

Aula 4 – Modulo Iniciante



Recapitulando...

Até agora estudamos:

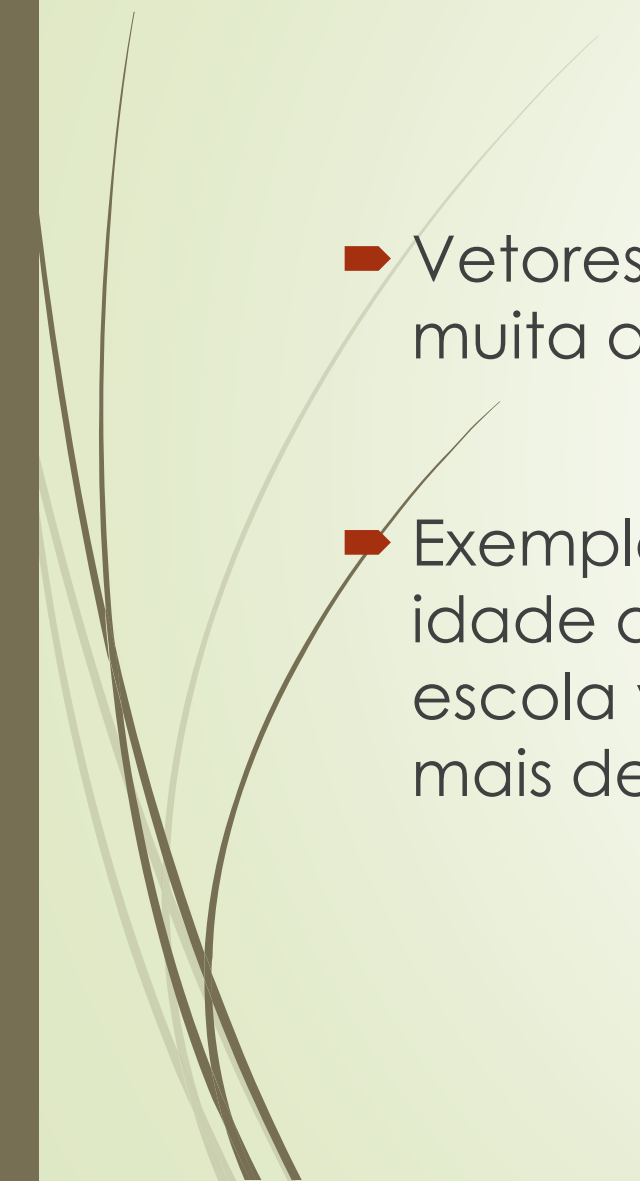
- Estrutura de um código
- Variáveis e seus tipos
- Operadores Matemáticos, Lógicos, de Incremento, de Decremento e de Comparação
- Printf e Scanf
- If – Else
- While e For

Vetores

- Vetores(ou arrays, em ingles), são nada mais que UM MONTE DE VARIÁVEIS, com O MESMO NOME, localizadas uma do lado da outra na memória.
- Elas são distinguidas umas das outras pelo ÍNDICE DELAS NO VETOR. Uma maneira de imaginar um vetor, é como se fosse um trem, e cada vagão do trem é uma posição do vetor.



Quando e por que usar vetores?

- Vetores são utilizados quando você vai trabalhar com muita quantidade de dados.
 - Exemplo: Se você for criar um programa pra armazenar a idade dos alunos de uma escola, com certeza essa escola vai ter muitos alunos, muitos mesmo, talvez até mais de 1000.
- 

Quando e por que usar vetores?

**Pensando
no exemplo
passado**

- Como vocês já sabem, para cada aluno, eu teria que criar uma variável, pra armazenar a idade dele.
- Mas, vocês devem concordar comigo que não seria nada legal se eu tivesse que declarar 1000 variáveis diferentes, uma por uma, e depois colocar 1000 scanf's diferentes pra ler esses valores.
- Para solucionar esse problema, nós utilizamos vetores. Ao invés de criar 1000 variáveis diferentes uma por uma, criamos um vetor com 1000 variáveis diferentes uma por uma, criamos um vetor com 1000 posições, onde todas as variáveis possuem o mesmo nome, e são distinguidas apenas pelo índice de suas posições.

Como declarar um vetor

- Declarar um vetor é quase a mesma coisa de se declarar uma variável, a diferença é que você precisa dizer pro computador quantas POSIÇÕES o seu vetor vai ter, entre colchetes “[]”.

Exemplo: Declaração de um vetor de inteiros chamado “banana”, com 100 posições

```
#include <stdio.h>
main() {
    int banana[100];
}
```

* Os vetores pode ser de qualquer tipo: int, float, double, char, long long int, ...

Como percorrer o vetor(ler, printar, ...)

- Para se percorrer um vetor, é preciso ir da primeira posição, até a ultima, certo?!
- A primeira posição de um vetor é sempre a posição 0. Logo, se o seu vetor tem 100 posições, a primeira vai ser a posição 0, e a ultima vai ser a posição 99.
- Então, pensem comigo, se eu preciso percorrer o vetor, posição por posição, um número determinado de vezes, eu uso o FOR, onde o contador começa em 0, e repete enquanto o contador for menor que o número de posições.

Como percorrer o vetor(ler, printar, ...)

Exemplo: Leia um número N ($1 \leq N \leq 1000$). Em seguida N valores.

```
#include <stdio.h>
main(){
    int cont, vetor[1100], N;
    scanf("%d", &N);
    for(cont = 0; cont < N; cont++){
        scanf("%d", &vetor[cont]);
    }
}
```

******Percebam que N pode ser no máximo igual a 1000. Mas eu declarei 1100, por quê ?? Essa é uma técnica usada em maratonas, olimpíadas e uri, para evitar que você declare um vetor com um número insuficiente de posições, então pegue essa prática.

Matrizes

- ▶ Matrizes são estruturas de dados do tipo vetor com duas ou mais dimensões.
- ▶ Um vetor pode ser representado como uma linha:

34	23	1	-12	2	344
----	----	---	-----	---	-----

- ▶ Uma matriz de duas dimensões pode ser representada como uma tabela:

324	23	53	-13	75	123
0	43	76	32	41	-51
213	-12	45	87	39	22



Vetores

- Podemos pensar os vetores como prédios que possuem apenas um apartamento por andar.

Matrizes

- Podemos pensar as matrizes de duas posições como prédios que possuem vários andares e que cada andar possui vários apartamentos.

Como declarar uma matriz?

- Assim como os vetores, os itens de uma matriz devem ser sempre do mesmo tipo.
- Declaramos quase que da mesma forma que declaramos vetores:
`tipo nome[tamanho][tamanho]...[tamanho];`
- O tipo da matriz pode ser o mesmo que o de qualquer vetor (int, double, float, char, ...).
- A quantidade de `[]` indica a quantidade de dimensões que a matriz terá.
- ***O tamanho das dimensões não precisa ser o mesmo!**

Exemplo de Matriz

- Supondo que estudamos em uma escola que possui dois andares e 10 salas por andar, podemos criar uma matriz para saber quantos alunos estudam em cada sala da seguinte maneira:

```
int alunos[2][10];
```

- Assim, para sabermos quantos alunos estudam na segunda sala do primeiro andar:

```
printf("Alunos: %d\n", alunos[0][1]);
```

*Lembre-se de que em um vetor ou em uma matriz a primeira posição é indicada com 0 e não com 1!



E pra finalizar...

- Exercícios URI
- 1172
- 1174
- 1177
- 1181
- 1183
- 1187