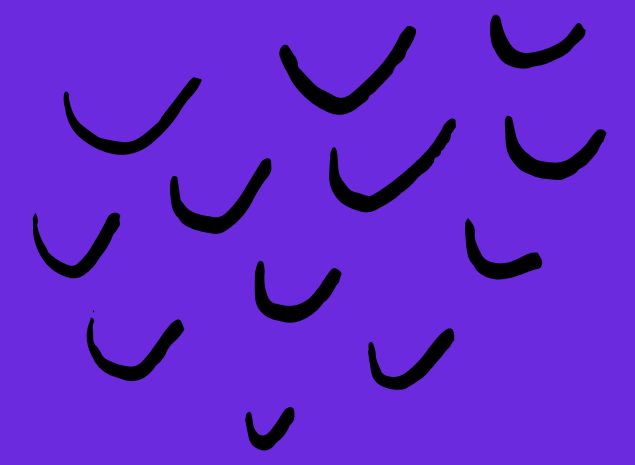
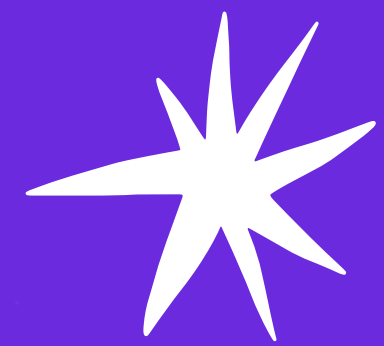


VARIÁVEIS COMPOSTAS HOMOGÊNEAS



Vetores

unidimensional





NÃO PODEMOS RESOLVER
PROBLEMAS USANDO O
MESMO TIPO DE
PENSAMENTO QUE USAMOS
QUANDO OS CRIAMOS.



ALBERT EINSTEIN

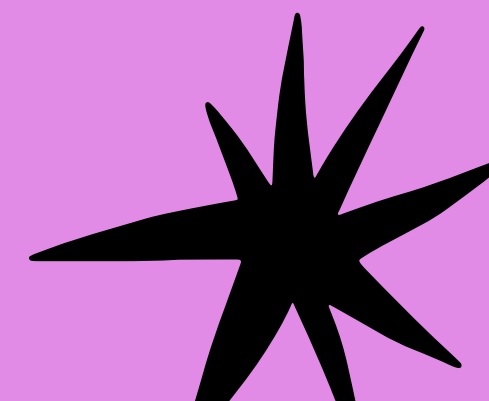
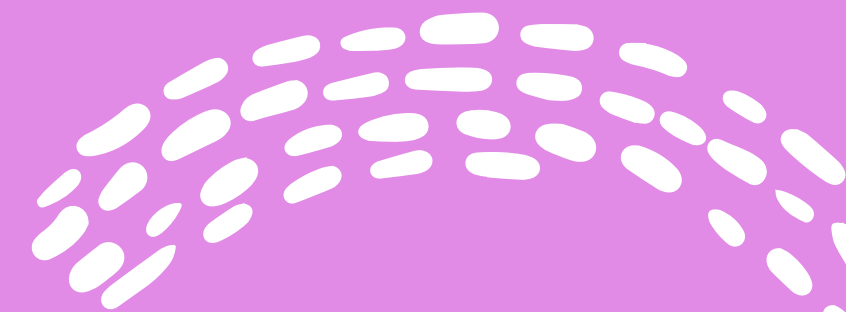
NECESSIDADE



Armazenamento de dados semelhantes

Exemplo:

Crie um programa que leia a nota e o nome de três alunos e mostre quais alunos ficaram acima da média



```
1 #include <stdio.h>
2 int main(){
3     float notal, nota2, nota3, media;
4     char nome1[20], nome2[20], nome3[20];
5     printf("Nome do aluno 1: ");
6     scanf("%[^\n]", nome1);
7     printf("Nota do aluno 1: ");
8     scanf("%f", &notal);
9     printf("Nome do aluno 2: ");
10    scanf("%[^\n]", nome2);
11    printf("Nota do aluno 2: ");
12    scanf("%f", &nota2);
13    printf("Nome do aluno 3: ");
14    scanf("%[^\n]", nome3);
15    printf("Nota do aluno 3: ");
16    scanf("%f", &nota3);
17    media = (notal + nota2 + nota3) / 3;
18    printf("\nMedia = %.1f\n", media);
19    if (notal > media) {
20        printf("\n%s obteve nota acima da media", nome1);
21    }
22    if (nota2 > media) {
23        printf("\n%s obteve nota acima da media", nome2);
24    }
25    if (nota3 > media) {
26        printf("\n%s obteve nota acima da media", nome3);
27    }
28 }
```

