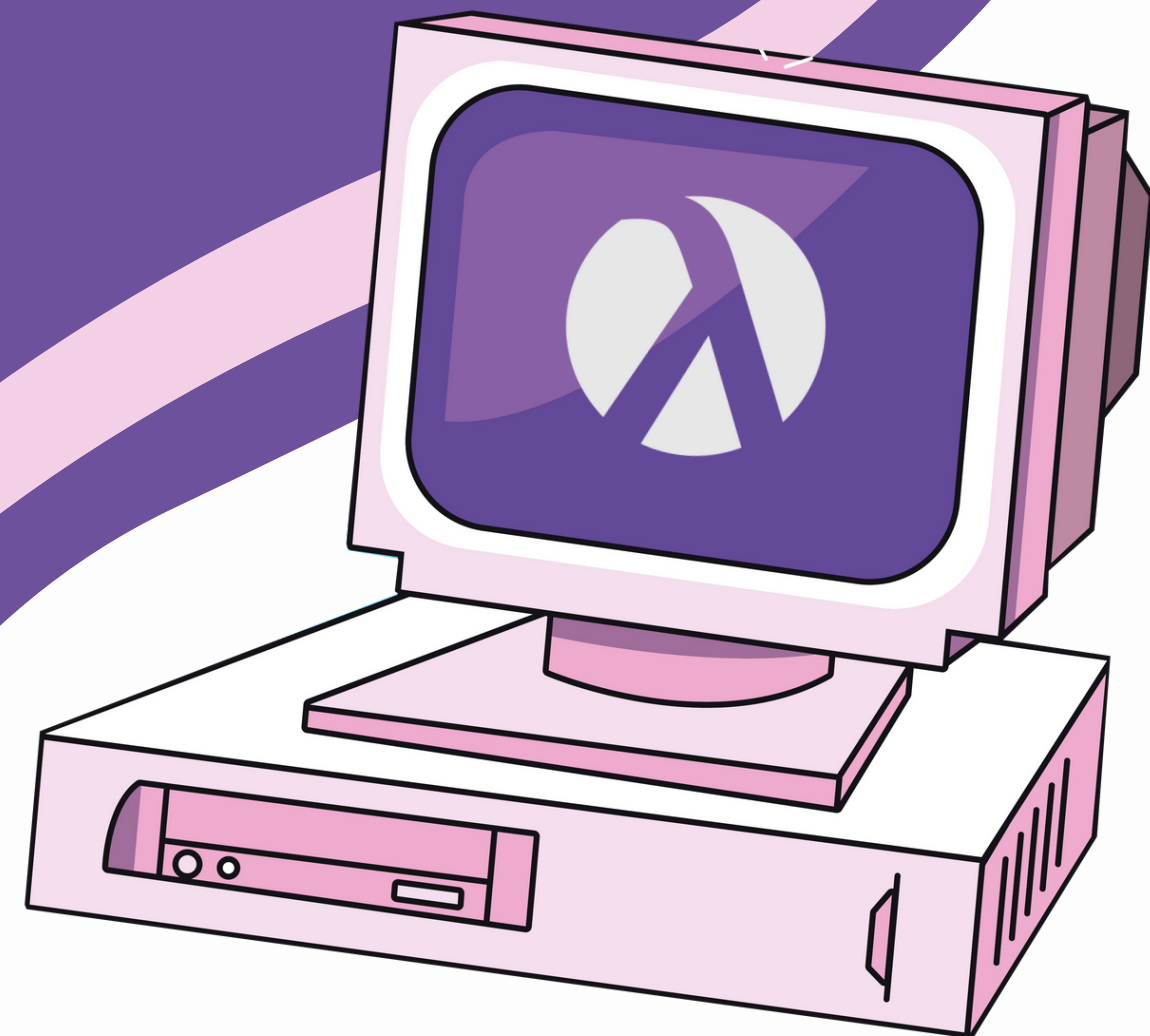


LÓGICA DE PROGRAMAÇÃO

EC

DIA 2



Revisando...

Scanf!

- Para que uma variável receba um valor, devemos fazer a leitura utilizando o scanf:

`scanf("%d", &variável);`

Qual variável irá receber



Indica o tipo da
variável: int, float, char...

