

**Ex. 0** Considerando os vetores definidos abaixo, execute os comandos.

	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29		
A	[	0	1	2	3	4	4	3	22	21	6	7	8	8	8	9	2	13	4	5	6	3	8	2	3	1	12	14	5	6	7	]
B	[	10	9	5	6	5	6	3	22	21	6	7	18	28	38	9	2	31	4	5	6	3	8	2	3	1	12	14	25	26	27	]
C	[	0	1	-2	3	-4	5	3	2	1	6	7	8	-8	8	-9	2	31	-4	-5	-6	-3	-8	-2	-3	1	21	41	5	6	7	]

```

x = A[3] + B[3] - C[4];
y = C[x];
z = A[ B[ C[ C[y] ] ] ];
w = C[ A[7] + C[14] - C[27] ];

i = 0;
while ( i < 10 )
{
    C[i] = B[ A[i] ];
    i = i + 1;
}

```

**Ex. 1** Escreva um programa que lê o seu nome completo do teclado, uma letra por vez, armazena as letras num vetor (*com quantos elementos e de qual tipo?*) e imprime na seu nome de trás para a frente.

**Ex. 2** Uma sequência é dita *monotônica crescente* se, para qualquer par de elementos da sequência,  $e_{i+1} \geq e_i$ . Posto de outra forma, a partir do primeiro elemento, os elementos da sequência crescem, ou permanecem iguais aos anteriores, mas nunca decrescem. Para uma sequência *monotônica decrescente* temos que  $\forall i \bullet e_{i+1} \leq e_i$ . Uma sequência não monotônica é dita *aleatória*.

Escreva um programa que lê do teclado uma sequência de números naturais e os armazena num vetor (*de qual tamanho?*). A sequência é terminada por -1. Ao final da sequência seu programa indica que a sequência é monotônica crescente (imprime na tela “Crescente”), monotônica decrescente (imprime na tela “Decrescente”), ou se a sequência não é monotônica (imprime na tela “Aleatória”).

Escreva duas funções, a função `mais()` recebe dois números naturais como argumentos e retorna TRUE se o primeiro argumento é maior ou igual ao segundo, e retorna FALSE do contrário. A outra função, `menos()` recebe dois números naturais como argumentos e retorna TRUE se o primeiro argumento é menor ou igual ao segundo, e retorna FALSE do contrário.

O programa principal lê a sequência de números, chama as duas funções a cada novo elemento da sequência, e ao final, escreve na tela qual o tipo da sequência. Sequências vazias e unitárias são aleatórias. *Pista:* quais operadores lógicos são associativos?

Exemplo de execução:

Digite a sequência:

2 3 5 5 5 6 7 -1

Crescente

Outro exemplo de execução:

Digite a sequência:

2 3 5 4 5 6 7 -1

Aleatória

**Vire a folha.**

**Ex. 3** Uma *redução* é uma operação que “reduz” um conjunto de números para um único valor. Somatórios, produtórios e médias são exemplos de reduções.

Escreva um programa que efetua reduções por soma (somatórios) ou por produto (produtórios) de conjuntos de números. Seu programa principal lê do teclado um conjunto de números naturais. O conjunto é terminado por  $-1$  e deve ser armazenado num vetor (*de qual tamanho?*).

Escreva duas funções, a função `soma()` recebe o valor acumulado da soma até o momento e mais um valor e acrescenta este valor ao acumulado; a função `prod()` recebe o produto parcial e multiplica o novo valor pelo produto parcial. Ao final do conjunto, seu programa principal imprime as duas reduções do conjunto de números.