lllll

```
Nome do aluno 1: lucilia
Nota do aluno 1: 10
Nome do aluno 2: bruna
Nota do aluno 2: 2.5
Nome do aluno 3: andre
Nota do aluno 3: 6.1

Media = 6.2

lucilia obteve nota acima da media
```



E SE FOSSEM **100**
ALUNOS???

CRIAR 100 VARIÁVEIS

**100 LINHAS PARA
ENTRADA DE DADOS**

**100 COMPARAÇÕES
COM A MÉDIA!!!!**



A SOLUÇÃO É ARMAZENAR EM UM VETOR

DADOS DO ALUNO 1
Nome.....: lucilia
Nota.....: 10

DADOS DO ALUNO 2
Nome.....: bruna
Nota.....: 8.5

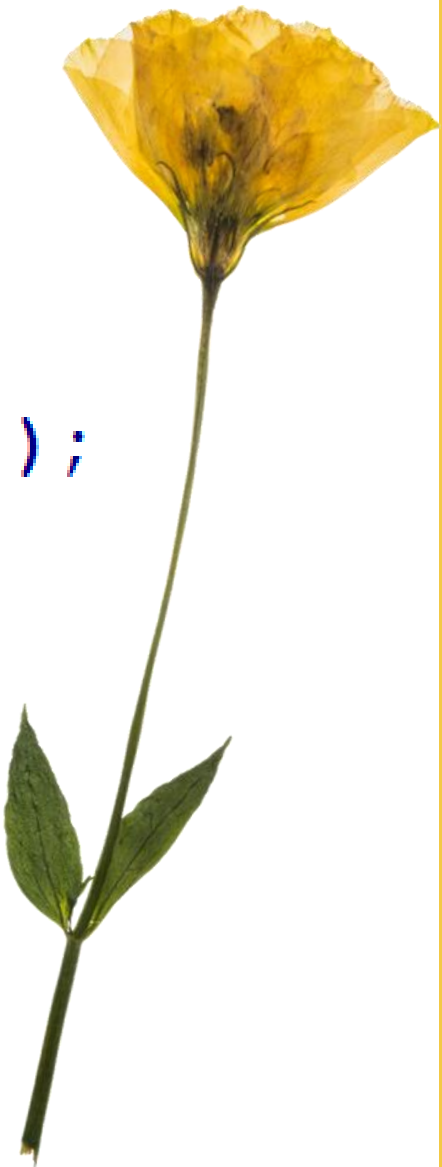
DADOS DO ALUNO 3
Nome.....: andre
Nota.....: 3

Media = 7.2

ALUNOS ACIMA DA MEDIA...