Exemplos de leitura!



```
float salario;  
scanf("%f", &salario);
```

```
char estado[15];  
scanf("%s", estado);
```



```
int idade;  
char inicial_nome;
```

```
scanf("%d %c", &idade, &inicial_nome);
```

Revisando...

printf!

- A nossa saída dos programas, utilizado para mostrar ao usuário os valores dos dados!

```
printf("Salário: %f", variável );
```

Indica que a var
é do tipo float

Qual variável é
mostrada



Como o PC lê?



- Já dizia um sábio: "Ahh, mas isso aí é muito simples! O PC lê de cima pra baixo em todas as linhas."
- Pensa assim: *Mostrar uma mensagem ao usuário, pedindo o nome dele.*

Como o PC lê?



```
1  #include <stdio.h>
2
3  int main(){
4      char nome[10];
5      printf("Digite seu nome: ");
6      scanf("%s", nome);
7
8      printf("Nome digitado: %s", nome);
9  }
```

Como o PC lê?



```
1  #include <stdio.h>
2
3  int main(){
4      char nome[10];
5      printf("Digite seu nome: ");
6      scanf("%s", nome);
7
8      printf("Nome digitado: %s", nome);
9  }
```

Digite seu nome: Fran

Nome digitado: Fran

Bibliotecas



Bibliotecas

- O que são?
- Como utilizá-las
- Principais Bibliotecas



O que são?

- Bibliotecas são conjuntos de funções prontas que podem ser utilizadas pelo programador para facilitar a implementação do código.
- O conjunto de funções de cada biblioteca é descrito em um arquivo-interface (header-file), que tem o mesmo nome da biblioteca e sufixo .h
- Exemplos:

`stdio.h`

`stdlib.h`

`math.h`



Como utilizá-las



- Para ter acesso a uma biblioteca, o programador deve incluí-la em seu programa, caso a biblioteca não seja incluída, suas funções não poderão ser utilizadas.
- Em C, a sintaxe para o acesso de bibliotecas é utilizando a diretiva `#include`
- Exemplos:

```
#include <stdio.h>
#include <stdlib.h>
```

Principais Bibliotecas



- Ao longo do curso, serão utilizadas algumas bibliotecas, dentre elas as mais importantes são: `stdlib`, `stdio` e `string`.
- `stdio`: A biblioteca `stdio` é responsável pelas funções de entrada e saída.

```
1  #include <stdio.h>
2
3  ✓ int main()
4  {
5      int i=0;
6      printf("Qual sua idade?\n");
7      scanf("%d",&i);
8  }
```

Principais Bibliotecas



- **stdlib:** A biblioteca `stdlib.h` é responsável por funções de randomização, alocação de memória e constantes básicas.
- **string:** A biblioteca `string.h` é utilizada por possuir funções de manipulações de string.

```
1  #include <stdlib.h>
2
3  #define NULL 0
4
5  int rand (void);
6  void *malloc (size_t N);
7  void free (void*);
```

```
1  #include <stdio.h>
2  #include <string.h>
3
4  int main()
5  {
6      int t;
7      char string_d[10], string_o[10];
8      strncpy(string_d, string_o, t);
9      strcmp(string_d, string_o);
10     strlen(string_d);
11 }
```

