Resumo da Aula: Vetores (Arrays)

Nesta aula, aprendemos sobre um novo tipo de variável em programação chamado \*\*vetor\*\* (ou array).  
Se você já usou variáveis simples como `int idade = 20;`, um vetor é como uma \*\*caixinha com várias gavetas\*\*,   
onde cada gaveta pode guardar um número.  
  
### O que é um vetor?  
Um vetor é uma \*\*estrutura de dados\*\* que guarda \*\*vários valores do mesmo tipo\*\* (como vários números inteiros, por exemplo).  
Esses valores são armazenados lado a lado na memória do computador e têm \*\*índices\*\* que começam em 0.  
  
Exemplo de vetor com 5 números:  
 int numeros[5];  
  
Isso significa que temos espaço para 5 inteiros: os índices serão 0, 1, 2, 3 e 4.  
  
### Para que servem os vetores?  
Eles são úteis quando precisamos \*\*guardar vários valores\*\*, como:  
- Notas de 5 alunos  
- Idades de 10 pessoas  
- 30 nomes de clientes  
  
### Como usar vetores em C:  
1. Declarar: int notas[5];  
2. Preencher: notas[0] = 10;  
3. Acessar: printf("%d", notas[0]);  
  
### Usando laços com vetores  
É comum usar `for` para preencher ou mostrar todos os elementos:  
 for (int i = 0; i < 5; i++) {  
 scanf("%d", &notas[i]);  
 }  
  
### Exemplo de programa com vetor:  
 int numeros[5] = {1, 2, 3, 4, 5};  
 for (int i = 0; i < 5; i++) {  
 printf("%d ", numeros[i]);  
 }  
  
### Importante lembrar:  
- Os índices \*\*sempre começam em 0\*\*.  
- O vetor tem um \*\*tamanho fixo\*\* (não muda durante a execução).  
- Os vetores podem ser de tipos diferentes: `int`, `char`, `float` etc.  
  
### Exemplos da aula:  
- Mostrar os números armazenados no vetor  
- Separar positivos e negativos  
- Somar todos os valores de um vetor  
- Trabalhar com texto como vetores de caracteres (strings)  
  
Com prática, vetores se tornam uma ferramenta poderosa para armazenar e organizar dados de forma simples e eficiente.