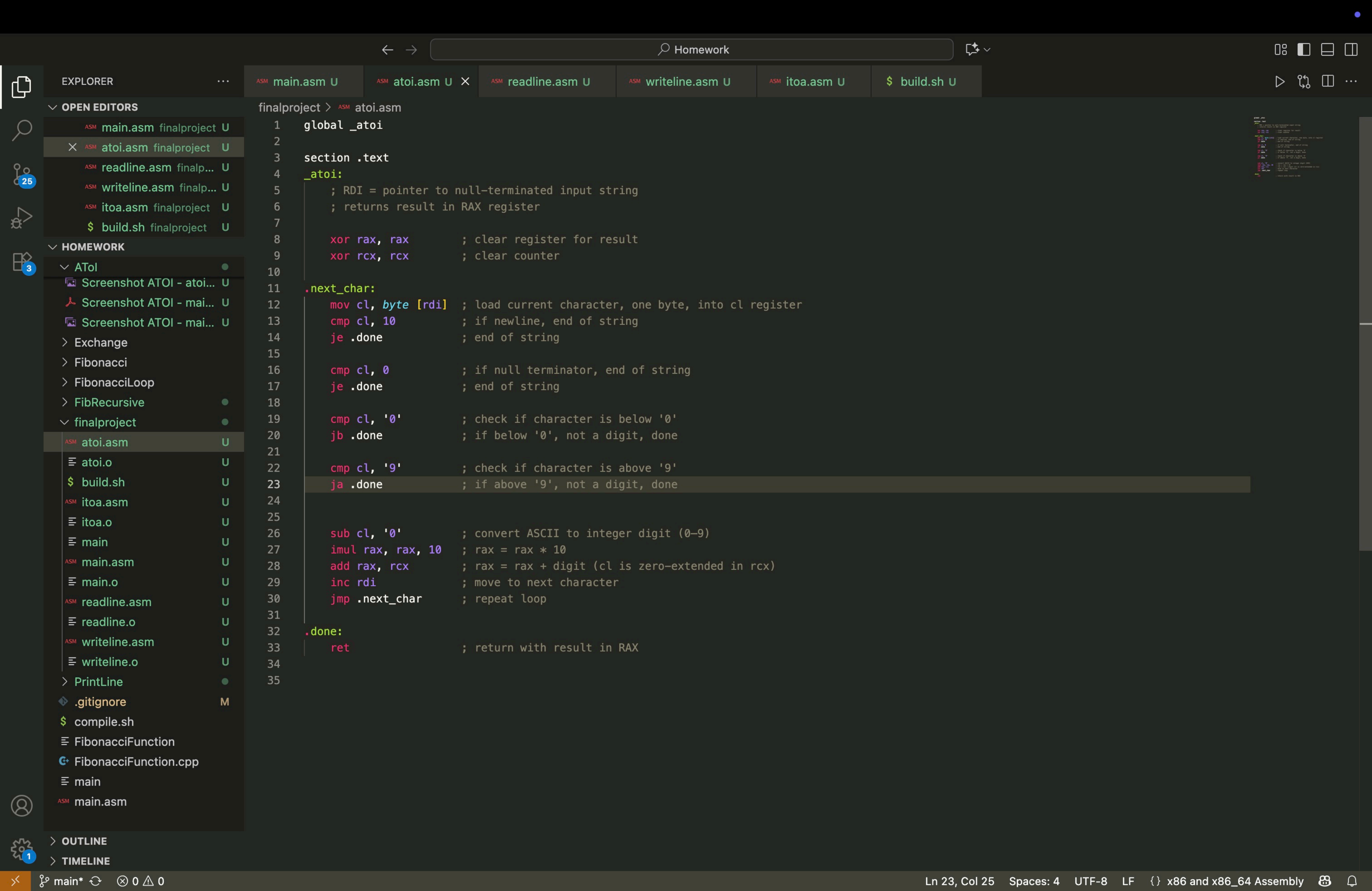

```
global _atoi

section .text
_atoi:
    ; RDI = pointer to null-terminated input string
    ; returns result in RAX register

    xor rax, rax       ; clear register for result
    xor rcx, rcx       ; clear counter

.next_char:
    mov cl, byte [rdi]  ; load current character, one byte, into cl register
    cmp cl, 10         ; if newline, end of string
    je .done           ; end of string

    cmp cl, 0          ; if null terminator, end of string
    je .done           ; end of string

    cmp cl, '0'        ; check if character is below '0'
    jb .done           ; if below '0', not a digit, done

    cmp cl, '9'        ; check if character is above '9'
    ja .done           ; if above '9', not a digit, done


    sub cl, '0'        ; convert ASCII to integer digit (0-9)
    imul rax, rax, 10  ; rax = rax * 10
    add rax, rcx       ; rax = rax + digit (cl is zero-extended in rcx)
    inc rdi            ; move to next character
    jmp .next_char     ; repeat loop

.done:
    ret                ; return with result in RAX
```