



**INSTITUTO FEDERAL**

Norte de Minas Gerais

Campus Januária

# Estruturas de Dados I

*- Arrays -*



# Arrays

- **Imagine a solução para o seguinte problema...**
- **Sorteie 10 números inteiros aleatórios.**

**Soluções???**



# Arrays

- Imagine a solução para o seguinte problema...
- Sorteie 10 números inteiros aleatórios **não-repetitivos**.

**Soluções???**



# Arrays

```
int main(){
    int num1,num2,num3,num4,num5,
        num6,num7,num8,num9,num10;

    num1 = rand() % 25;

    do{
        num2 = rand() % 25;
    }while(num2==num1);

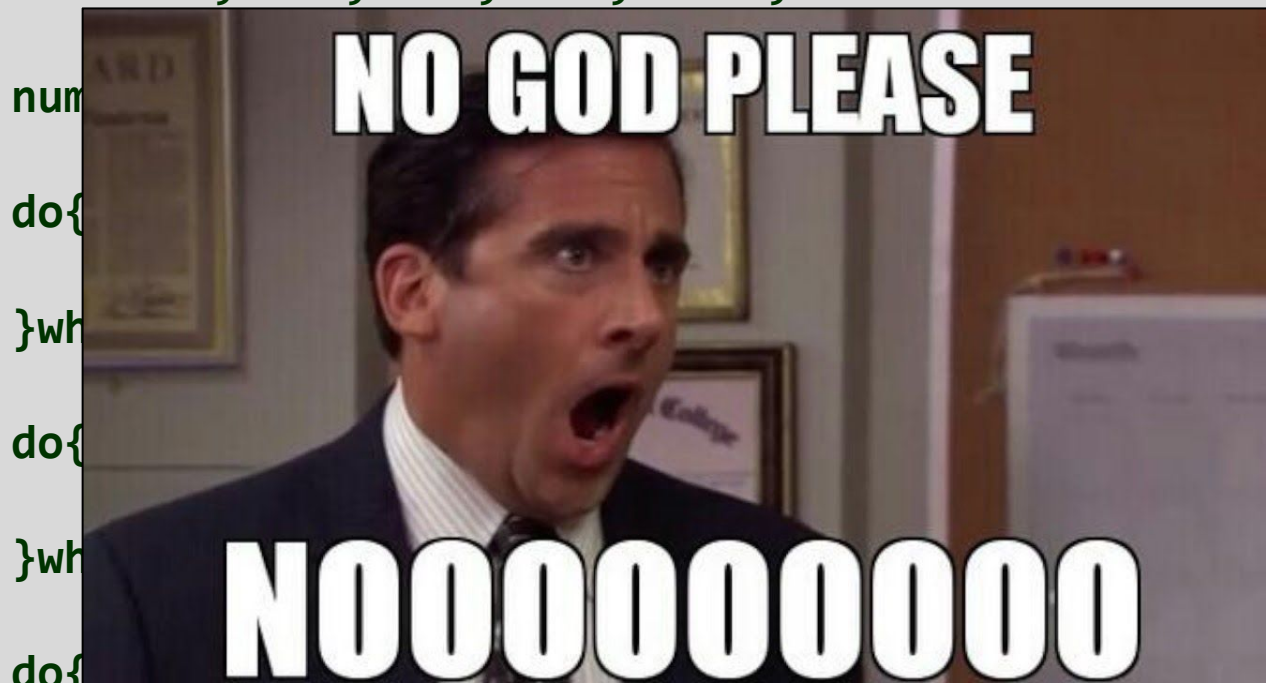
    do{
        num3 = rand() % 25;
    }while((num3==num1) || (num3==num2));

    do{
        num4 = rand() % 25;
    }while((num4==num1) || (num4==num2) || (num4==num3));
    (...)
}
```



# Arrays

```
int main(){
    int num1,num2,num3,num4,num5,
        num6,num7,num8,num9,num10;
```



```
num4 = rand() % 25;
}while((num4==num1) || (num4==num2) || (num4==num3));
(...)
}
```





# 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
7	15	8	9	10	16	17	...	...	28

- 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** com valores:

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

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

```
int k[10] = {0}
```

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



# 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++) //navega pelas linhas  
        for (int j=0; j<20; j++)//colunas de cada linha  
            printf(" %d ",mat[i][j]);  
}
```



# Bora CODAR!!!

1. Faça um programa que sorteie um vetor com 100 números aleatórios entre 1 e 20. Peça ao usuário um número X e imprima quantas vezes este número foi sorteado (marque também um \* ao lado de cada vez o valor X aparecer na impressão). Repita a operação para o mesmo conjunto de dados até  $X \leq 0$ .
2. Um apostador joga um dado para o ar N vezes. Sabendo que um dado possui 6 faces, faça um programa que simula 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 N x N (o valor N definido pelo usuário). Imprima os n<sup>os</sup> em formato de tabela.
5. Gere uma matriz N x N onde os valores da diagonal principal sejam a soma dos valores existentes na sua linha e coluna.