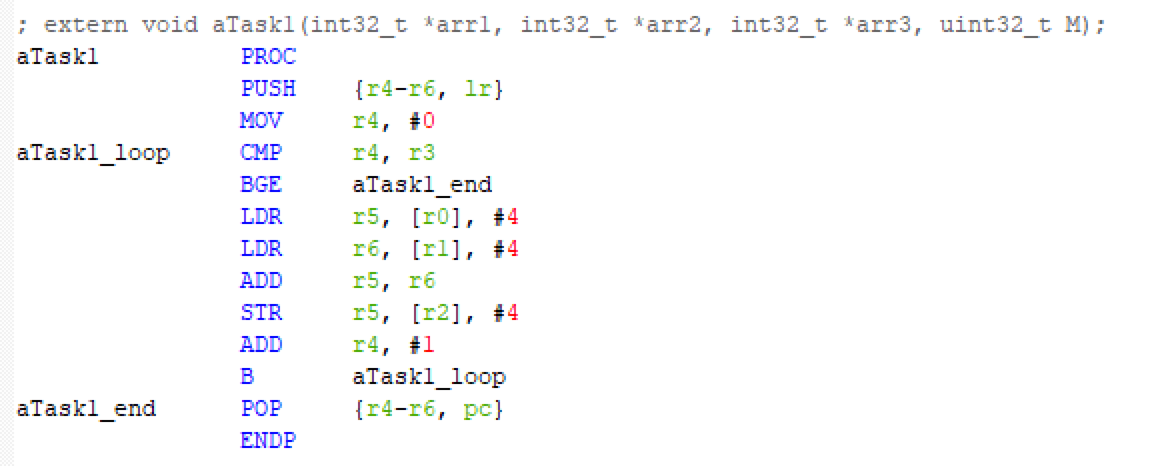
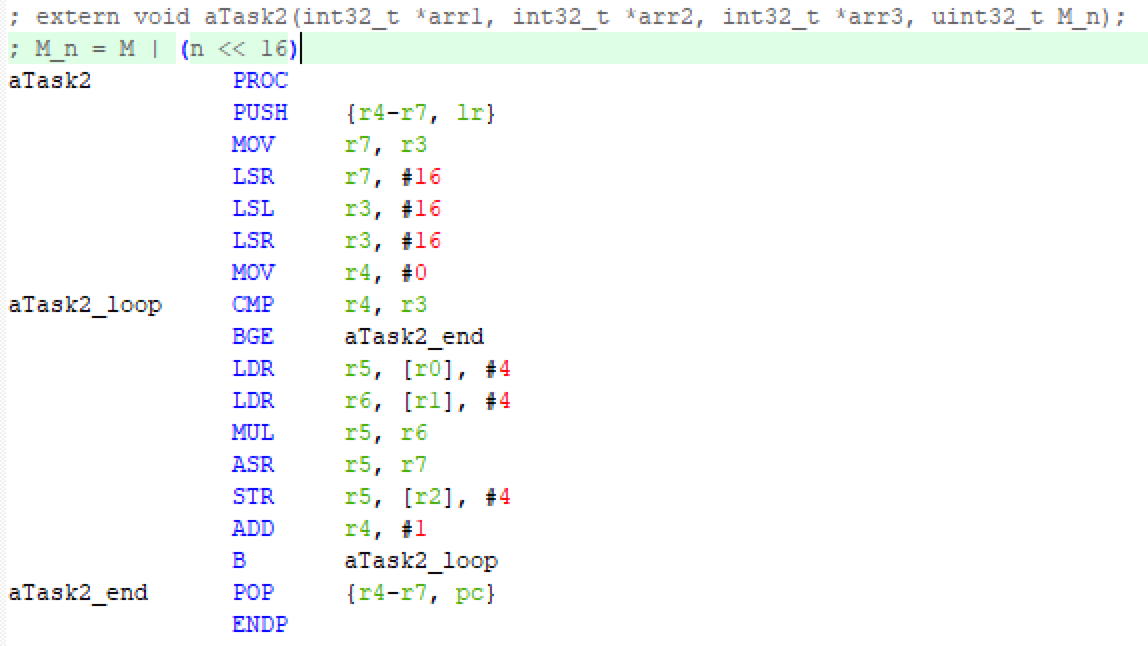
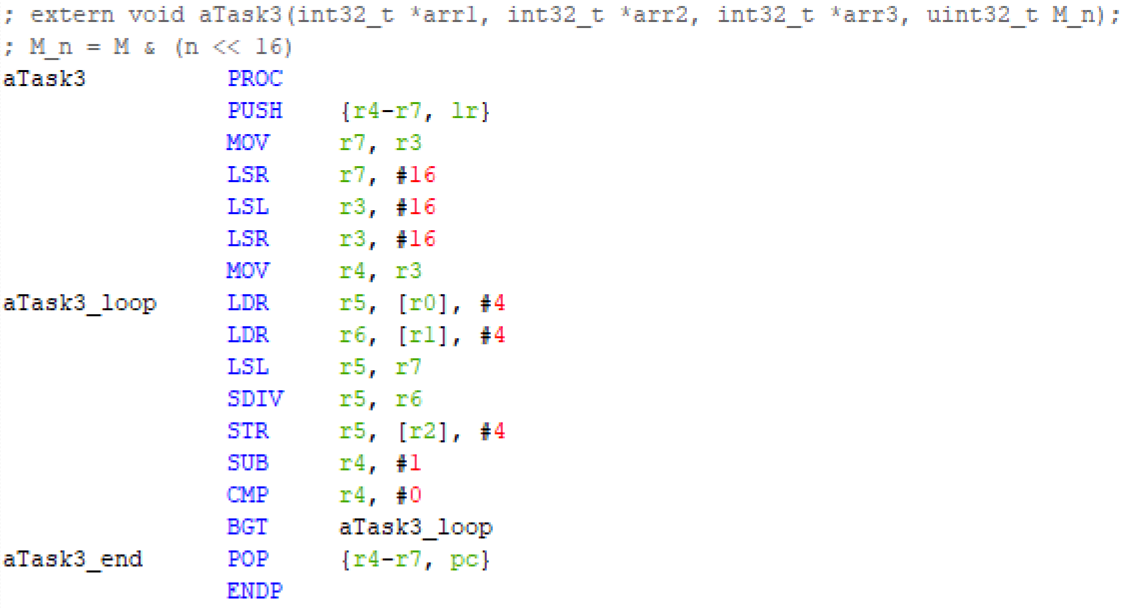


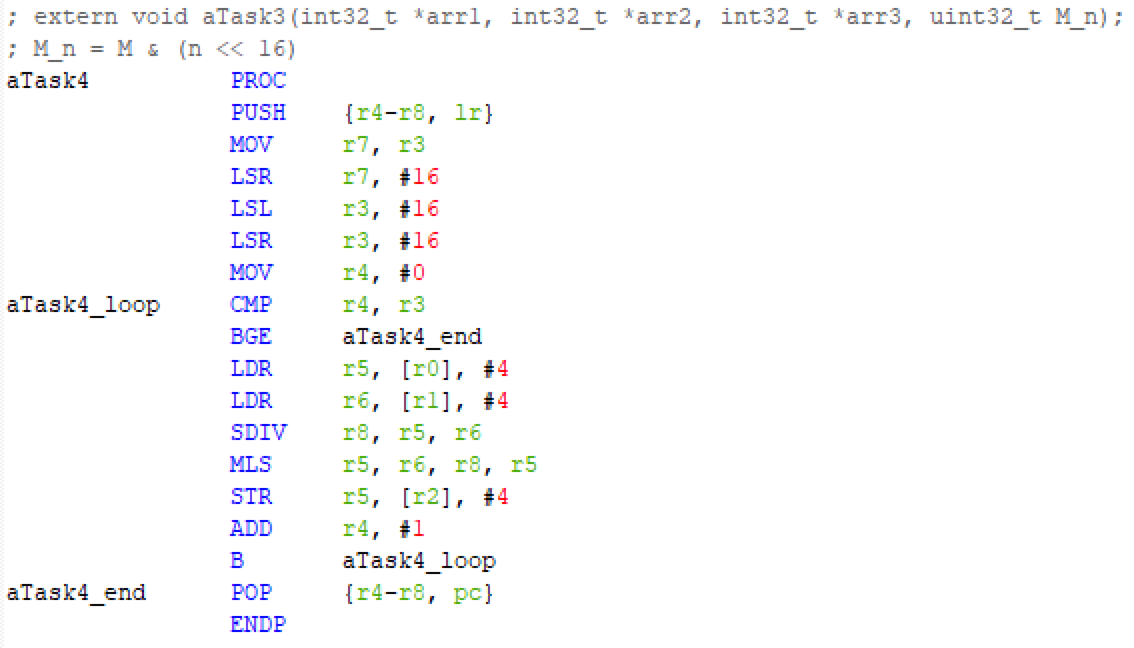
```
; extern void aTask1(int32_t *arr1, int32_t *arr2, int32_t *arr3, uint32_t M);
aTask1          PROC
                PUSH    {r4-r6, lr}
                MOV     r4, #0
aTask1_loop     CMP     r4, r3
                BGE     aTask1_end
                LDR     r5, [r0], #4
                LDR     r6, [r1], #4
                ADD     r5, r6
                STR     r5, [r2], #4
                ADD     r4, #1
                B       aTask1_loop
aTask1_end      POP     {r4-r6, pc}
                ENDP
```



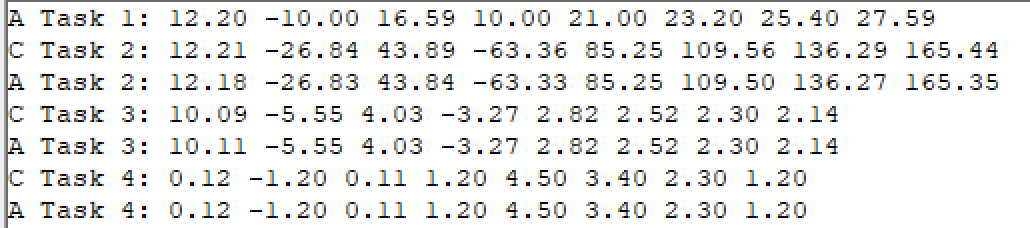
```
; extern void aTask2(int32_t *arr1, int32_t *arr2, int32_t *arr3, uint32_t M_n);
; M_n = M | (n << 16)
aTask2          PROC
                PUSH    {r4-r7, lr}
                MOV     r7, r3
                LSR     r7, #16
                LSL     r3, #16
                LSR     r3, #16
                MOV     r4, #0
aTask2_loop     CMP     r4, r3
