

SSC0101 - ICC1 – Teórica

Introdução à Ciência da Computação I

**Estrutura de Programas:
Repetição – Parte II**

Prof. Vanderlei Bonato: vbonato@icmc.usp.br

Prof. Claudio Fabiano Motta Toledo: claudio@icmc.usp.br

Sumário

- Estrutura de Repetição
- Estrutura de Repetição em Algoritmos
 - PARA
 - ENQUANTO
 - REPITA
- Estrutura de Repetição em Linguagem C
 - for
 - while
 - do-while
- Comando de controle de laço
 - break
 - continue
 - goto

Comando for

```
for (inicialização; teste; incremento)
    instrução;
```

```
for (inicialização; teste; incremento)
{
    instrução_1;
    instrução_2;
    ...
    instrução_n;
}
```

Comando for

Exemplo:

//imprime números de 0 a 9

```
int main()
```

```
{
```

```
    int conta;
```

```
    for(conta=0; conta<10; conta++)
```

```
        //para mais de uma instrução no corpo do for deve-se utilizar chaves
```

```
        {
```

```
            printf("conta=%d\t",conta);
```

```
            printf("conta=%d\n",-(conta-9));
```

```
        }
```

```
        system("PAUSE");
```

```
}
```

Comando for

Exemplo :

```
//imprime o valor ASCII
```

```
int main()
```

```
{
```

```
    char ch;
```

```
    for(ch='0'; ch<='9'; ch++)
```

```
        printf("O valor ASCII de %c e' %d.\n", ch, ch);
```

```
    system("PAUSE");
```

```
}
```

Comando for

Exemplos:

```
for(i = 1; i<=10; i++)  
    printf("%d ", i);
```

⇒ 1 2 3 4 5 6 7 8 9 10

```
for(i = 1; i<=10; i=i+1)  
    printf("%d ", i);
```

⇒ 1 2 3 4 5 6 7 8 9 10

```
for(i = 10; i>=1; i--)  
    printf("%d ", i);
```

⇒ 10 9 8 7 6 5 4 3 2 1

Comando for

Exemplos:

```
for(i = 1; i<=10; i=i+2)
```

```
    printf("%d ", i);
```

⇒ 1 3 5 7 9

```
    printf("\n\ni=");
```

```
for(i = 10; i>=1; i=i-3)
```

```
    printf("%d ", i);
```

⇒ 10 7 4 1

```
    printf("\n\ni=");
```

```
for(i = -10; i<=10; i=i+5)
```

```
    printf("%d ", i);
```

⇒ -10 -5 0 5 10

Comando for

Exemplos:

```
//Usando contador float
```

```
for(k = 0; k<=1; k += 0.1)
```

```
    printf("%3.1f ", k);
```

⇒ 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

```
//Usando contador char
```

```
for(letra = 'A'; letra<='Z'; letra++)
```

```
    printf("%c ", letra);
```

⇒ A B CX Y Z

```
for(letra = 'z'; letra>='a'; letra--)
```

```
    putchar(letra);
```

⇒ z y xc b a

Comando for

//lê um caractere e imprime o seguinte

```
int main()
{
    char ch;
    for(ch=getch(); ch!='X'; ch=getch())
        printf("%c", ch+1);
}
```

Como sair do loop?

Comando for

- Qualquer uma das três partes do laço “for” pode ser omitida
- Todavia, o ponto-e-vírgula deve estar presente.
- Se as expressões de inicialização e de incremento forem omitidas, elas apenas serão desconsideradas
- Porém, se a expressão de teste for omitida, a mesma é considerada permanentemente verdadeira
 - O que acontece nessa situação?

Comando for

Exemplos: considere `int i, int sum = 0`. Os trechos abaixo são equivalentes

<pre>for(i=1; i<=10; i++) sum += i;</pre>	<pre>i=1 for(; i<= 10;i++) sum += i;</pre>	<pre>i=1; for(; i<= 10;) sum += i++;</pre>
--	--	--

O que ocorre com esse exemplo?

```
i=1; sum=0;
```

```
for(; ; )
```

```
  sum += i++;
```

Comando for

- Qualquer expressão do comando “for” pode conter várias instruções separadas por vírgula.
- O operador vírgula usado no “for” permite múltiplas inicializações e múltiplos processamento de índices.
- Um par de expressões separadas por vírgulas é avaliado da esquerda para a direita.

