

1. Um código com uma **array** com n números e uma variável **var** na *section data*. Logo, compare o valor de **var** com o valor da **array**, enquanto for diferente, continue o código, se for igual, imprima: "Valor encontrado no vetor" .

EXEMPLO BÁSICO

```
1 section .data
2     array db 3,4,5,6,7,8
3
4 section .text
5     global _start
6
7 _start:
8     mov rbx, [array+0]
9
10
11     mov rax, 60
12     mov rdi, rbx
13     syscall
```

Saída do programa

[Execution complete with exit code 3]

EXEMPLO BÁSICO

```
1 section .data
2     array db 3,4,5,6,7,8
3
4 section .text
5     global _start
6
7 _start:
8     mov rbx, [array+1]
9
10
11     mov rax, 60
12     mov rdi, rbx
13     syscall
```

Saída do programa

[Execution complete with exit code 4]

EXEMPLO BÁSICO

```
1 section .data
2     array db 3,4,5,6,7,8
3
4 section .text
5     global _start
6
7 _start:
8     mov rbx, [array+6]
9
10
11     mov rax, 60
12     mov rdi, rbx
13     syscall
```


Saída do programa

[Execution complete with exit code 0]

1. Um código com uma **array** com n números e uma variável **var** na *section data*. Logo, compare o valor de **var** com o valor da **array**, enquanto for diferente, continue o código, se for igual, imprima: "Valor encontrado no vetor" .

```
1 section .data
2
3 array db 6,3,4,5,0
4 var    db 3
5 msg    db "Valor encontrado no vetor."
6 lenmsg equ $-msg
7
8
```

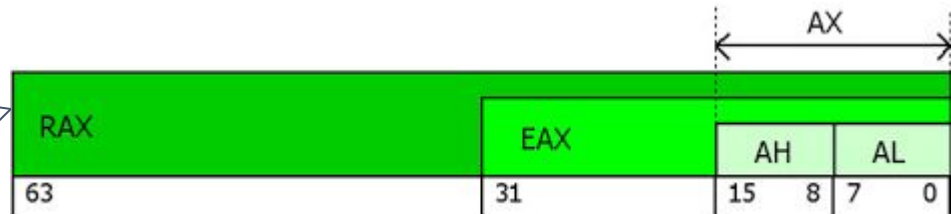
Define o fim do seu array



```

12 ▾ findval:
13     mov rax, 0
14     mov bl, [var]
15
16 ▾ .loop:
17     cmp byte [array+rax], 0
18     je end
19
20     cmp byte [array+rax], bl
21     je find
22
23     inc rax
24     jmp .loop
25
26 ▾ find:
27     mov rax, 1
28     mov rdi, 1
29     mov rsi, msg
30     mov rdx, lenmsg
31     syscall
32     ret
33
34
35 ▾ end:
36     ret

```



```
12 ▾ findval:
13     mov rax, 0
14     mov bl, [var]
15
16 ▾     .loop:
17         cmp byte [array+rax], 0
18         je end
19
20         cmp byte [array+rax], bl
21         je find
22
23         inc rax
24         jmp .loop
25
26 ▾     find:
27         mov rax, 1
28         mov rdi, 1
29         mov rsi, msg
30         mov rdx, lenmsg
31         syscall
32         ret
33
34
35 ▾     end:
36         ret
```



```

12 ▾ findval:
13     mov rax, 0
14     mov bl, [var]
15
16 ▾     .loop:
17         cmp byte [array+rax], 0
18         je end
19
20         cmp byte [array+rax], bl
21         je find
22
23         inc rax
24         jmp .loop
25
26 ▾     find:
27         mov rax, 1
28         mov rdi, 1
29         mov rsi, msg
30         mov rdx, lenmsg
31         syscall
32         ret
33
34
35 ▾     end:
36         ret

```

```

global _start
▾ _start:

    call findval

    mov rdi, 0
    mov rax, 60
    syscall

```

https://www.mycopiler.io/pt/new/asm-x86_64?fork=Kn6PYf3VBmr

section.data

array db 6,3,4,5,0

var db

msg db "Valor encontrado no vetor"

lenmsg equ \$-msg

section.text

findval:

mov rax, 0

mov bl, [var]

.loop:

cmp byte [array+rax], 0

je end

cmp byte [array+rax], bl

je find

inc rax

jmp .loop

find:

mov rax, 1

mov rdi, 1

mov rsi, msg

mov rdx, lenmsg

syscall

ret

end:

ret