



INSTITUTO FEDERAL

Norte de Minas Gerais

Campus Januária

Estruturas de Dados I

- Arrays -



Arrays

- Imagine a solução para o seguinte problema...
- Sorteie 30 números inteiros.
- Leia um número inteiro N.
- Imprima na tela a **quantidade de vezes** que o número N foi inserido anteriormente.

Soluções???



■ Imagine

■ Sorteie 30

■ Leia um n

■ Imprima número

```
int main(){  
    int num1,num2,num3,num4,  
    num5,num6,num7,num8,  
    num9,...,num30;
```

```
    num1 = rand() % 10;
```

```
    num2 = rand() % 10;
```

```
    num3 = rand() % 10;
```

```
    (...)
```

```
    num30 = rand() % 10;
```

```
    scanf(" %d", &n);
```

```
    if (num1 == n) cont++;
```

```
    if (num2 == n) cont++;
```

```
    if (num3 == n) cont++;
```

```
    (...)
```

```
    if (num30 == n) cont++;
```

```
}
```

ema...

ue o



INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

■ Imagine

- Sorteie 30 números entre 1 e 100
- Leia um número entre 1 e 100
- Imprima quantas vezes esse número aparece

```
int main(){  
    int num1,num2,num3,num4,  
    num5,num6,num7,num8,  
    num9,...),num30;
```

```
    srand(time(0));  
    for(int i=1; i<=30; i++){  
        num[i] = rand() % 100 + 1;
```

```
    }
```

```
    int n;
```

```
    scanf("%d", &n);
```

```
    int cont = 0;
```

```
    for(int i=1; i<=30; i++){  
        if(num[i] == n) cont++;
```

ema...

ue o



Arrays

- **Array** é uma **estrutura de dados** que permite a **agregação** de um conjunto de **variáveis de mesmo tipo**, sendo **referenciadas** por um **identificador comum**.
- **Vetor** == Array Unidimensional

```
int v[30];
```

- **Matriz** == Array Multidimensional

```
int m[10][10]
```




Representação

■ Representação de um **Vetor** (*Array Unidimensional*)

```
int main(){  
    int v[30];  
    v[2] = 8;  
    v[5] = v[2]*2;  
}
```

Índice	0	1	2	3	4	5	6	29
Valor			8			16				



Representação

■ Representação de uma **Matriz** (Array N-dimensional)

```
int main(){
    int m[4][10];
    m[3][2] = 7;
    m[2][4] = m[3][2]*3;
}
```

Índices	0	1	2	3	4	5	6	7	8	9
0										
1										
2					21					
3			7							



Atenção!

- O primeiro índice de um **array** sempre será o valor 0.

0	1	2	3	4	5	6	29
¶	¶	8	¶	¶	16	¶	¶	¶	¶

- Ao declarar um **array**, é comum que o espaço alocado na memória para armazenar o conteúdo possua resíduos (**lixo de memória**) de outros processos em execução no computador.



Atenção!

- Também é possível inicializar um **array** já com seus valores...

```
int k[6] = {0,2,4,6,8,10}
```

0	1	2	3	4	5
0	2	4	6	8	10

- Porém, essa prática não é comum, pois geralmente os valores de um **array** só serão conhecidos em tempo de execução.



Navegando por um Array

- Para se navegar pelos elementos de um Array, é muito comum a utilização de uma **estrutura de repetição** (que quase sempre é o ***FOR***)

```
int main(){  
    int vet[30];  
    for (int i=0; i<30; i++)  
        scanf(" %d",&vet[i]);  
}
```



Navegando por um Array

■ Array Bidimensional

```
int main(){  
    int mat[30][20];  
    for (int i=0; i<30; i++)  
        for (int j=0; j<20; j++)  
            printf(" %d ",mat[i][j]);  
}
```

Vamos à Prática!

1. Faça um programa que sorteie um vetor com 100 números aleatórios entre 0 e 9. Peça para o usuário informar um número X e imprima quantas vezes o número X foi sorteado anteriormente.
2. Um apostador joga um dado para o ar N vezes. Sabendo que um dado possui 6 faces, faça um programa que simule o experimento, e imprima quantas vezes que cada face caiu para cima.
3. Faça um programa que gera aleatoriamente uma aposta da lotofácil (15 números não repetitivos no intervalo entre 01 e 25).
4. Sorteie números aleatórios para preencher uma tabela NxN (o valor N definido pelo usuário). Imprima os n^ºs em formato de tabela.