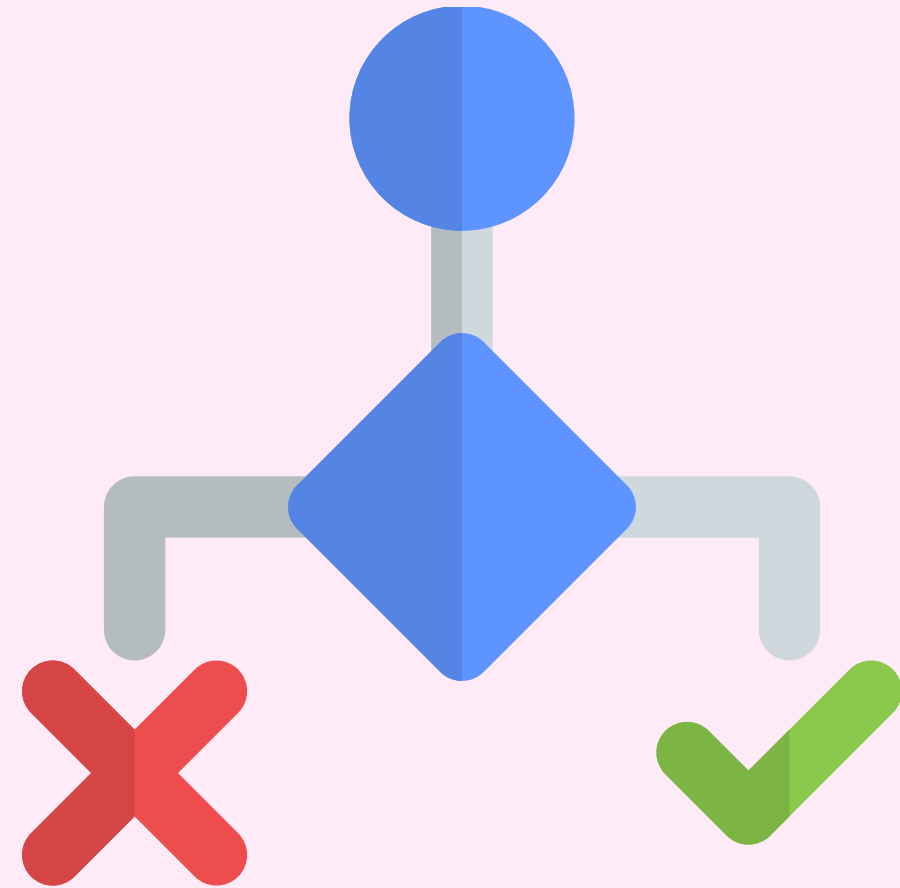
Estruturas Condicionais



Estruturas Condicionais



- O que são condicionais?
- Como utilizá-las
- Switch



O que são condicionais?



- Chamamos de estruturas condicionais as instruções para testar se uma condição é verdadeira ou falsa.
- Essas estruturas permitem com que o programa siga caminhos alternativos durante a execução, de acordo com uma condição.

O que são condicionais?



- Chamamos de estruturas condicionais as instruções para testar se uma condição é verdadeira ou falsa.
- Essas estruturas permitem que o programa siga caminhos alternativos durante a execução, de acordo com uma condição.
- Exemplo: Se eu sair para a festa hoje então voltarei tarde, senão pedirei pizza.
- Eu pedi pizza. Logo, eu não fui à festa.