NOME	NOTA
lucilia	10.0
bruna	8.5

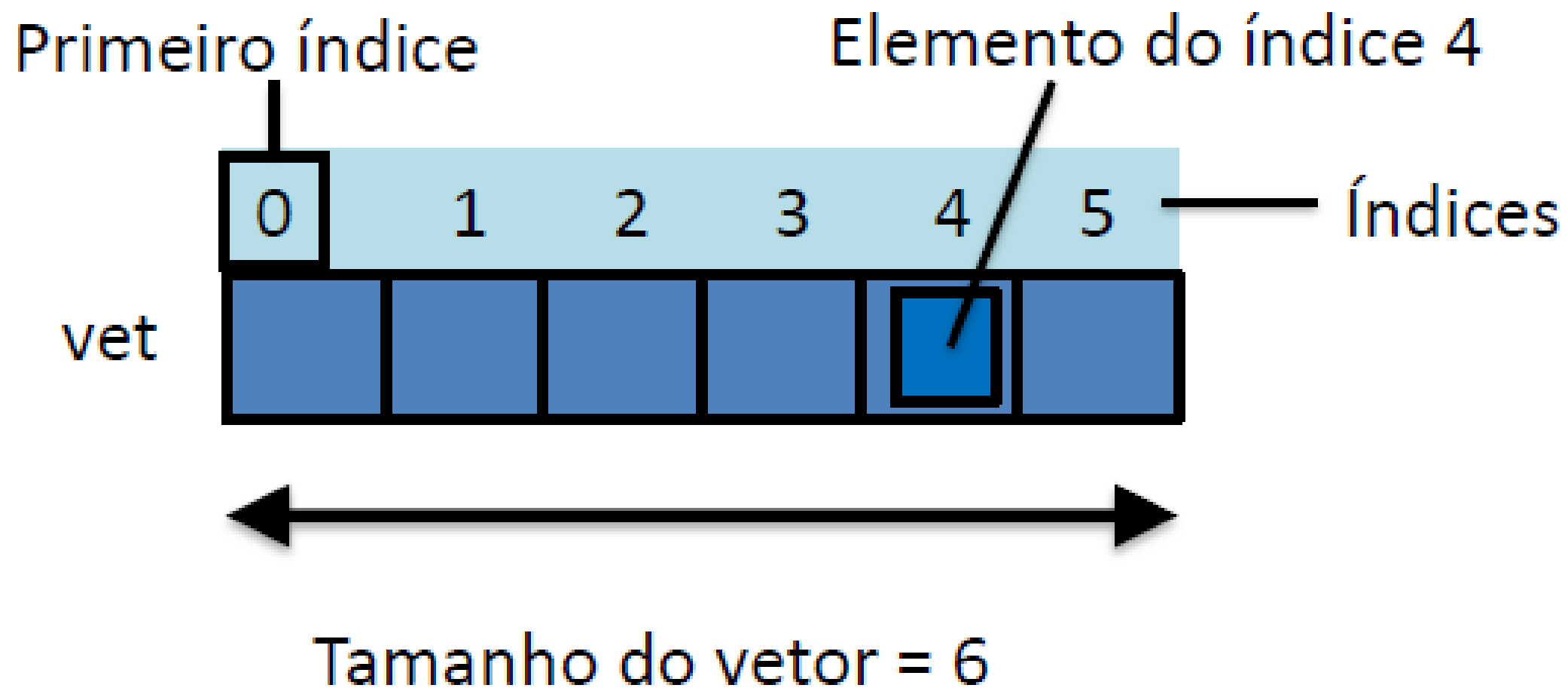
```
1  #include <stdio.h>
2  int main() {
3      const int DIM = 3;
4      float nota[3], media, somaNotas = 0;
5      char nome[3][20];
6      for (int i = 0; i < DIM; i++) {
7          printf("\nDADOS DO ALUNO %d\n", (i + 1));
8          printf("Nome.....: ");
9          scanf("%[^\n]", nome[i]);
10         printf("Nota.....: ");
11         scanf("%f", &nota[i]);
12         somaNotas = somaNotas + nota[i];
13     }
14     media = somaNotas / DIM;
15     printf("\nMedia = %.1f\n", media);
16     printf("\n\nALUNOS ACIMA DA MEDIA...\n");
17     printf("%-30s \t %5s", "NOME", "NOTA");
18     for (int i = 0; i < DIM; i++) {
19         if (nota[i] > media) {
20             printf("\n%-30s \t %5.1f", nome[i], nota[i]);
21         }
22     }
23     return 0;
24 }
```



