

ADDITION IN PAIL WITH USER INPUT

CODE:

```
global _start
```

```
section .data
```

```
msg1 db "Enter 1st Value:", 10
```

```
msg1len equ $ - msg1
```

```
msg2 db "Enter 2nd Value:", 10
```

```
msg2len equ $ - msg2
```

```
msg3 db "The addition is: "
```

```
msg3len equ $ - msg3
```

```
newline db 10
```

```
section .bss
```

```
val1 resb 2 ; will store input + newline
```

```
val2 resb 2
```

```
result resb 2
```

```
section .text
```

```
_start:
```

```
; Prompt for first value
```

```
mov eax, 4
```

```
mov ebx, 1
mov ecx, msg1
mov edx, msg1len
int 0x80
```

```
; Read first value (2 bytes: digit + newline)
```

```
mov eax, 3
mov ebx, 0
mov ecx, val1
mov edx, 2
int 0x80
```

```
; Prompt for second value
```

```
mov eax, 4
mov ebx, 1
mov ecx, msg2
mov edx, msg2len
int 0x80
```

```
; Read second value (2 bytes: digit + newline)
```

```
mov eax, 3
mov ebx, 0
mov ecx, val2
mov edx, 2
int 0x80
```

; Convert val1[0] and val2[0] from ASCII to integer

mov al, [val1]

sub al, '0' ; AL = val1 numeric

mov bl, [val2]

sub bl, '0' ; BL = val2 numeric

; Add values

add al, bl ; AL = val1 + val2

; Convert result to ASCII (max 18)

mov ah, 0

mov bl, 10

div bl ; AL = quotient (tens), AH = remainder (units)

add al, '0'

mov [result], al

add ah, '0'

mov [result+1], ah

; Print result message

mov eax, 4

mov ebx, 1

mov ecx, msg3

```
mov edx, msg3len
```

```
int 0x80
```

```
; Print result digits
```

```
mov eax, 4
```

```
mov ebx, 1
```

```
mov ecx, result
```

```
mov edx, 2
```

```
int 0x80
```

```
; Print newline
```

```
mov eax, 4
```

```
mov ebx, 1
```

```
mov ecx, newline
```

```
mov edx, 1
```

```
int 0x80
```

```
; Exit
```

```
mov eax, 1
```

```
xor ebx, ebx
```

```
int 0x80
```