SAMPLE TEST CASE:

```
4000    MOVC R0, #1000        /* starts main program */
4004    MOVC R1, #4052        /* address of code starting called function */
4008    MOVC R2, #50
4012    STORE R2, R0, #0      /* Stores R2 at data memory address 1000 */
4016    JALR R10, R1, #0      /* function call */
4020    ADD R2, R2, #1
4024    STORE R2, R0, #4      /* Stores incremented value of R2 at location 1004 */
4028    JALR, R10, R1, #0     /* second call to same function */
4032    ADD R2, R2, #1        /* Increments R2 to value in R2 after function call */
4036    STORE R2, R0, #8.     /* Stores R2 at location 1008 */
4040    ADD R2, R2, #1        /* Increments R2 before 2nd function call */
4044    STORE R2, R0, #12     /* Stores incremented value of R2 at location 1012 */
4048    HALT                  /* main program ends here */
4052    MOVC R11, #2          /* Start of function */
4056    MUL R2, R2, R11.      /* function simply doubles what is in R2 */
4060    JUMP R10, #0          /* Return from function */
4064    MOVC R12, #0          /* this and all following instructions should be squashed */
4068    ADD R13, R12, R12
4072    MOVC R14, # 1012
4076    STORE R13, R14, #0    /* If squashed correctly, should have no impact
4082    HALT
```

SOLUTION:

STATE OF ARCHITECTURAL REGISTER AND DATA MEMORY:

MOVC R0, #1000		R0	1000		D[1000]	50	
MOVC R1, #4052		R1	4052		D[1004]	101	
MOVC R2, #50		R2	204		D[1008]	203	
STORE R2, R0, #0		R3			D[1012]	204	
JALR R10, R1, #0		R4					
ADD R2, R2, #1		R5					
STORE R2, R0, #4		R6					
JALR, R10,R1, #0		R7					
ADD R2, R2, #1		R9					
STORE R2, R0, #8		R10	4032				
ADD R2, R2,#1		R11	2				
STORE R2, R0, #12		R12					
HALT		R13					
MOVC R11, #2		R14					
MUL R2, R2, R11		R15					
JUMP R10, #0							
MOVC R12, #0							
ADD R13, R12, R12							
MOVC R14, # 1012							
STORE R13, R14, #0							
HALT							