



Experimento 2. Ordenar Vetor

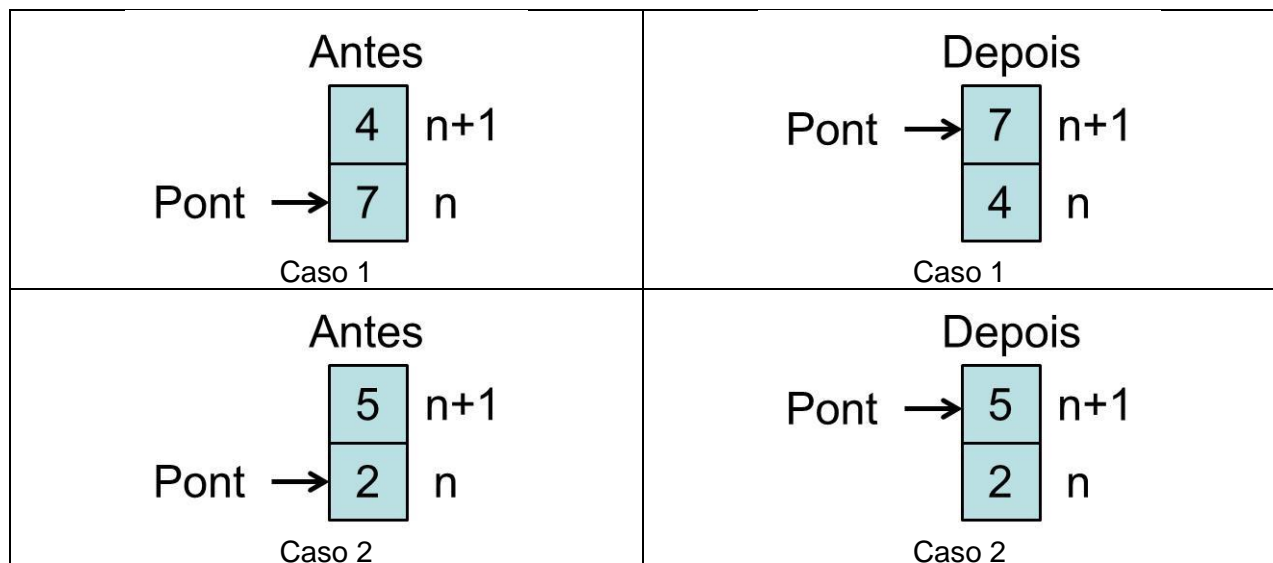
OBJETIVO:

Escrever subrotina que ordena, de forma crescente, um vetor.

DADOS:

Um método muito conhecido para ordenar os elementos de um vetor é o método da BOLHA. Este método pede uma subrotina auxiliar que vamos chamar de ORD.

A subrotina ORD ordena de forma crescente duas posições de memória: a apontada pelo ponteiro e a seguinte. Veja os exemplos abaixo. No Caso 1, as duas posições foram trocadas. No Caso 2, a ordem já estava correta e nada foi feito. Note que em ambos os casos, o ponteiro terminou apontando para a segunda posição.



Vamos agora considerar um vetor com n elementos, como mostrado abaixo. Note que o vetor está na vertical. Se varrermos todo este vetor com a subrotina ORD, no topo do vetor deverá estar o maior elemento (Maior 1). Se varrermos novamente o vetor, exceto a última posição, selecionaremos o segundo maior elemento (MAIOR 2). Repetimos esse procedimento $n-1$ vezes e o vetor será ordenado.

n	y	Maior 1	Maior 1	Maior 1	...	Maior 1
n-1	x	y	Maior 2	Maior 2	...	Maior 2
...	Maior 3	...	Maior 3
3	c	c	c
2	b	b	b	b	...	Menor 2
1	a	a	a	a	...	Menor 1
Seq.	Original	Varrida 1	Varrida 2	Varrida 3	...	Varrida n-1



Exemplo: vetor [4, 7, 3, 5, 1] com 5 elementos, na horizontal.

	Tamanho	Elementos do vetor				
Original	5	4	7	3	5	1
Varrida 1 (4 comparações)	5	4	3	5	1	7
Varrida 2 (3 comparações)	5	3	4	1	5	7
Varrida 3 (2 comparações)	5	3	1	4	5	7
Varrida 4 (1 comparação)	5	1	3	4	5	7

PEDIDOS:

Programa 6:

Escreva subrotina **ORDENA** que recebe em R5 o endereço de início de um vetor de bytes (sem sinal) e o ordena. Organize os registradores da forma abaixo:

- R5 → apontador para memória
- R6 → contador principal
- R7 → contador secundário
- R8 → guarda endereço de início do vetor
- R9 e R10 → rascunho para a subrotina ORD

Para este programa, declare um vetor de bytes formado pela concatenação do código ASCII dos nomes completos de cada membro da equipe. Note que o montador já converte as letras para o código ASCII correspondente, como mostrado abaixo. Use letras maiúsculas, omita os espaços e não use acentos. Preste atenção ao tipo das aspas. Note que usamos a mesma formatação de vetor do Roteiro 1, Programa 1.