Como utilizá-las



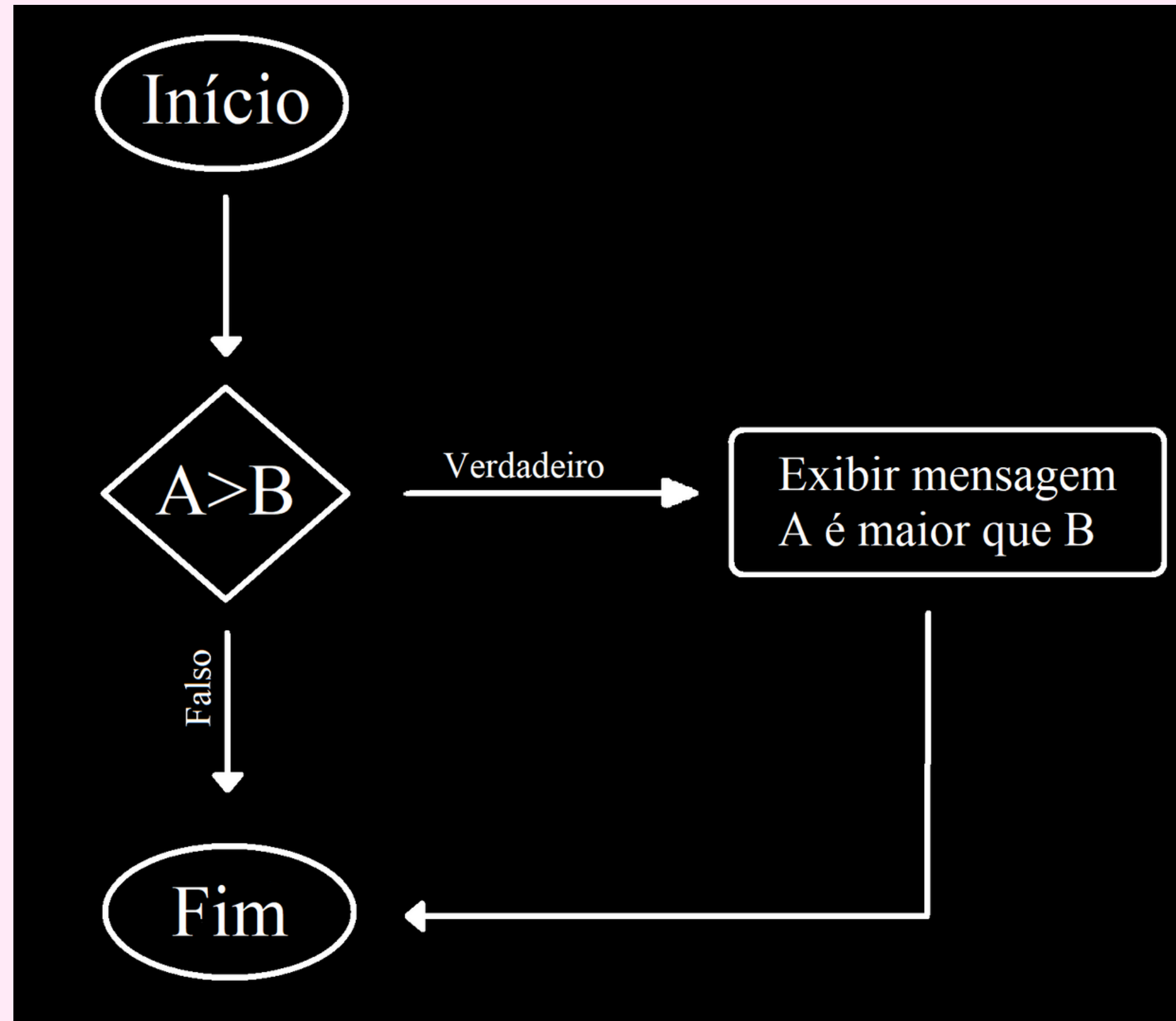
- Algumas das condicionais que utilizamos em C são: **if()**; **else**; **else if()**;

- Utilizamos a estrutura **if()** como uma condição inicial. Por exemplo: **if(a > b)**.

- Na imagem, testamos se a condição (a > b) é verdadeira. No caso de ser verdade o programa printará "a > b".

```
1  #include <stdio.h>
2
3  int main() {
4      int a, b;
5      scanf("%d %d", &a, &b);
6      if (a > b) {
7          printf("a > b");
8      }
9  }
```

Como utilizá-las



Como utilizá-las



- Em continuação ao **if()**, podemos utilizar **else** como uma condição final.
- Caso a condição do **if** não seja cumprida, o **else** será executado pelo programa, deixando implícito que "a" não é maior que "b".

```
1  #include <stdio.h>
2
3  int main() {
4      int a, b;
5      scanf("%d %d", &a, &b);
6      if (a > b) {
7          printf("a > b");
8      } else {
9          printf("a <= b");
10     }
11 }
12
```

Como utilizá-las



- Para não nos limitarmos em apenas duas condições (**if** e **else**), podemos utilizar as condicionais **else if** para adicionar mais condições em nossos programas.
- Exemplo: A imagem mostra uma condicional com 3 condições e a partir disso conseguimos descobrir se **(a>b)**, **(a=b)** ou **(a<b)**.

```
1  #include <stdio.h>
2
3  int main() {
4      int a, b;
5      scanf("%d %d", &a, &b);
6      if (a > b) {
7          printf("a > b");
8      } else if (a == b) {
9          printf("a = b");
10     } else {
11         printf("a < b");
12     }
13 }
14
```

Como utilizá-las

- Podemos utilizar junto com as condições os operadores lógicos E(&&) e OU(||), para deixar a condição mais restrita.

Exemplo:

```
1  #include <stdio.h>
2
3  int main(){
4      int a, b, c;
5      if((a>b) && (b>c)){
6          printf("a > b > c");
7      }
8      .
9      .
10     .
11     else{
12         printf("a < b < c");
13     }
14 }
```

Exercícios



1. Faça um programa em C que **recebe** a **média final** de um aluno e **retorne** **seu conceito**:

- de 0,0 a 4,9: D;
- de 5,0 a 6,9: C;
- de 7,0 a 8,9: B;
- de 9,0 a 10,0: A.

