

PROGRAMAÇÃO IMPERATIVA

Prof. Dr. Edson J. R. Justino
Prof. Alcides Calsavara

Semana 4

Pontifícia Universidade Católica do Paraná - PUCPR

CONTROLE DE FLUXO

- **Estrutura de Controle de Fluxo**

- for
- do-while
- break
- continue

- **Estrutura Condicional Múltipla**

- switch

Comando *for*

Sintaxe:

```
for (expr-inicial ; expr-condição ; expr-incremento)  
{  
    comandos;  
}
```

Sendo:

expr-inicial - executada apenas uma vez, no início do comando for. Pode ser uma série de expressões separadas por vírgulas.

expr-condição – expressão lógica avaliada a cada iteração, antes da execução do bloco de comandos, para decidir se a repetição continua: no caso de ser avaliado “falso”, indica o fim da repetição.

expr-incremento – executada a cada iteração, após a execução do bloco de comandos. Pode ser uma série de expressões separadas por vírgulas.

Exemplo do comando *for*

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

Exemplo do comando *for*

```
for (int i = 0, j = 10; i != j; i++, j--)  
{  
    printf("%d %d\n", i, j);  
}
```

Exemplo do comando *for*

```
for (int i = 0, j = 10; i != j; )  
{  
    printf("%d %d\n", i, j);  
    i++;  
    j--;  
}
```

Exemplo do comando *for*

```
for (int i = 0, j = 10; ; )  
{  
    printf("%d %d\n", i, j);  
    i++;  
    j--;  
}
```

Exemplo do comando *for*

```
int i = 0, j = 10;

for ( ; ; )
{
    printf("%d %d\n", i, j);
    i++;
    j--;
}
```


Exercício

Converta o programa em Python para um programa em C, usando o comando *for*

```
# Python
PI = 3.1416
R = 0
while R <= 6:
    VOLUME = 4/3 * PI * (R**3)
    print(R, VOLUME)
    R = R + 2
```

Comando *do-while*

Sintaxe:

```
do  
{  
    comandos;  
}  
while (condição);
```

A execução ocorre do seguinte modo:

- 1) O bloco de comandos é executado
- 2) A expressão (condição) é avaliada
- 3) Caso o valor seja “verdadeiro”, o fluxo de execução volta para o item 1
- 4) Caso o valor seja “falso”, o “loop” termina e o processamento segue a partir do primeiro comando após o do-while

Exemplo do comando *do-while*

```
int i = 1;
do
{
    printf("i = %d\n", i);
    i++;
}
while(i < 10);
```

Exemplo do comando *do-while*

```
// código usando do-while:  
char letra = 'a';  
do  
{  
    putchar(letra);  
    letra++;  
}  
while (letra <= 'z');
```

```
// código usando while:  
char letra = 'a';  
while (letra <= 'z')  
{  
    putchar(letra);  
    letra++;  
}
```

Exemplo do comando *do-while*

```
int k; // um valor entre 10 e 20, inclusive

do
{
    printf("Digite um valor inteiro entre 10 e 20: ");
    scanf("%d", &k);
}
while (k < 10 || k > 20);

printf("Valor digitado: %d\n", k);
```

Exercício

Converta o programa em Python para um programa em C, usando *do-while*.

```
# Python
PI = 3.1416
R = 0
while R <= 6:
    VOLUME = 4/3 * PI * (R**3)
    print(R, VOLUME)
    R = R + 2
```

Comandos *break* e *continue*

Pode ser conveniente fazer o controle de execução de uma repetição independentemente do teste lógico efetuado.

- O *break*, quando utilizado dentro do bloco de comandos de um “loop”, (*for*, *while* ou *do-while*) faz com que o “loop” seja imediatamente interrompido, transferindo o fluxo de execução para o próximo comando após o “loop”.
- O *continue* desvia o fluxo de execução para a próxima iteração, ou seja, para o teste de parada do “loop”.

Exemplo do comando *break*

```
int i = 0;
do
{
    i++;
    if ( (i % 4) == 0 )
        break;
    printf("i = %d\n", i);
}
while (i < 10);
```


Exemplo do comando *continue*

```
int i = 0;
do
{
    i++;
    if ( (i % 2) == 0 )
        continue;
    printf("i = %d\n", i);
}
while (i < 10);
```

Exercício

Converte o programa em Python para um programa em C, usando *while* e *break*.

```
# Python
l=10
while l > 0 :
    l = l - 0.25
    print (l)
    if (l**2) + 1 ≥ 1.45 :
        l = l + 0,20;
    else
        break
```

Comando switch

Permite tomada de múltiplas decisões baseadas em uma expressão

Sintaxe:

```
switch (expressão)
{
    case 1:
        comandos;
        break;
    case 2:
        comandos;
        break;
    default:
        comandos;
        break;
}
```

Exemplo do comando *switch*

```
int k;  
scanf("%d", &k);  
switch ( k )  
{  
    case 1: puts("UM"); break;  
    case 2: puts("DOIS"); break;  
    case 3: puts("TRÊS"); break;  
    default: puts("DESCONHECIDO");  
}
```

Exemplo do comando *switch*

```
int vogais = 0, consoantes = 0, hifens = 0;
```

switch_char.c

```
char palavra[80];
```

```
printf("Digite um palavra: ");
```

```
scanf("%s", palavra);
```

```
for (int i = 0; palavra[i] != '\0'; i++)
```

```
switch ( palavra[i] )
```

```
{
```

```
    case 'a':
```

```
    case 'e':
```

```
    case 'i':
```

```
    case 'o':
```

```
    case 'u': vogais++; break;
```

```
    case '-': hifens++; break;
```

```
    default: consoantes++;
```

```
}
```

```
printf("vogais: %d, consoantes: %d, hifens: %d\n",  
       vogais, consoantes, hifens);
```

Exercício

Elabore um programa em C para simular uma calculadora com as quatro operações básicas (adição, subtração, multiplicação e divisão), usando a estrutura *switch*. O programa deve receber, como entrada, dois operandos (*double*) e um operador (*char*). Como saída, o programa deve exibir o resultado da aplicação do operador sobre os dois operandos.

Exercícios complementares

1) Sendo $H = 1 + 1/2 + 1/3 + 1/4 + \dots + 1/N$, elaborar um programa. em C para gerar o número H. O valor de N deverá ser lido. Use a estrutura **for** para somar a sequência de termos.

2) Escreva um programa em C para calcular o fatorial de um número N digitado. Use a estrutura **for** para gerar a sequência de termos.

3) Fazer um programa em C para calcular o valor da série S abaixo. O valor de N deve ser digitado. Use a estrutura **do-while** para somar a sequência de termos.

$$S = 1/N + 2/N-1 + 3/N-2 + \dots + N-1/2 + N/1$$

4) Elabore um programa em C para imprimir os 30 primeiros números naturais pares. Use a estrutura **for** para gerar a sequência de termos.