

Link explicação - <https://youtu.be/sGSCsbRPHJM>

JMP start

hello: DB "teste" ; Variable

DB 0 ; String terminator

start:

MOV C, hello ; Point to var

MOV A, C ;Movendo os 3 registradores pro inicio da entrada

MOV B, C

MOV D, C

CALL finalVet ;botando meu registrador na ultima posição

bubble:

INC B ;incremento B pra comparar com o A

MOV D, B ; vendo se não é a ultima posição

CMP D, C ; Aqui eu to verificando se é a ultima posição

JE outroFor ;Caso seja o ultimo, ele vai pra função de iteração

PUSH A

PUSH B

MOV A, [A] ;movendo pro valor de operação

MOV B, [B]

CMP A, B ;comparando se os valores de dentro do registrador

POP B

POP A

JA troca ;caso seja maior, ele vai pra função troca

INC A ; vai pro proximo teste

JMP bubble

outroFor:

DEC C ;decrementa o C, já que ao maior valor já ta no final

CMP C, 2 ;compara pra ver se é o ultimo elemento

```
JE print ; se for, ele vai printar na saida  
MOV A, 2 ; coloca o A e o B na primeira posição  
MOV B, 2  
JMP bubble ; volta pro bubble
```

troca:

```
;escrevendo o A no Aux  
MOV D, 0 ;move o D pra um endereço auxiliar  
PUSH A ; coloca o A na pilha  
MOV A, [A] ; move o A pro valor de A  
MOV [D], A ; escreve o A no endereço do D  
POP A ;volta a posição do A inicial  
;escrevendo o B no A  
PUSH B  
MOV B, [B]  
MOV [A], B  
POP B  
;escrevendo o Aux no B  
PUSH D  
MOV D, [D]  
MOV [B], D  
POP D  
INC A  
JMP bubble
```

finalVet:

```
INC C ;Vai percorrendo até achar o fim  
MOV D, C  
MOV D, [D]  
CMP D, 0 ; Verifica se é o fim
```

```

JNE finalVet ; volta pro laço se nao for o fim

CALL bubble ; se for o fim ele vai pro bubble

print:                ; print(C:*from, D:*to)

MOV C, 2      ;move pra posição inicial

MOV D, 232    ;E o D pra posição de escrever na memoria

PUSH A

PUSH B

MOV B, 0

.loop:

MOV A, [C]    ; pega o char da variavel

MOV [D], A    ; escreve na saida

INC C        ; incrementa a saida e a variavel que percorre

INC D

CMP B, [C]    ; checa se é fim

JNZ .loop    ; volta se nao for


POP B

POP A

HLT

```