EXPLORER

OPEN EDITORS
main.asm finalproject U
atoi.asm finalproject U
readline.asm finalp... U
writeline.asm finalp... U
itoa.asm finalproject U
build.sh finalproject U

HOMEWORK
AToI
Screenshot ATOI - atoi... U
Screenshot ATOI - mai... U
Screenshot ATOI - mai... U
Exchange
Fibonacci
FibonacciLoop
FibRecursive
finalproject
atoi.asm U
atoi.o U
build.sh U
itoa.asm U
itoa.o U
main U
main.asm U
main.o U
readline.asm U
readline.o U
writeline.asm U
writeline.o U
PrintLine
.gitignore M
compile.sh
FibonacciFunction
FibonacciFunction.cpp
main
main.asm

OUTLINE
TIMELINE


```
22  _main:
31
32      lea rdi, [rel buffer1]          ; load address of buffer1 into rdi register
33      call _atoi                      ; call _atoi to convert string to integer
34      mov [rel num1], rax             ; store full 64-bit number in num1
35
36      ; ---- Prompt for second number ----
37      lea rdi, [rel prompt2]          ; load address of prompt2 into rdi register
38      call writeline                  ; call writeline to print prompt2
39
40      ; ---- Read second number ----
41      lea rdi, [rel buffer2]          ; load address of buffer2 into rdi register
42      mov rsi, 1024                   ; buffer size
43      call readline                   ; call readline to read input from console into buffer2
44
45      lea rdi, [rel buffer2]          ; load address of buffer2 into rdi register
46      call _atoi                      ; call _atoi to convert string to integer
47      mov [rel num2], rax             ; store full 64-bit number in num2
48
49      ; ---- Multiply (64-bit) ----
50      mov rax, [rel num1]             ; load num1 into RAX register
51      mov rbx, [rel num2]             ; load num2 into RBX register
52      mul rbx                         ; RAX = RAX * RBX (result in RAX)
53
54      ; ---- Convert product to string ----
55      lea rdi, [rel outbuf]           ; load address of outbuf into rdi register
56      mov rsi, rax                    ; move multiplication result into rsi register
57      call _itoa                      ; call _itoa to convert integer to string
58
59      ; ---- Print "The multiplication result is: " ----
60      lea rdi, [rel resultmsg]        ; load address of resultmsg into rdi register
61      call writeline                  ; call writeline to print resultmsg
62
63      ; ---- Print multiplication result ----
64      lea rdi, [rel outbuf]           ; load address of outbuf into rdi register
65      call writeline                  ; call writeline to print multiplication result
66
67      ; ---- Print newline ----
68      lea rdi, [rel newline]
69      call writeline
70
71      ; ---- Exit program ----
72      xor rdi, rdi                    ; exit code 0
73      call _exit
```