Switch() case:



- **Switch** é outro tipo de estrutura condicional. Ela executa vários testes de valores sobre o conteúdo de uma mesma variável.
- Podemos entendê-la como uma sucessão de vários **if** testados para uma mesma condicional.
- Por exemplo: um cardápio em que selecionamos o lanche que queremos, e o programa retornar o valor desse lanche.

Exemplo:

```
1  #include <stdio.h>
2
3  int main(){
4      int lanche, valor_1 = 10, valor_2 = 15,
5          valor_3 = 7;
6
7      printf("Selecione o lanche:");
8      scanf("%d", &lanche);
9      switch(lanche){
10         case 1:
11             printf("valor: %d", valor_1);
12             break;
13         case 2:
14             printf("valor: %d", valor_2);
15             break;
16         case 3:
17             printf("valor: %d", valor_3);
18             break;
19         .
20         .
21         default:
22             printf("não recebeu valor");
23     }
24 }
```



Exercícios



Exercícios



2. Escreva um programa em C que receba **3 valores** reais **X**, **Y** e **Z** e que verifique se esses valores podem ser os **comprimentos** dos lados de um **triângulo** e, neste caso, **retornar qual o tipo de triângulo formado**. Para que **X**, **Y** e **Z** formem um triângulo é necessário que a seguinte propriedade seja satisfeita: o comprimento de cada lado de um triângulo é menor do que a soma do comprimento dos outros dois lados. Caso seja possível formar o **triângulo**, o programa deve **identificar o tipo de triângulo** formado observando as seguintes definições:

- **Triângulo Equilátero:** os comprimentos dos 3 lados são iguais.
- **Triângulo Isósceles:** os comprimentos de 2 lados são iguais.
- **Triângulo Escaleno:** os comprimentos dos 3 lados são diferentes.

Exercícios



3. Faça um programa que **leia** um **número** entre **0** e **10**, e **escreva** este **número por extenso**. Use o comando **Switch/Case**.

Laços de Repetição



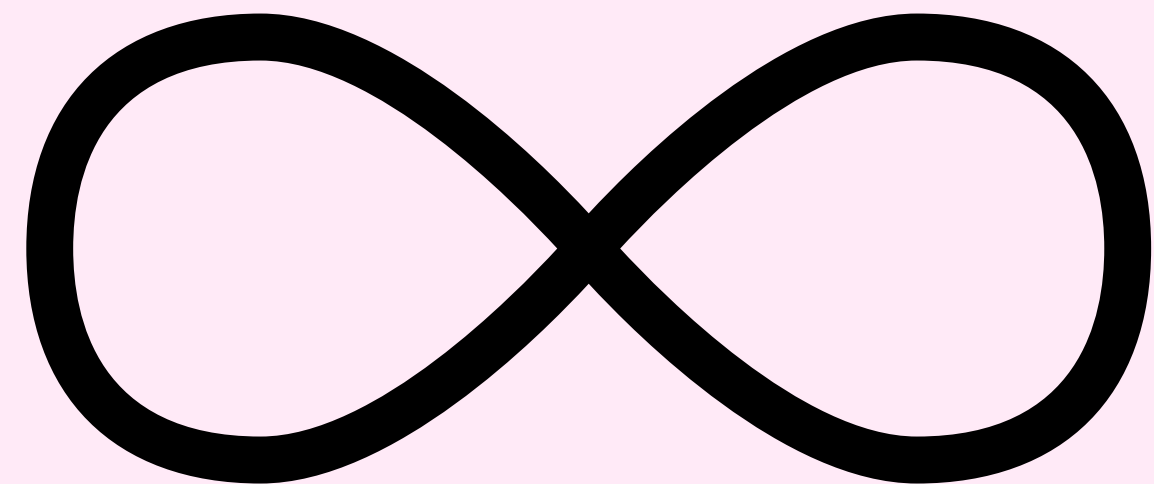
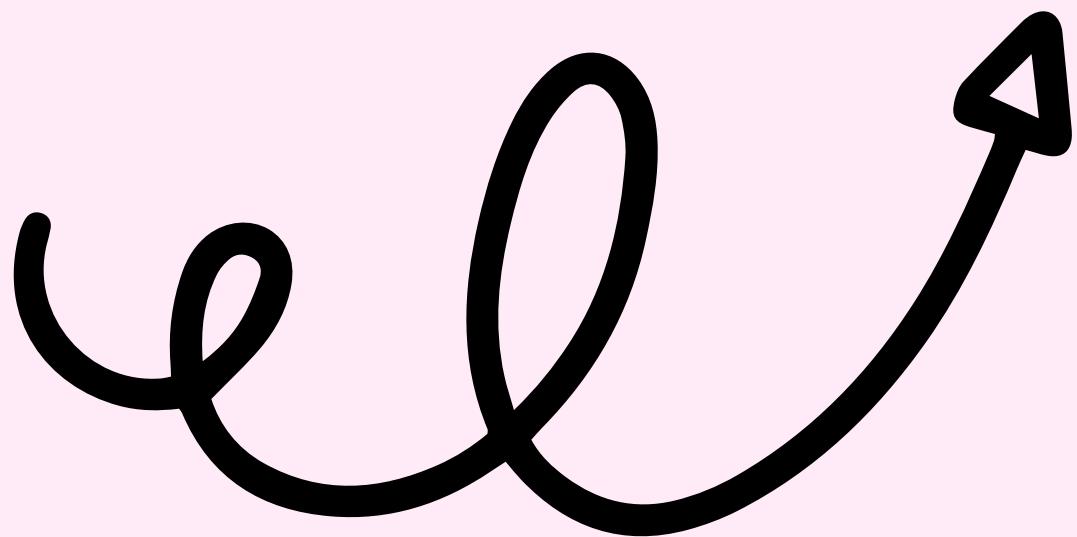
Laços de Repetição

