```
;----------------------------------------------------------------------
; quicksort.s
;
; You will need to use map.ini with this file.
;
; Implement all five functions in quicksort.c: main(), quicksort(),
; partition(), swap(), and binarysearch().  As a leaf function, swap()
; does not need an activation frame, if you’re careful.  The remaining
; functions all need activation frames.
;
; Use the start-up code below.  Note that it will call main() for you.
; Also note that it stores values into r4--r10.  Those values should
; still be there once main() returns and sp and fp should be at their
; initial values.  The clean-up code verifies this.
;
; In order to check the values of found, passed, and failed in main(),
; you’ll need to set a breakpoint just prior to main() popping its
; frame.
;
; Follow the ARM Procedure Call Standard:
;
;  1) Register r0 is used for return values.
;
;  2) Registers r0--r3 are used for parameter passing.  They may also
;     be used for scratch storage.
;
;  3) Registers r4--r10 are used for local variables, and must be
;     saved and restored if used.
;
;  4) Register r11 is the frame pointer, fp.
;
;  5) Register r12 may be used for scratch storage.
;
;  6) Register r13 is the stack pointer, sp.
;
;  7) Register r14 is the link register, lr.
;
;  8) Register r15 is the program counter.
;
;  9) Do NOT assume that values in the scratch registers (r0--r3 and
;     r12) are preserved across function calls.  For example,
;     parameters that need to be preserved across function calls
;     should be copied to local variable registers and the values in
;     the local variable registers should be the values used.
;----------------------------------------------------------------------


;----------------------------------------------------------------------
; Program area follows.
;----------------------------------------------------------------------

        area prog, code, readwrite
        entry

;----------------------------------------------------------------------
; Start-up code.
;----------------------------------------------------------------------

        mov r4, #-1
        mov r5, #-2
        mov r6, #-3
        mov r7, #-4
        mov r8, #-5
        mov r9, #-6
        mov r10, #-7
        ldr sp, =bos
        mov fp, sp

        bl main

        cmp r4, #-1   ; Clean-up code.
        bne whoops
        cmp r5, #-2
        bne whoops
        cmp r6, #-3
        bne whoops
        cmp r7, #-4
        bne whoops
        cmp r8, #-5
        bne whoops
        cmp r9, #-6
        bne whoops
        cmp r10, #-7
        bne whoops
        ldr r0, =bos
        cmp r0, sp
        bne whoops
        cmp r0, fp
        bne whoops

good    b good

whoops  b whoops


;----------------------------------------------------------------------
; main()
;----------------------------------------------------------------------


;----------------------------------------------------------------------
; quicksort()
;----------------------------------------------------------------------


;----------------------------------------------------------------------
; partition()
;----------------------------------------------------------------------


;----------------------------------------------------------------------
; swap()
;----------------------------------------------------------------------


;----------------------------------------------------------------------
; binarysearch()
;----------------------------------------------------------------------


;----------------------------------------------------------------------
; End of code.
;----------------------------------------------------------------------


;----------------------------------------------------------------------
```

```
; Data area follows.
;-----------------------------------------------------------------------

        area progdata, data, readwrite

        dcd 0XDABBAD00   ; 1 word "fence" used to find the beginning/end
                         ; of arrays in the debugger’s memory window.

data1   dcd 34           ; 10 element array
        dcd 14
        dcd 85
        dcd 18
        dcd 80
        dcd 81
        dcd 55
        dcd 10
        dcd 41
        dcd 56

        dcd 0XDABBAD00   ; 1 word "fence"

keys    dcd 85           ; 4 element array
        dcd 17
        dcd 41
        dcd 99

        dcd 0XDABBAD00   ; 1 word "fence"

data2   dcd 13202        ; 100 element array
        dcd 421339
        dcd 40723
        dcd 648473
        dcd 93130
        dcd 174743
        dcd 961310
        dcd 163681
        dcd 266528
        dcd 809332
        dcd 844292
        dcd 52042
        dcd 528863
        dcd 257713
        dcd 167976
        dcd 428474
        dcd 156392
        dcd 669761
        dcd 961107
        dcd 754483
        dcd 34199
        dcd 220947
        dcd 498637
        dcd 987780
        dcd 195297
        dcd 651326
        dcd 523288
        dcd 802797
        dcd 676022
        dcd 535849
        dcd 128108
        dcd 205576
        dcd 957188
        dcd 685184
        dcd 854049
        dcd 566671
        dcd 859927
        dcd 331711
        dcd 730352
        dcd 642807
        dcd 657395
        dcd 574644
        dcd 211201
        dcd 186258
        dcd 348709
        dcd 379177
        dcd 131085
        dcd 505101
        dcd 565291
        dcd 92192
        dcd 775936
        dcd 599490
        dcd 829491
        dcd 274573
        dcd 103622
        dcd 24789
        dcd 442251
        dcd 626911
        dcd 343938
        dcd 634625
        dcd 679112
        dcd 472046
        dcd 356553
        dcd 152652
        dcd 673582
        dcd 726954
        dcd 719323
        dcd 49861
        dcd 58666
        dcd 966027
        dcd 209020
        dcd 232413
        dcd 540671
        dcd 420222
        dcd 935024
        dcd 405733
        dcd 799399
        dcd 66109
        dcd 427186
        dcd 881042
        dcd 674653
        dcd 719475
        dcd 996884
        dcd 20496
        dcd 510400
        dcd 100507
        dcd 561637
        dcd 952652
        dcd 727418
        dcd 905575
        dcd 103629
        dcd 406530
        dcd 893974
        dcd 460183
        dcd 75534
        dcd 83908
        dcd 187137
        dcd 794858
        dcd 133770
        dcd 13203

        dcd 0XDABBAD00   ; 1 word "fence"

        space 4096
bos     dcd 0XDABBAD00   ; 1K word stack

        end
```