

Estruturas de Repetição

O que foi aprendido



QUAIS SÃO OS TIPOS ESTRUTURAS DE REPETIÇÃO, AFINAL?

01

Laço while: Repete um bloco de código enquanto uma condição é verdadeira.

02

Laço do-while: Executa um bloco de código pelo menos uma vez, e então repete enquanto uma condição é verdadeira.

03

Laço for: Utilizado para repetir uma operação um número específico de vezes.

while

```
while (condição) {
    // Corpo do loop
    // Código a ser repetido enquanto a condição for verdadeira
    }
}
```

do-while

```
do {
    do {
        // Corpo do loop
        // Código a ser repetido pelo menos uma vez
    } while (condição);
```

for

```
for (inicialização; condição; atualização) {
    // Corpo do loop
    // Código a ser repetido
  }
}
```

Cada um possui o seu próprio papel e necessidade.





QUAL É A APLICABILIDADE?

Os laços de repetição são aplicáveis sempre que precisamos executar uma ou mais instruções múltiplas vezes. Eles nos permitem automatizar tarefas repetitivas e processar conjuntos de dados de forma eficiente, economizando tempo e esforço.

EXEMPLO E POSSÍVEIS SOLUÇÕES:

Escreva um programa em C que solicite ao usuário que insira um número inteiro positivo e calcule o produto de todos os dígitos desse número.



01

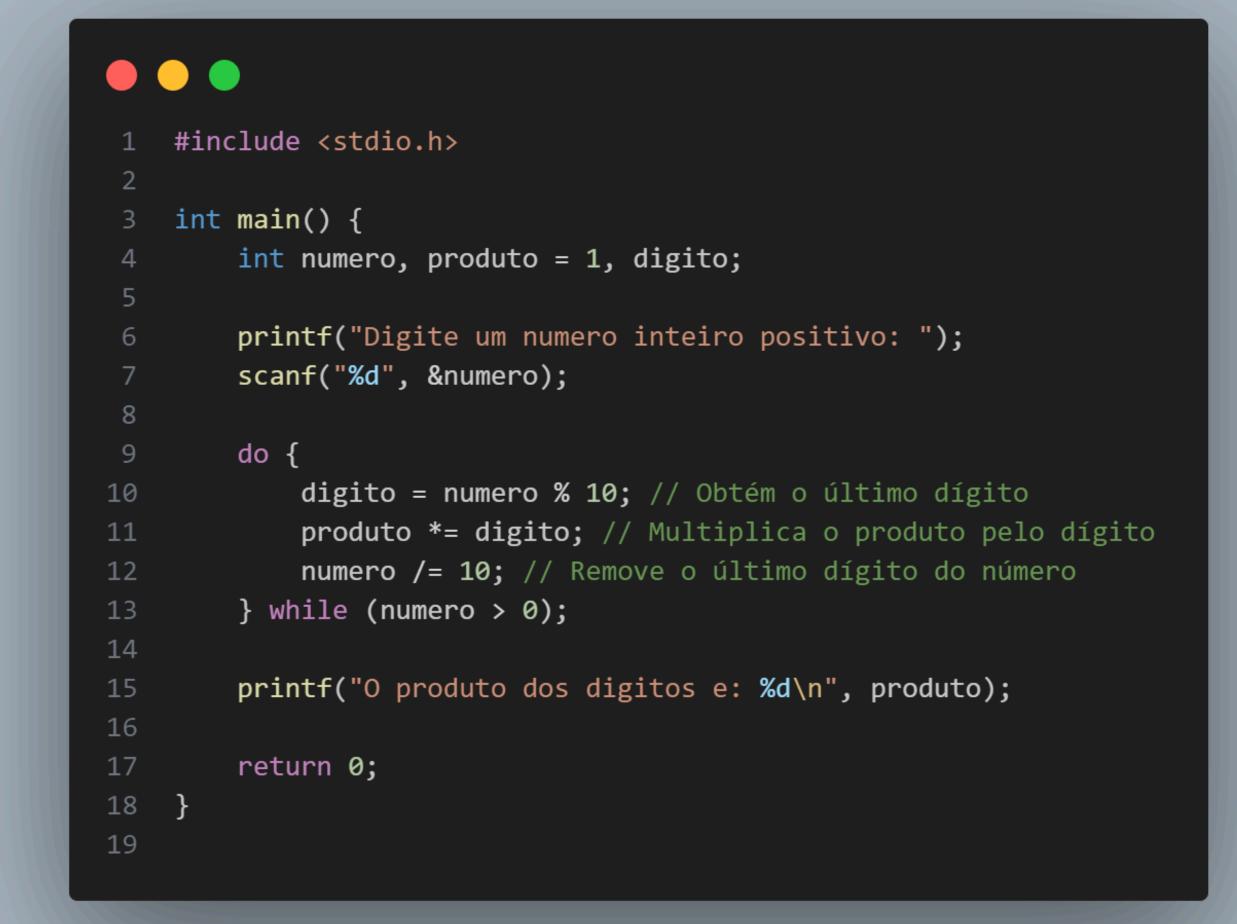
Resolução utilizando while:

```
#include <stdio.h>
    int main() {
        int numero, produto = 1, digito;
        printf("Digite um numero inteiro positivo: ");
 6
        scanf("%d", &numero);
 8
        while (numero > 0) {
 9
            digito = numero % 10; // Obtém o último dígito
10
11
            produto *= digito; // Multiplica o produto pelo dígito
            numero /= 10; // Remove o último dígito do número
12
13
14
15
        printf("0 produto dos digitos e: %d\n", produto);
16
        return 0;
17
18
```



