来存储两个数组，但拷贝后 a 与 b 实际指向同一数组；(b)需要消耗额外的时间与空间进行拷贝，但完成后数组 a 与数组 b 是分别独立存在的数组，都可以继续使用。

3.是 否

T5

1. 2560

2. 448

T6

通常情况下学生的输出应该是 26213，结构体由于 4 字节对齐而将字符'd'放在整型变量 i 的前面，同时电脑使用小端序存储数据。若使用大端序则结果为 1701183488。在极少数的情况下学生环境的 int 数据不是 32 位，可能会导致其他结果。