Comando for

Exemplos: Considere int i, sum=0. Os trechos abaixo são equivalentes

```
for(i=1; i<=10; i++)  
    sum += i;
```

```
for(sum=0, i=1; i<= 10; i++ )  
    sum += i;
```

```
for(sum=0, i=1; i<= 10; sum+=i, ++i )  
    ;
```

O trecho abaixo NÃO é equivalente

```
for(sum=0, i=1; i<= 10; ++i, sum+=i )  
    ;
```

Comando for

Exemplo:

```
//Usando mais de um contador
int main()
{
    int i,j;
    printf("\n\n(i,j)=");
    for(i = 1, j=1; i<=5, j<=5; i++, j++)
        printf("(%d,%d) ", i, j);
    system("PAUSE");
    return 0;
}
```

(i,j)=(1,1) (2,2) (3,3) (4,4) (5,5)

Comando for

Exemplo:

//imprime os números de 0 a 98 em incremento de 2

```
int main()
{
    int x,y;
    for(x=0, y=0; x+y < 100; x=x+1, y=y+1)
        printf("%d ",x+y);
    system("PAUSE");
    return 0;
}
```

0 2 4 6 8 10 1294 96 98

Comando for - aninhado

- Pode existir n comandos for aninhados

```
for (inicialização1; teste1; incremento1)
```

```
    for (inicialização2; teste2; incremento2)
```

```
        for (inicialização3; teste3; incremento3)
```

```
            ....
```

```
                for (inicializaçãoN; testeN; incrementoN)
```

```
                    instrução;
```


Comando while

```
while (expressão de teste)  
    instrução;
```

```
while (expressão de teste)  
{  
    instrução_1;  
    instrução_2;  
    ...  
    instrução_n;  
}
```

Comando while

- O “while” pode substituir o “for” do seguinte modo:

Inicialização da variável de teste

```
while(teste)
```

```
{
```

```
    Incremento da variável de teste;
```

```
    ....
```

```
}
```

Comando while

- Há uma equivalência entre os comandos while e for.

```
for (expr1; expr2; expr3)
    instrução1;
Instrução2;
```

```
expr1;
while(expr2){
    instrução1;
    expr3;
}
Instrução2;
```

- Laço for é equivalente ao while, considerando que expr2 ocorre e que não há um comando **continue** no corpo de um laço for.

Comando while

- Exemplo 1

```
int main()
{
    int conta=0;
    int total=0;
    while(conta<10)
    {
        total+=conta;
        printf("conta=%d, total=%d\n", conta, total);
        conta++;
    }
    system("pause");
}
```

Comando while

- Exemplo 2

```
int main()
{
    int conta=0;
    printf("Digite uma frase:\n");
    //13 é o valor do caractere "enter" ou "cr"(carriage return)
    while(getche()!=13)
    {
        conta++;
    }
    printf("A frase possui %d caracteres",conta);

    system("pause");
}
```

Veja a tabela ASCII: <http://www.asciitable.com/>

Comando while

- Exemplo 3: Há algum problema nos trechos de código abaixo?

```
while (++i<LIMIT) do{ //trecho de código 1
    j=2*i+3;
    printf(“%d\n”,j);
}
```

```
printf(“Input na integer: “); //trecho de código 2
scanf(“%d”, &n);
while (--n)
    ..... /* faça alguma coisa */
```

```
while ((c=getchar()) == ‘ ‘) //trecho de código 3
    ;
```

Comando while - aninhado

```
while (expressão de teste1)
  while(expressão de teste2)
    while(expressão de teste3)
      ...
      while(expressão de testeN)
        instrução;
```

Exercício I

- Faça um programa que receba números inteiros maiores que 1. A cada valor recebido, exiba um mensagem dizendo se o número é primo ou não.

Exercício II

- Faça um programa que receba valores inteiros positivos , calcule e mostre:
 - A soma dos números digitados.
 - A quantidade de números digitados.
 - A média dos números digitados.
 - O maior número digitado.
 - O menor número digitado.
 - A média dos números pares.
 - A percentagem dos números ímpares entre todos os números digitados.

Referências

Ascencio AFG, Campos EAV. Fundamentos de programação de computadores. São Paulo : Pearson Prentice Hall, 2006. 385 p.

Kelley, A.; Pohl, I., *A Book on C: programming in C*. 4ª Edição. Massachusetts: Pearson, 2010, 726p.

Kernighan, B.W.; Ritchie, D.M. *C, A Linguagem de Programação: padrão ANSI*. 2ª Edição. Rio de Janeiro: Campus, 1989, 290p.

FIM Aula 8
