Debugging with printf

you can call printf from an ARM assembly language program

some of the details will be explained later, but for now use this skeleton - note that you have to use the registers r0 and r1 for the call to printf

```asm
main: push {lr}
    <your program>
    ldr  r0, =fmt
    mov  r1, <register holding the value to be printed>
    bl   printf
    <rest of your program>
    mov r0, #0
    pop {pc}
    .section ".rodata"
fmt:  .asciz "%d\n"
```
Debugging with printf

you can change the format string in "fmt" to include a label, use a different format specifier, etc. - if you want to print additional values, you will need to move those values into r2, r3, etc.

you can also add additional format strings
Debugging with printf

main: push {lr}
    <your program>
    ldr  r0, =fmt1
    mov  r1, <register holding first value to be printed>
    mov  r2, <register holding second value to be printed>
    bl   printf
    ldr  r0, =fmt2
    mov  r1, <register holding first value to be printed>
    mov  r2, <register holding second value to be printed>
    bl   printf
    <rest of your program>
    mov  r0, #0
    pop  {pc}
    .section  ".rodata"

fmt1: .asciz "values in decimal are %d and %d\n"
fmt2: .asciz "values in hex are 0x%08x and 0x%08x\n"