02

Resolução utilizando do-while:





03

Resolução utilizando for:

```
#include <stdio.h>
    int main() {
        int numero, produto = 1, digito;
        printf("Digite um numero inteiro positivo: ");
        scanf("%d", &numero);
 8
        for (; numero > 0; numero /= 10) {
9
            digito = numero % 10; // Obtém o último dígito
10
            produto *= digito; // Multiplica o produto pelo dígito
11
12
13
        printf("0 produto dos digitos e: %d\n", produto);
14
15
16
        return 0;
17
```



NESTE CASO, QUAL A MELHOR OPÇÃO?

- while: É adequado quando não temos garantia de que o bloco de código será executado pelo menos uma vez, como quando precisamos verificar se o número é positivo antes de calcular o produto.
- do-while: Não traz vantagem aqui, pois não é necessário calcular o produto antes de verificar a validade do número.
- for: É a opção mais natural, pois sabemos quantas vezes o bloco de código deve ser executado (uma vez para cada dígito). É mais conciso e expressivo para percorrer os dígitos do número.



Desenvolva um programa em C que exiba a tabuada de multiplicação não apenas do número fornecido pelo usuário, mas também do número anterior e do número posterior. O programa deve solicitar ao usuário que insira um número e, em seguida, mostrar a tabuada desses três números (número anterior, número fornecido e número posterior) de 1 a 10.

- Ol Utilize laço for para calcular a tabuada do número anterior ao passado pelo usuário.
- Utilize laço while para calcular a tabuada do número passado pelo usuári pa
- Utilize laço do-while para calcular a tabuada do número posterior ao passado pelo usuário.



01

```
#include <stdio.h>

int main() {
   int numero, i;

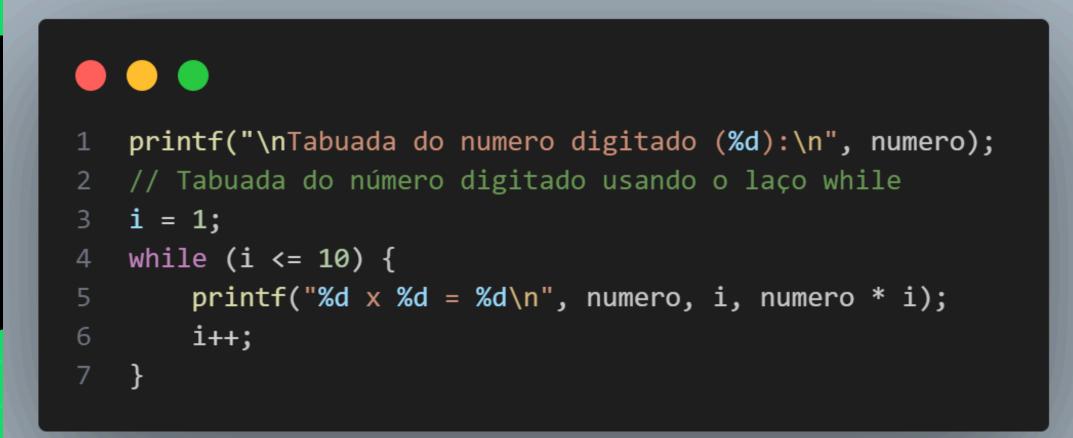
   // Solicita ao usuário que insira um número
   printf("Digite um numero: ");
   scanf("%d", &numero);

printf("\nTabuada do numero anterior (%d):\n", numero - 1);
   // Tabuada do número anterior usando o laço for
   for (i = 1; i <= 10; i++) {
        printf("%d x %d = %d\n", numero - 1, i, (numero - 1) * i);
   }
}</pre>
```

Primeiro deve-se declarar as variáveis, pedir o número da tabuada ao usuário e então usar do laço for para retornar a tabuada de seu anterior.



