

Programação de Computadores

Prof. Dr. Josenalde Barbosa de Oliveira

josenalde.oliveira@ufrn.br

Variáveis indexadas (vetores, arrays unidimensionais)

- **Conjuntos homogêneos**, ou seja, com o mesmo tipo de dado
- Ao invés da declaração de uma única variável, tem-se a declaração de um conjunto de variáveis do mesmo tipo, que pode ter cada valor individual manipulado por meio de um **índice** (idx). Em Javascript é atribuído ao tipo OBJECT

```
let x, y;  
x = 20; // atribuição  
y = x;  
// acesso ao valor de x
```

```
let x = new Array(5), y; // let x = []  
x[2] = 20;  
/* atribuição do valor 20 ao índice 2  
do vetor x */  
y = x[2]; // acesso ao índice 2 de x
```

- Em Javascript, o primeiro índice é 0, logo o vetor x acima tem 5 elementos, com índices de 0 a 4. Ou seja, se o tamanho do vetor é N, os índices vão de 0 a N-1.

índices:

x:	0	1	2	3	4
	undefined	undefined	20	undefined	undefined

- Um vetor pode ser declarado e inicializado com uma lista de valores

```
var x = new Array(1,4,7,10,15); // var x = [1,4,7,10,15]
```

índices:

x:	0	1	2	3	4
Valores:	1	4	7	10	15

- Em Javascript é possível atribuir variável ao tamanho do vetor e acessar os índices OU criar um vetor vazio e adicionar os itens dinamicamente (seja com índice ou com o método PUSH, que adiciona um item após o outro, como que numa PILHA, com o índice 0 sendo a base da PILHA

```
let N,i;
N = parseInt(prompt());
let v = new Array(N);
for (let i=0; i<N; i++) {
    v[i] = parseInt(prompt('v[' + i + ']:')); //ou
    //v.push(parseInt(prompt('v[' + i + ']:'))); ou scanf(%d,v[i])
}
console.log(v.length); //tamanho do vetor PROPRIEDADE CHAVE PARA ALGORITMOS
```

No caso do push, irá gerar o dobro de elementos, sendo os três primeiros vazios, pois foram criados na inicialização

► Exemplo: OBI 2007 peça Perdida

► <https://olimpiada.ic.unicamp.br/pratique/p1/2007/f1/perdida/>: Escreva um programa que, dado um inteiro N e $N - 1$ inteiros numerados de 1 a N , descubra qual inteiro está faltando. $2 \leq N \leq 1000$

► https://olimpiada.ic.unicamp.br/static/extras/obi2007/gabaritos/2007f1p1_perdida

Exemplos:

3	2	4	1
3 1		2 4 3	

5	4
1 2 3 5	

https://github.com/josenalde/computer_programming/blob/main/src/perdida.js

<https://olimpiada.ic.unicamp.br/saci/cursos/provaf2/2021/>

Função **sort** por padrão compara caracteres e não números!

É preciso passar como parâmetro função de comparação (a,b)

Se retorno de **a-b** é negativo, ordena crescente

Para ordenar decrescente fazer retorno de **b-a**

► Exemplo: ler N números, salvar num vetor, e informar o maior número:

```
maior = v[0];  
for (let i=1; i<N; i++)  
    if (v[i]>maior)  
        maior = v[i];  
console.log(maior);
```

Algoritmo básico percorrendo o vetor!
A solução abaixo usa o Math.max.apply direto

```
function getRandom(min, max) {  
    return Math.floor(Math.random() * (max - min + 1)) + min;  
}
```

```
let a = [];  
for (let i = 0 ; i < 10 ; i++) {  
    a.push(getRandom(-9,9));  
}  
console.log(a);  
maior = Math.max.apply(null, a)  
console.log(maior)
```

https://github.com/josenalde/computer_programming/blob/main/src/max_min.js

Um número pseudo-aleatório entre 0 (inclusivo) e 1 (exclusivo).

A função `Math.floor(x)` retorna o menor número inteiro dentre o número "x"

TOPO

➡ Além do método `push()`, temos os métodos `pop()`, `shift()` e `unshift` para manipular



Se `v = [10, 20, 30, 40]`, com índices 0, 1, 2, 3:

`v.pop()`: remove elemento de maior índice (topo da pilha)

`v = [10, 20, 30]`

`v.unshift(50)`: inclui no início da pilha, deslocando os demais “para cima”

`v = [50, 10, 20, 30]`

`v.push(60)`: inclui no topo da pilha

`v = [50, 10, 20, 30, 60]`

`v.shift()`: remove do início da pilha

`v=[10,20,30,60]`

3	40
2	30
1	20
0	10

Pode-se inserir em determinado índice com o comando `splice`:

`v.splice(2,0,70)`; //a partir do `idx=2` remove 0 itens e inclui o 70

`v=[10,20,70,30,60]`;

► Fatiar arrays com método slice

Se $v = [10, 20, 30, 40]$, com índices 0, 1, 2, 3:

```
x = v.slice(1,3); //extraí do segundo até o terceiro elemento (idx=2)
```

INDICE

POSIÇÃO = idx+1

```
x = [20,30]
```

► Juntar arrays com método concat

Se $v1 = [10, 20]$ e $v2 = [50,60,70]$

```
v1v2 = v1.concat(v2); //não altera arrays originais
```

```
v1v2 = [10,20,50,60,70];
```

```
var num1 = [1, 2, 3];
```

```
var num2 = [4, 5, 6];
```

```
var num3 = [7, 8, 9];
```

```
var nums = num1.concat(num2, num3);
```

