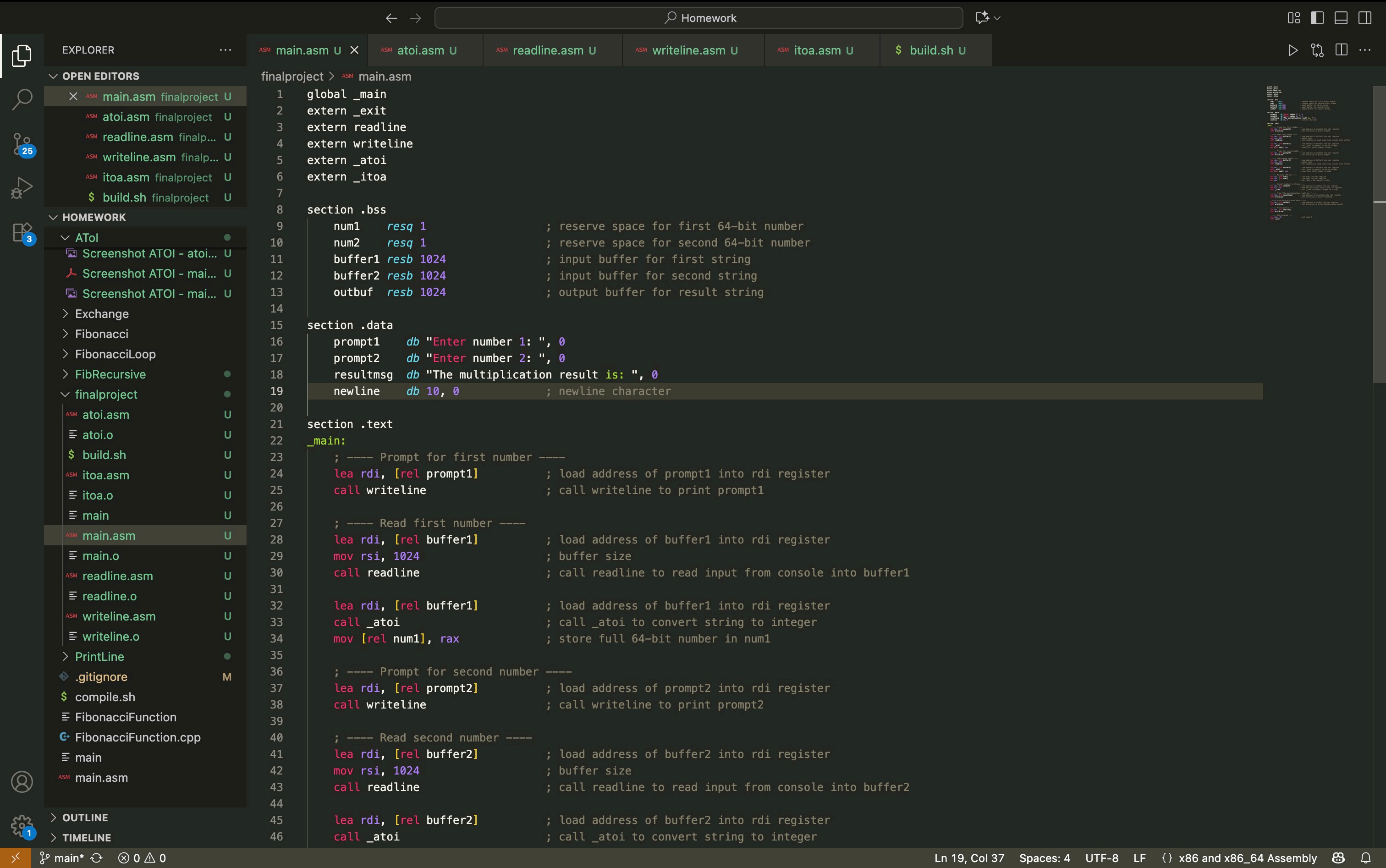

```
global _main
extern _exit
extern readline
extern writeline
extern _atoi
extern _itoa

section .bss
    num1    resq 1                  ; reserve space for first 64-bit number
    num2    resq 1                  ; reserve space for second 64-bit number
    buffer1 resb 1024               ; input buffer for first string
    buffer2 resb 1024               ; input buffer for second string
    outbuf  resb 1024               ; output buffer for result string

section .data
    prompt1    db "Enter number 1: ", 0
    prompt2    db "Enter number 2: ", 0
    resultmsg  db "The multiplication result is: ", 0
    newline    db 10, 0             ; newline character

section .text
_main:
    ; ---- Prompt for first number ----
    lea rdi, [rel prompt1]          ; load address of prompt1 into rdi register
    call writeline                  ; call writeline to print prompt1

    ; ---- Read first number ----
    lea rdi, [rel buffer1]          ; load address of buffer1 into rdi register
    mov rsi, 1024                   ; buffer size
    call readline                   ; call readline to read input from console into buffer1

    lea rdi, [rel buffer1]          ; load address of buffer1 into rdi register
    call _atoi                      ; call _atoi to convert string to integer
    mov [rel num1], rax             ; store full 64-bit number in num1

    ; ---- Prompt for second number ----
    lea rdi, [rel prompt2]          ; load address of prompt2 into rdi register
    call writeline                  ; call writeline to print prompt2

    ; ---- Read second number ----
    lea rdi, [rel buffer2]          ; load address of buffer2 into rdi register
    mov rsi, 1024                   ; buffer size
    call readline                   ; call readline to read input from console into buffer2

    lea rdi, [rel buffer2]          ; load address of buffer2 into rdi register
    call _atoi                      ; call _atoi to convert string to integer
```