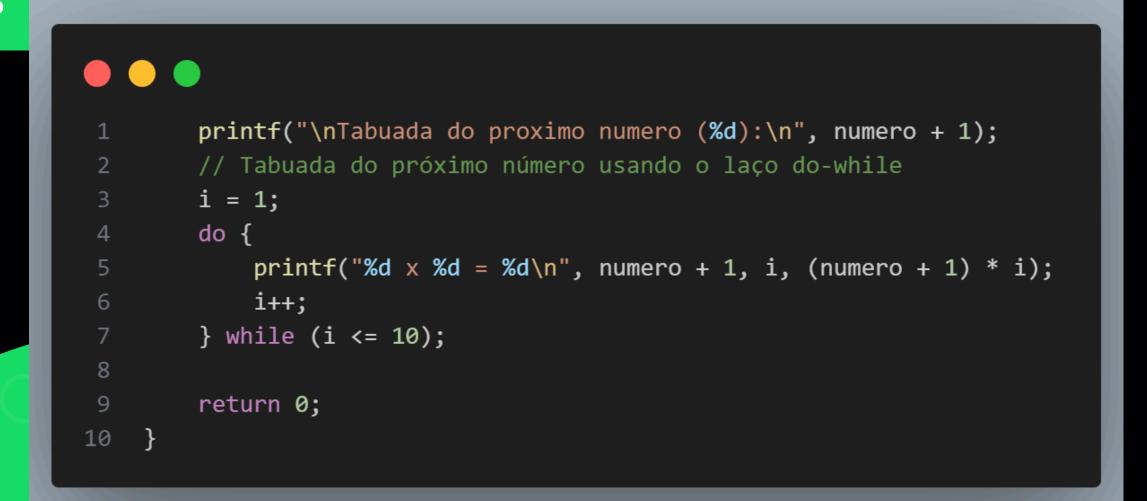
02



Agora use o laço while para retornar a tabuada do valor parrado pelo usuário.



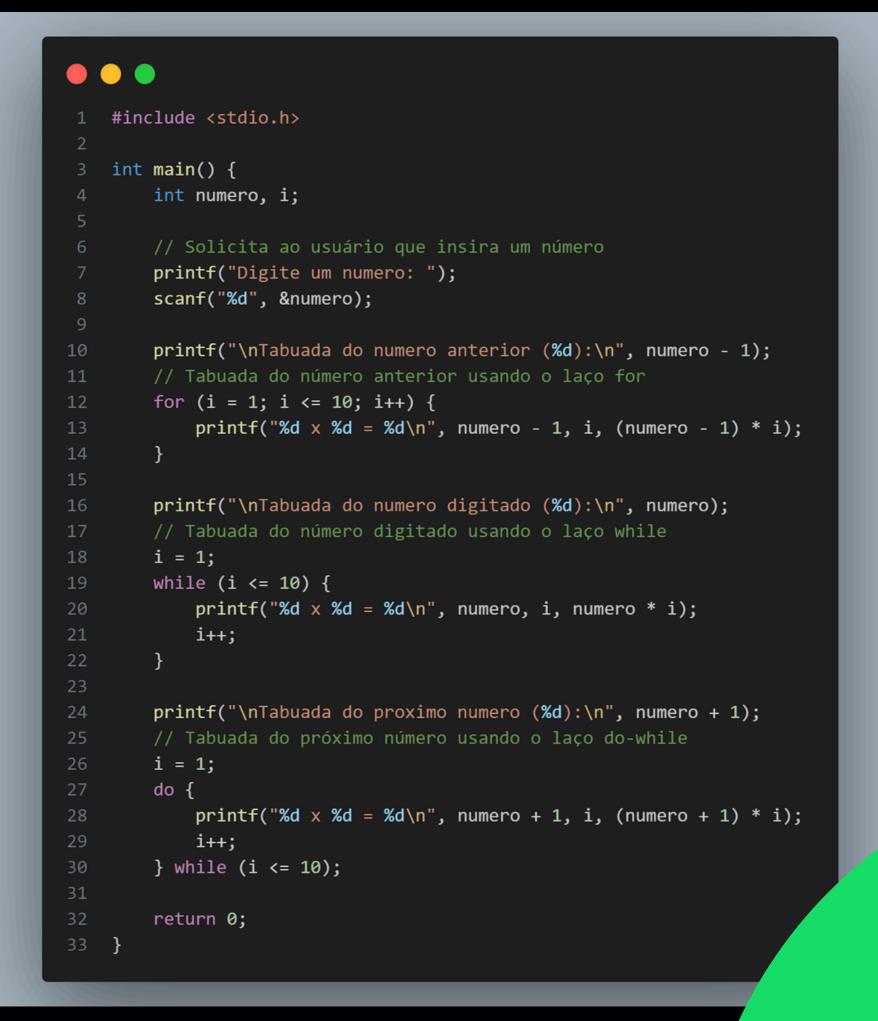
03

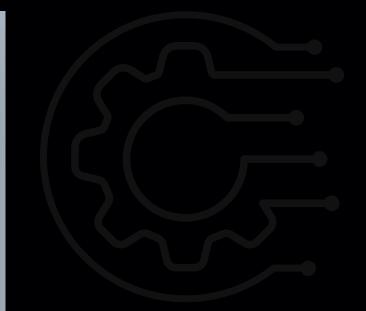


Agora faça o mesmo para o numero posterior utilizando do laço do-while.



CODIGO FINAL:







DESAFIO

Desenvolva um programa em C que solicite ao usuário um número inteiro positivo n e, em seguida, imprima um triângulo composto por asteriscos (*) de altura n. Por exemplo, se o usuário inserir 5, o programa deve exibir o seguinte padrão:



* *