Atenção: Não use a instrução swpb (swap bytes) pois ela opera em palavras de 16-bits e está sempre alinhada em endereços pares, ou seja, não vai funcionar em endereços ímpares.

```
.data
; Declarar vetor com 11 elementos [JOAQUIMJOSE]
vetor:      .byte 11, 'J','O','A','Q','U','I','M','J','O','S','E'
```

O programa deve ter a seguinte organização:

```
;-----
; Main loop here
;-----
;
    mov    #vetor,R5    ;inicializar R5 com o endereço do vetor
    call   #ORDENA      ;chamar subrotina
    jmp    $            ;travar execução
    nop                ;exigido pelo montador
;
ORDENA:    ...
            ...
            ret
```



Programa 7:

Escreva subrotina **ORDENAS** que recebe em R5 o endereço de início de um vetor com palavras de 16 bits (W16) com sinal e o ordena. Tome como partida o programa anterior. Serão necessárias apenas algumas alterações.

Para formar o vetor vamos usar o número de matrícula e o ano de nascimento de cada membro da equipe. Veja o exemplo para uma equipe com 2 alunos:

Aluno 1: matrícula = 12/1234567 e nasceu em 1990 → 121, 234, 567, -1990.

Aluno 2: matrícula = 11/786745 e nasceu em 1980 (está velho) → 117, 867, 45, -1980

```
.data  
; Declarar vetor com 8 elementos [121, 234, 567, -1990, 117, 867, 45, -1980]  
vetor:      .word      8, 121, 234, 567, -1990, 117, 867, 45, -1980
```

SUGESTÕES:

- Esboce o fluxograma para a subrotina ORD.
- Teste a subrotina ORD
- Documente as sub-rotinas, é provável que você as use em experimentos futuros.
- Inicie os testes com vetores curtos, por exemplo, de tamanho 3.

RELATÓRIO

O relatório é individual, e deve ser entregue impresso (ou feito à mão). Em hipótese alguma será admitida a entrega do relatório de forma eletrônica.

Questão 1 (4 pontos)

Apresente o Programa 6, comentando as partes mais importantes.

Questão 2 (1 ponto)

Apresente o Programa 7, comentando as partes mais importantes.

Questão 3 (1 ponto)

Em ambos os programas, logo abaixo de cada instrução emulada, usando uma linha de comentários, escreva a instrução que a substitui.

Questão 4 (1 ponto)

Caso o vetor fosse composto por palavras de 8 bits com sinal, onde a rotina ORD do Programa 6 precisaria ser alterada?

Questão 5 (1 ponto)

Caso o vetor fosse composto por palavras de 16 bits (W16) sem sinal, onde a subrotina ORD do Programa 7 seria alterada.



TABELA ASCII PADRÃO

	0_	1_	2_	3_	4_	5_	6_	7_
_0	NUL	DLE	SP	0	@	P	`	p
_1	SOH	DC1	!	1	A	Q	a	q
_2	STX	DC2	"	2	B	R	b	r
_3	ETX	DC3	#	3	C	S	c	s
_4	EOT	DC4	\$	4	D	T	d	t
_5	ENQ	NAK	%	5	E	U	e	u
_6	ACK	SYN	&	6	F	V	f	v
_7	BEL	ETB	'	7	G	W	g	w
_8	BS	CAN	(8	H	X	h	x
_9	HT	EM)	9	I	Y	i	y
_A	LF	SUB	*	:	J	Z	j	z
_B	VT	ESC	+	;	K	[k	{
_C	FF	FS	,	<	L	\	l	
_D	CR	GS	-	=	M]	m	}
_E	SO	RS	.	>	N	^	n	~
_F	SI	UP	/	?	O	_	o	DEL

Exemplos: símbolo 'A' = código ASCII 41h
 símbolo 'B' = código ASCII 42h
 símbolo 'a' = código ASCII 61h
 símbolo 'z' = código ASCII 7Ah

Hexa	ASCII	Significado
00	NUL	NULL
01	SOH	Start Of Heading
02	STX	Start of TeXt
03	ETX	End of TeXt
04	EOT	End Of Transmission
05	ENQ	ENquire
06	ACK	ACKnowledge
07	BEL	BELL
08	BS	Back Space
09	HT	Horizontal Tab
0A	LF	Line Feed
0B	VT	Vertical Tab
0C	FF	Form Feed
0D	CR	Carriage Return
0E	SO	Shift Out
0F	SI	Shift In
7F	DEL	DELete"

Hexa	ASCII	Significado
10	DLE	Data Link Escape
11	DC1	Device Control 1
12	DC2	Device Control 2
13	DC3	Device Control 3
14	DC4	Device Control 4
15	NAK	Negative AcKnowledge
16	SYN	SYNchronism idle
17	ETB	End of Transmission Block
18	CAN	CANcel
19	EM	End of Medium
1A	SUB	SUBstitute
1B	ESC	ESCape
1C	FS	File Separator
1D	GS	Group Separator
1E	RS	Record Separator
1F	UP	Unit Separator
20	SP	SPace