Variáveis indexadas (vetores, arrays unidimensionais)

► Percorrer um vetor (ordem crescente e inversa), calcular media e outras estatísticas são tarefas comuns e básicas com vetores.

```
var v = [];  
var mA, s=0;  
for (let i=0; i<5; i++) {  
    //leia(v[i])  
    s += v[i];  
}  
mA = s / 5;  
//exiba(mA)
```

```
var flag = false;  
// para somar apenas diferentes  
var v = [];  
var mA, s=0;  
for (let i=0; i<5; i++) {  
    v[i] = parseInt(prompt());  
    if (i>0) {  
        for (let j=i-1; j>=0; j--) {  
            if (v[i] == v[j]) {  
                flag=true;  
                i-=1;  
                break;  
            }  
        }  
    }  
    console.log(v[i]);  
    if (!flag) s += v[i];  
    flag = 0;  
}  
console.log('soma: ' + s);
```


Variáveis indexadas (vetores, arrays unidimensionais)

► Código básico para ordenação

```
var v = [3,0,-1,1,2];  
for (let i=0;i < 4;i++) {  
  if (v[i]>v[i+1]) {  
    let aux=v[i];  
    v[i] = v[i+1];  
    v[i+1] = aux;  
    i = -1; // reset  
  }  
}
```

Para crescente >

Para decrescente <

Algoritmo da TROCA (swap)

Vetor ordenado:

-1	0	1	2	3
-----------	----------	----------	----------	----------

Problema 01 com vetores

Peça perdida (OBI 2007, P1, F1):

<https://olimpiada.ic.unicamp.br/pratique/p1/2007/f1/perdida/>

Joãozinho adora quebra-cabeças, essa é sua brincadeira favorita. O grande problema, porém, é que às vezes o jogo vem com uma peça faltando. Isso irrita bastante o pobre menino, que tem de descobrir qual peça está faltando e solicitar uma peça de reposição ao fabricante do jogo. Sabendo que o quebra-cabeças tem N peças, numeradas de 1 a N e que exatamente uma está faltando, ajude Joãozinho a saber qual peça ele tem de pedir.

Tarefa: Escreva um programa que, dado um inteiro N e $N - 1$ inteiros numerados de 1 a N , descubra qual inteiro está faltando.

Entrada: A entrada contém um único conjunto de testes, que deve ser lido do *dispositivo de entrada padrão* (normalmente o teclado). A entrada contém 2 linhas. A primeira linha contém um inteiro N ($2 \leq N \leq 1.000$). A segunda linha contém $N - 1$ inteiros numerados de 1 a N (sem repetições).

Saída: Seu programa deve imprimir, na *saída padrão*, uma única linha, contendo o número que está faltando na sequência dada.

Problema 01 com vetores

Peça perdida (OBI 2007, P1, F1):

Entrada:

3

3 1

5

1 2 3 5

4

2 4 3

Saída:

2

4

1

Problema 02: análise de pseudocódigo

Analise o algoritmo abaixo, verificando a saída para a seguinte sequência de pares de x e y , que devem estar num vetor com 10 elementos:

10 e 5

26 e 13

30 e 22

2 e 1

36 e 10

Início

Enquanto ($y \neq 0$) **Faça**

$r \leftarrow x \bmod y$

$x \leftarrow y$

$y \leftarrow r$

Fim_Enquanto

Exiba(x)

O professor de história precisa dividir uma turma de alunos em grupos, de modo que cada grupo tenha a mesma quantidade de alunos. Nessa turma temos 24 alunas e 16 alunos. Quantos componentes terá cada grupo? (MDC, ALGORITMO DE EUCLIDES)

O piso de uma sala retangular, medindo $3,52 \text{ m} \times 4,16 \text{ m}$, será revestido com ladrilhos quadrados, de mesma dimensão, inteiros, de forma que não fique espaço vazio entre ladrilhos vizinhos. Os ladrilhos serão escolhidos de modo que tenham a maior dimensão possível. Na situação apresentada, o lado do ladrilho deverá medir:

Problema 03:

Projete o algoritmo e implemente um programa de auxílio a uma eleição. Os votos válidos são representados pelos números 1, 2 e 3, cada um correspondendo a um candidato. O voto em branco é representado pelo número 0 e o voto nulo, pelo número -1. Esse fluxograma deverá processar N respostas da votação, as quais são lidas do teclado, separadas por espaço. O programa deve calcular e exibir:

- a) O total de votos para cada candidato
- b) O total de votos em branco
- c) O total de votos nulos
- d) O número do candidato vencedor (ou indicar se não houve vencedor, caso todos os votos tenham sido branco ou anulados)

Dica: para testes considere usar a geração de 100 (ou mais) inteiros aleatórios entre -1 e 3

E estudem como usar o método `filter()` para contar itens no array que atende requisitos

<https://askjavascript.com/javascript-array-count/>

<https://blog.betrybe.com/javascript/javascript-filter/>

Problema 04:

Escreva e implemente uma solução que dado um número natural N , exiba a decomposição do mesmo em fatores primos, assim:

Exemplo:

$$6 = 2(1) 3(1)$$

$$9 = 3(2)$$

$$24 = 2(3) 3(1)$$

Problema 05:

Escreva um programa que gere os N primeiros números perfeitos. Um número perfeito é aquele que é igual a soma dos seus divisores. Exemplo: $6 = 1+2+3$