0x08000460 B098 SUB sp,sp,#0x60

В начале Main указатель стека выделяет место под 24 int’овые переменные (10 и 13 в массивах и 1 для хранения медианы).

Затем создаётся первый массив из 10 элементов

0x08000462 2228 MOVS r2,#0x28

В r2 записывается 40, что является размером 10 элементов создаюшегося массива. Видимо, одним из аргументом встроенной функции создания массива является его размер.

0x08000464 490A LDR r1,[pc,#40] ; @0x08000490

0x08000466 A80E ADD r0,sp,#0x38

Записываем в регистр r0 значение регистра sp+0x38. Это адрес первого элемента массива.

0x08000468 F000F814 BL.W \_\_aeabi\_memcpy4 (0x08000494)080001A4) eof(buf)/4));

Приступаем к выполнению тела функции.

0x08000494 B510 PUSH {r4,lr}

Сохраняем контекст вызова.

Инициализируется массив.

0x080004DA 4770 BXEQ lr

Выходим из функции. Так как сразу начинается выполнения следующей команды (без возвращения), можем предположить, что функция инициализации массива имеет тип void.

Переходим к выполнению сортировки.

0x0800046C 210A MOVS r1,#0x0A

В регистр помещается вычисленное значение аргумента вызываемой функции.

0x0800046E A80E ADD r0,sp,#0x38

Записываем в регистр r0 значение регистра sp+0x38. Это адрес первого элемента массива.

0x08000470 F7FFFFDA BL.W sort (0x08000428)

Вызов функции сортировки-переход по адресу первой инструкции этой функции.

0x08000428 B530 PUSH {r4-r5,lr}

Закидываем в стек содержимое регистров r4,r5 и lr. В lr хранится адрес возврата из функции. При записи в стек адрес ожидаемо убывает на (кол-во регистров)\*4

0x0800045E BD30 POP {r4-r5,pc}

Возвращаем контекст вызова функции сортировки, одновременно перемещая указатель pc

Аналогичным образом создаётся второй массив.

Теперь поиск медианы.

0x08000480 210D MOVS r1,#0x0D

Заносим аргумент в регистр

0x08000482 A801 ADD r0,sp,#0x04

Это адрес переменной answer, куда будет записано вычисленное значение медианы данного массива.

Поиск медианы.

0x08000484 F7FFFF90 BL.W getMedian (0x080003A8)

Переходим к выполнению первой инструкции в функции getMedian

0x080003A8 B5F0 PUSH {r4-r7,lr}

Заносим в стек регистры. Это контекст вызова функции.

0x08000410 B917 CBNZ r7,0x08000418

0x08000412 F8520023 LDR r0,[r2,r3,LSL #2]

Проверяем условие в этой строчке на неравенство нулю и переходим (если не равно) к следующему элементу. Если же равенство нулю соблюдается, перехода не происходит и выполняется следующая строка ассемблерного кода.

if((equal-abs(sum))%2==0) return buf[i];

В r0 записывается элемент массива, для которого выполнились условия, описанные в спецификации (громкое слово для околокомментария на ломаном английском) перед функцией.

0x08000416 BDF0 POP {r4-r7,pc}

Возвращаем контекст.

0x08000488 9000 STR r0,[sp,#0x00]

Записываем по адресу переменной answer вычисленное функцией значение.