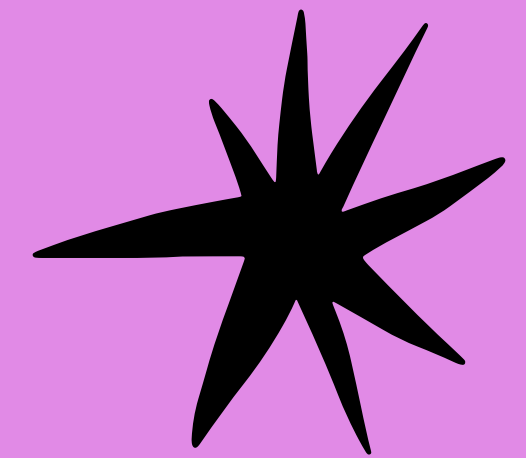
VETOR

0 que é

Variável composta de vários valores do mesmo tipo, acessíveis por um índice



VETOR



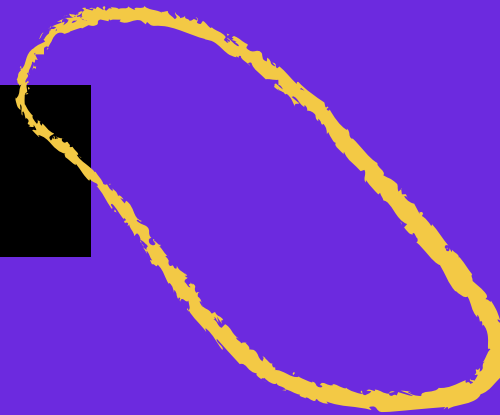
- Sequência de vários valores do mesmo tipo, armazenadas sequencialmente na memória, e fazendo uso de um mesmo nome de variável para acessar valores
- Cada elemento desta sequência pode ser acessado individualmente através de um índice (número inteiro)
- Os elementos são indexados de 0 até $n - 1$, onde n é a quantidade de elementos do vetor, também chamado de dimensão ou tamanho do vetor
- O vetor tem tamanho fixo (definido na declaração)



ACESSO AO CONTEÚDO DO VETOR

Realizado por meio do operador de índice

- Utiliza uma referência de memória (variável do tipo vetor)
- Utiliza um índice (número inteiro)
- Retorna uma referência para o elemento correspondente ao índice (valor)

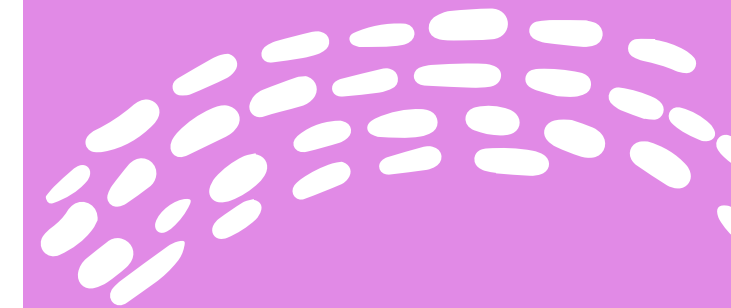
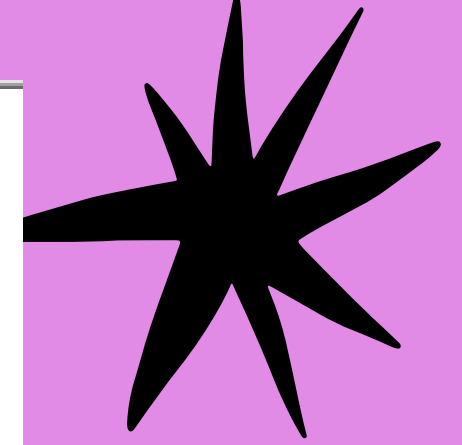


EXEMPLO

Faça um programa que leia dez números, os imprima em ordem inversa e mostre o maior e o menor número. Para isso, é necessário armazenar os dez números para poder imprimi-los.



```
1  #include <stdio.h>
2  int main() {
3      const int TAMANHO = 10;
4      int maior, menor, numero[TAMANHO];
5      printf("Digite 10 numeros inteiros:\n");
6      for (int i = 0; i < TAMANHO; i++) {
7          printf("[%d]: ", (i + 1));
8          scanf("%d", &numero[i]);
9          if (i == 0) {
10             maior = numero[i];
11             menor = numero[i];
12         } else {
13             if (numero[i] > maior) {
14                 maior = numero[i];
15             }
16             if (numero[i] < menor) {
17                 menor = numero[i];
18             }
19         }
20     }
21     printf("\nValores em ordem reversa: ");
22     for (int i = (TAMANHO - 1); i >= 0; i--) {
23         printf("%d, ", numero[i]);
24     }
25     printf("\nMaior = %d e Menor = %d\n\n", maior, menor);
26 }
```





```
Digite 10 numeros inteiros:
```

```
[1]: 5  
[2]: 1  
[3]: 25125  
[4]: -18  
[5]: -5  
[6]: -1  
[7]: 0  
[8]: 8  
[9]: 6  
[10]: 7
```



```
Valores em ordem reversa: 7, 6, 8, 0, -1, -5, -18, 25125, 1, 5,  
Maior = 25125 e Menor = -18
```



OBRIÇADA!

Alguma Pergunta?

professora@lucilia.com.br