- O que são?
- Como utilizá-las



O que são?

- Laços de repetição ou loops, são comandos que permitem a repetição de um código diversas vezes. Os utilizados em C são: `while()`; `do while()`; `for()`;
- Os laços de repetição são importantes, pois possibilitam a redução do código, percorrer um vetor verificando todos os elementos presentes e também limitar a repetição de acordo com uma condição.



Como utilizá-los



- O `while` é conhecido como enquanto. Ou seja, iremos repetir a linha de código dentro do `while`, enquanto satisfazermos a condição proposta. Ex: `while(x > 1);`
- No `while`, a verificação da condição ocorre desde o início do laço.

```
1  #include <stdio.h>
2
3  int main(){
4      int x = 0;
5      while(x < 10){
6          x += 1;
7          printf("%d", x);
8      }
9  }
```


Exercícios



1. Escreva um programa que **leia** valores **inteiros** e encontre o **maior** e o **menor** deles. Encerre o programa **se** o usuário digitar **zero (0)**;

Como utilizá-los



- O laço de repetição **do While**, conhecido como faça enquanto. Diferente do anterior, este irá executar as linhas de códigos para só depois verificar a condição fornecida.
- Dentro dos laços podemos indicar mais de uma condição, utilizando os operadores lógicos **E(&&)** **OU(||)**.

```
1  #include <stdio.h>
2
3  int main(){
4      int x = 0;
5      do{
6          x += 1;
7          printf("%d", x);
8      }while(x < 10);
9  }
```

Como utilizá-los



- O **For**, ou para. É um laço de repetição utilizado para incrementar o valor das variáveis a cada execução.
- Recebe três parâmetros, sendo esses: **variável**, **condição** e o **valor da incrementação**.
- No exemplo fizemos o mesmo código dos outros laços, mas de uma maneira mais fácil.

```
#include <stdio.h>

int main() {
    for (int x = 0; x < 10; x++) {
        printf("%d\n", x);
    }
}
```

Exercícios



Exercícios



2. Sendo $H = 1 + 1/2 + 1/3 + 1/4 + \dots + 1/N$. Faça um programa em C que **calcule** o valor de H . O valor de N é informado pelo usuário.

Exercícios



3. Escreva um programa que **leia 7 números** do usuário e **exiba** quantos números são **maiores do que 8**.

Projeto dia 02



- Criar um menu com um laço de repetição para o usuário escolher qual produto deseja comprar ou se deseja finalizar o programa.
- Tratar o troco para que não seja possível receber um valor menor do que o valor total.



MUITO OBRIGADO.

 @petinfouem

 pet@din.uem.br

 petinformaticauem

 discord.gg/5JaS4p4mWJ

