

Vetores

Como armazenar 3 notas?

```
float nota1, nota2, nota3;

printf("Nota do aluno 1: ");
scanf("%f", &nota1);
printf("Nota do aluno 2: ");
scanf("%f", &nota2);
printf("Nota do aluno 3: ");
scanf("%f", &nota3);
```

Como armazenar n (≤ 100) notas?

```
float nota1, nota2, nota3, ..., nota100;
if (n >= 1){
    printf("Nota do aluno 1: ");
    scanf("%f", &nota1);
}
if (n >= 2){
    printf("Nota do aluno 2: ");
    scanf("%f", &nota2);
} ...
if (n == 100){
    printf("Nota do aluno 100: ");
    scanf("%f", &nota100);
}
```

Como armazenar 100 notas?

```
float nota1, nota2, nota3, ..., nota100;

printf("Nota do aluno 1: ");
scanf("%f", &nota1);
printf("Nota do aluno 2: ");
scanf("%f", &nota2);

...

printf("Nota do aluno 100: ");
scanf("%f", &nota100);
```

Vetores

- Coleção de variáveis do mesmo tipo
- Armazenadas contiguamente na memória
- Acessadas pelo mesmo nome por meio de um índice
- Todos os elementos são igualmente acessíveis, isto é, o tempo e o tipo de procedimento para acessar qualquer um dos elementos são iguais

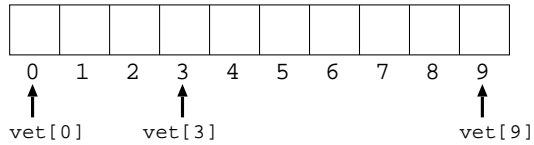
Como armazenar n (≤ 100) notas?

Declaração de Vetores

- Sintaxe:
`<tipo> identificador [<tamanho>];`
- A primeira posição de um vetor tem índice 0.
- A última posição de um vetor tem índice `<tamanho> - 1`.
- Exemplo:
`float notas[100];`

Exemplo

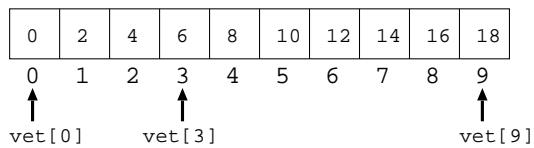
• `int vet[10];`



`nota[5] = 9.5;`

Inserindo dados

```
for (i = 0; i < 10; i++){
    vet[i] = 2 * i;
}
```



Exemplo

```
float nota[100], media;
int n, i;

printf("Numero de alunos: ");
scanf("%d", &n);

for (i = 0; i < n; i++) {
    printf("Nota do aluno %d: ", i+1);
    scanf("%f", &nota[i]);
}

media = 0;
for (i = 0; i < n; i++) {
    media = media + nota[i];
}
media = media / n;
printf("Nota media = %.1f\n", media);
```

Usando um vetor

`a = identificador [<posição>];`

- Cada elemento (posição) do vetor comporta-se como uma variável: retorna o seu valor como uma expressão e pode ter valores atribuídos
- Pode-se substituir uma variável de um determinado tipo por um único elemento de um determinado vetor.

`a = nota[10];`

Inicialização

- Pode ser inicializado no momento de sua declaração:


```
int vetor1[3] = {1, 2, 3};
int vetor2[10] = {1, 2, 3};
int vetor3[] = {1, 2, 3};
```
- Incorreto:


```
int vetor[2] = {1, 2, 3};
```

Vetores

```
int d; int vetor[5]; int f;
```

Nome	d	vetor					f
Índice	-	0	1	2	3	4	-

Importante!

- O tamanho do vetor é pré-definido. (Ou seja, após a compilação o tamanho não pode ser mudado)
- Índices fora dos limites podem causar comportamento anômalo do código.

```
vetor[3]=10;
```

Nome	d	vetor					f
Índice	-	0	1	2	3	4	-
					10		

```
d = vetor[3];
```

Nome	d	vetor					f
Índice	-	0	1	2	3	4	-
	10				10		

Exemplo

```
char aster[16], cifroes[16], arrobas[16];
int i;

printf("Endereço de arrobas 0x%x (%u)\n", arrobas, arrobas);
printf("Endereço de cifroes 0x%x (%u)\n", cifroes, cifroes);
printf("Endereço de aster 0x%x (%u)\n", aster, aster);
```

```
for (i = 0; i < 16; i++) {
    aster[i] = '*';
    cifroes[i] = '$';
    arrobas[i] = '@';
}
```

```
for (i = 0; i < 16; i++)
    printf ("%c", aster[i]);
printf ("\n");
```

```
for (i = 0; i < 16; i++)
    printf ("%c", cifroes[i]);
printf ("\n");
for (i = 0; i < 16; i++)
    printf ("%c", arrobas[i]);
printf ("\n");
```

```
for (i = 0; i < 40; i++)
    arrobas[i] = '@';
```

```
for (i = 0; i < 16; i++)
    printf ("%c", aster[i]);
printf ("\n");
for (i = 0; i < 16; i++)
    printf ("%c", cifroes[i]);
printf ("\n");
for (i = 0; i < 16; i++)
    printf ("%c", arrobas[i]);
```

Vetores

- O que ocorre se digitar os comandos:
vetor[5]=5;
vetor[-1]=1;

Nome	d	vetor					f
Índice	-	0	1	2	3	4	-
	1				10		5

Polinômios

Lê os coeficientes de um polinômio. Você deve definir o grau máximo.

```
float coef[26];
int grau;

for (i = grau; i >= 0; i--) {
    printf("coeficiente de x^%d: ", i);
    scanf("%f", &coef[i]);
}
```

Vetor

```
void le_vet(int vet[], int n_elem) {
    int i;
    for (i = 0; i < n_elem; i++) {
        printf("vet[%d]: ", i);
        scanf("%d", &vet[i]);
    }
}

void escreve_vet(int vet[], int n_elem) {
    int i;
    for (i = 0; i < n_elem; i++)
        printf("vet[%d]: %d\n", i, vet[i]);
}
```

Polinômio

Imprime cuidadosamente (utilizando sinais de + e -) o polinômio

```
printf("%.1fx^%d", coef[grau], grau);
for (i = grau - 1; i >= 0; i--) {
    if (coef[i] != 0) {
        if (coef[i] >= 0) {
            printf(" + %.1fx^%d", coef[i], i);
        } else {
            printf(" - %.1fx^%d", -coef[i], i);
        }
    }
}
```

Exercícios

1. Ler 10 valores e armazená-los em um vetor. Escrever uma função que soma todos os valores lidos e retorna o resultado da soma.
2. Ler 10 valores e armazená-los em um vetor. Escrever uma função que verifica o maior e o menor número lido.
3. Utilizar o exercício anterior para armazenar o menor número na primeira posição do vetor e o maior número na última posição do vetor.

Vetor

```
void altera(int vet[], int x) {
    vet[0] = -1;
    x = -1;
}

main() {
    int vet[] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
    int x = 0;

    altera(vet, x);
    printf("vet[0] = %d", vet[0]);
    printf("x      = %d", x);
}
```

Exercícios

4. Escreva uma função que ordene um vetor de 20 elementos.
5. Escreva uma função que rearranja um vetor de 20 elementos colocando os números pares antes dos ímpares.