                BGE     aTask2_end
                LDR     r5, [r0], #4
                LDR     r6, [r1], #4
                MUL     r5, r6
                ASR     r5, r7
                STR     r5, [r2], #4
                ADD     r4, #1
                B       aTask2_loop
aTask2_end      POP     {r4-r7, pc}
                ENDP
```



```
; extern void aTask3(int32_t *arr1, int32_t *arr2, int32_t *arr3, uint32_t M_n);
; M_n = M & (n << 16)
aTask3          PROC
                PUSH    {r4-r7, lr}
                MOV     r7, r3
                LSR     r7, #16
                LSL     r3, #16
                LSR     r3, #16
                MOV     r4, r3
aTask3_loop     LDR     r5, [r0], #4
                LDR     r6, [r1], #4
                LSL     r5, r7
                SDIV    r5, r6
                STR     r5, [r2], #4
                SUB     r4, #1
                CMP     r4, #0
                BGT     aTask3_loop
aTask3_end      POP     {r4-r7, pc}
                ENDP
```



```
; extern void aTask3(int32_t *arr1, int32_t *arr2, int32_t *arr3, uint32_t M_n);
; M_n = M & (n << 16)
aTask4          PROC
                PUSH    {r4-r8, lr}
                MOV     r7, r3
                LSR     r7, #16
                LSL     r3, #16
                LSR     r3, #16
                MOV     r4, #0
aTask4_loop     CMP     r4, r3
                BGE     aTask4_end
                LDR     r5, [r0], #4
                LDR     r6, [r1], #4
                SDIV    r8, r5, r6
                MLS     r5, r6, r8, r5
                STR     r5, [r2], #4
                ADD     r4, #1
                B       aTask4_loop
aTask4_end      POP     {r4-r8, pc}
                ENDP
```



```
A Task 1: 12.20 -10.00 16.59 10.00 21.00 23.20 25.40 27.59
C Task 2: 12.21 -26.84 43.89 -63.36 85.25 109.56 136.29 165.44
A Task 2: 12.18 -26.83 43.84 -63.33 85.25 109.50 136.27 165.35
C Task 3: 10.09 -5.55 4.03 -3.27 2.82 2.52 2.30 2.14
A Task 3: 10.11 -5.55 4.03 -3.27 2.82 2.52 2.30 2.14
C Task 4: 0.12 -1.20 0.11 1.20 4.50 3.40 2.30 1.20
A Task 4: 0.12 -1.20 0.11 1.20 4.50 3.40 2.30 1.20
```