

Laboratório de Programação

Profa. Ms. Valéria Pinheiro

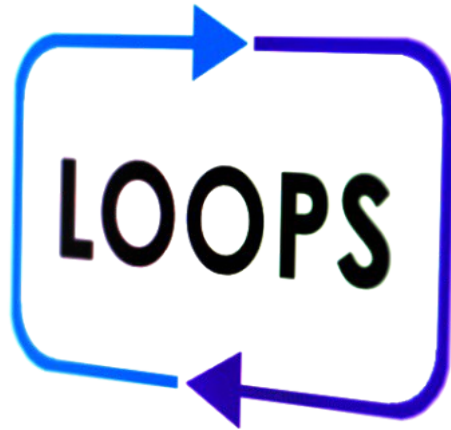


**UNIVERSIDADE
FEDERAL DO CEARÁ**
CAMPUS DE RUSSAS

Estrutura de repetição

Estrutura de repetição

- Toda linguagem moderna possui **comandos de repetição** que permitem que um conjunto de instruções seja executado até que certa *condição ocorra*.



Laços

```
for(inicialização ; condição ; passo){  
    comando1;  
    comando2;  
    ...  
    comandon;  
}
```

```
while(condição){  
    comando1;  
    comando2;  
    ...  
    comandon;  
}
```

```
do{  
    comando1;  
    comando2;  
    ...  
    comandon;  
}while(condição);
```

Laços – while / do- while

```
for(inicialização ; condição ; passo){  
    comando1;  
    comando2;  
    ...  
    comandon;  
}
```

```
while(condição){  
    comando1;  
    comando2;  
    ...  
    comandon;  
}
```

```
do{  
    comando1;  
    comando2;  
    ...  
    comandon;  
}while(condição);
```

Laços - while

```
for(inicialização ; condição ; passo){  
    comando1;  
    comando2;  
    ...  
    comandon;  
}
```

```
while(condição){  
    comando1;  
    comando2;  
    ...  
    comandon;  
}
```

```
do{  
    comando1;  
    comando2;  
    ...  
    comandon;  
}while(condição);
```

while

```
while(<condição>){  
    comando;  
}
```

```
while(<condição>){  
    comando1;  
    ....  
    comando_n;  
}
```

while – Soma até 0

```
soma = 0;
printf("numero a ser somado (0 para sair): ");
scanf("%d", &parcela);
while (parcela != 0) {
    soma += parcela;
    printf("numero a ser somado (0 para sair): ");
    scanf("%d", &parcela);
}
printf("Soma: %d\n", soma);
```


Laços – do - while

```
for(inicialização ; condição ; passo){  
    comando1;  
    comando2;  
    ...  
    comandon;  
}
```

```
while(condição){  
    comando1;  
    comando2;  
    ...  
    comandon;  
}
```

```
do{  
    comando1;  
    comando2;  
    ...  
    comandon;  
}while(condição);
```

do - while

```
do{  
    comando;  
} while(<condição>);
```

```
do{  
    comando_1....comando_n;  
} while(<condição>);
```

do - while – Soma até 0

```
soma = 0;
do {
    printf("numero a ser somado (0 para sair): ");
    scanf("%d", &parcela);
    soma += parcela;
} while (parcela != 0);
printf("Soma: %d\n", soma);
```

Resumindo do - while – while

while: A condição é verificada **antes** de cada execução do bloco de código. Se condição for verdadeira, o bloco de código é executado. Caso contrário, o bloco de código é ignorado e a execução do programa continua a partir da próxima instrução após **while**.

Por isso, o **while** também é conhecido como "*laço de pré-condição*".

do-while: O bloco de código é executado **pelo menos** uma vez antes que a condição seja verificada. Se a condição for verdadeira, o bloco de código executado novamente. Caso contrário, a execução do programa continua a partir da próxima instrução após o **do-while**.

Por isso, o **do-while** também é conhecido como "*laço de pós-condição*".

Exercícios

- 1- Dada uma sequência de números inteiros diferente de zero, terminada por um zero, imprima o quadrado de cada número da sequência.
- 2 - Sabe-se que uma determinada espécie de Bactéria divide-se a cada segundo. Faça um programa em C que calcule o número de bactérias obtidas em n segundos. O número de segundos deve ser passado pelo usuário.
- 3 - Faça um programa em C que receba um número n, passado pelo usuário, e retorne o enésimo termo da sequência: **1 3 7 15 31 63 127...**
- 4 - A conversão de graus Fahrenheit para Celsius é obtida por $T_c = (T_f - 32) * 5/9$, em que T_c é a temperatura em graus Celsius e T_f em Fahrenheit. Faça um programa C que calcule e que imprima uma tabela de graus Fahrenheit e graus Celsius, cujos graus variem de 50 a 65, de 1 em 1.
- 5 - Faça um programa que leia dois conjuntos de dois valores, o primeiro representando o número do aluno e o segundo representando a sua altura em centímetros. Encontre o aluno mais alto e o mais baixo. Mostre o número do aluno mais alto, e o número do aluno mais baixo, juntamente com suas alturas.