

Introdução à Ciência da Computação

Introdução à Linguagem C: Parte II

Prof. Ricardo J. G. B. Campello

Sumário

- Seleção Simples
- Seleção Composta
- Seleção Encadeada
- Seleção de Múltipla Escolha
- Repetição com Teste no Início
- Repetição com Teste no Final
- Repetição com Variável de Controle
- Exercícios

Seleção Simples

- Sintaxe:

```
if ( condição )  
{  
    instruções;    /* comandos do bloco verdade */  
}
```

- **if** é uma palavra reservada (palavra-chave)
- **condição** é uma variável ou expressão com valor lógico
 - verdadeiro (valor não nulo) ou falso (valor nulo)
- **instruções** é qualquer conjunto de comandos válidos em C
- Se *instruções* possui apenas uma instrução, os delimitadores do bloco verdade { e } podem ser omitidos

Seleção Simples

- Exemplo:

```
#include <stdio.h>  
void main(void){  
    int A, B, C;  
    printf("Entre com 3 números inteiros:\n");  
    scanf("%d%d%d", &A, &B, &C);  
    if (A + B < C)  
        printf("%d é maior que %d", C, A+B);  
}
```

Seleção Composta

- Sintaxe:

```
if ( condição )
{
    instruções 1;    /* comandos do bloco verdade */
}
else
{
    instruções 2;    /* comandos do bloco falsidade */
}
```

- Se *condição* for verdadeira, *instruções 1* serão executadas, caso contrário, *instruções 2* serão executadas

Seleção Composta

- Exemplo:

```
#include <stdio.h>
void main(void){
    int A, B, C;
    printf("Entre com 3 números inteiros:\n");
    scanf("%d%d%d", &A, &B, &C);
    if (A + B < C) printf("%d é maior que %d", C, A+B);
    else printf("%d é maior ou igual a %d", A+B, C);
}
```

Seleção Composta

- Operador "?"

- Sintaxe: (Exp1 ? Exp2 : Exp3)
 - Exp1, Exp2 e Exp3 são constantes, variáveis ou expressões
- Valor resultante:
 - **Exp2**: se **Exp1** for verdadeira (valor não nulo)
 - **Exp3**: se **Exp1** for falsa (valor nulo)
- Exemplo:

```
#include <stdio.h>
void main(void){
    int idade, menor;
    printf("Entre com a idade:\n");
    scanf("%d", &idade);
    menor = ( idade < 18 ? 1 : 0);
}
```

Seleção Encadeada

- C permite o encadeamento de estruturas de seleção
 - O encadeamento pode ser homogêneo ou heterogêneo
- Analisem o código a seguir verificando:
 - O encadeamento dos **if**, **if-else**, **if-else-if**
 - O uso ou não de blocos de instruções { ... }

Seleção Encadeada

```
#include<stdio.h>
void main(void){
    int A, B;
    printf("Entre com dois inteiros, A e B:\n");
    scanf("%d%d", &A, &B);
    if (A > B){
        printf("A é maior que B\n");
        if (A>10*B) printf("Na verdade, A é mais que 10x maior que B\n");
        else
            if (A<=2*B) printf("Porém, A não é mais que o dobro de B\n");
    }
    else
        if (A == B) printf("A e B são iguais");
        else printf("A é menor que B");
}
```

Seleção de Múltipla Escolha

Sintaxe:

```
switch ( seletor ) {
    case valor_1 :
        instruções 1;
        break;
    case valor_2 :
        instruções 2;
        break;
    ...
    case valor_n :
        instruções n;
        break;
    default:
        instruções;
}
```

- **seletor** : qualquer variável ou expressão cujo valor seja um caractere ou um número inteiro (possivelmente expressão com valor lógico)
- **valor_i** : valor inteiro ou caractere
 - i-ésimo conjunto de instruções será executado se *valor_i* = valor do *seletor*
- **break** : desvia o fluxo de execução para a instrução seguinte ao comando switch
 - não é obrigatório
 - porém, se não for inserido ao final do i-ésimo conjunto de instruções, quando este conjunto for executado o (i+1)-ésimo conjunto também será executado (até que se encontre um **break**)

Seleção de Múltipla Escolha

```
#include<stdio.h>
void main(void){
    char menu;
    printf("Entre com a Estação do Ano (p, v, o ou i) e ENTER:\n");
    scanf("%c", &menu);
    switch ( menu ) {
        case 'p':
            printf("Setembro a Dezembro");
            break;
        case 'v':
            printf("Dezembro a Março");
            break;
        case 'o':
            printf("Março a Junho");
            break;
        case 'i':
            printf("Junho a Setembro");
            break;
        default:
            printf("Entrada Inválida!");
    }
}
```

Repetição com Teste no Início

- Comando **while**
 - Equivale ao **enquanto** do pseudo-código
- Verifica antes de cada execução se é permitido executar o trecho do programa dentro do laço:
 - O trecho é repetido enquanto uma condição for verdadeira
 - Quando a condição se tornar falsa, a repetição é interrompida
- Se já na primeira vez a condição é falsa, as instruções internas não serão executadas

Repetição com Teste no Início

- Sintaxe:

```
while ( condição )  
{  
    instruções;  
}
```

- **condição** é uma variável ou expressão com valor lógico
 - verdadeiro (valor não nulo) ou falso (valor nulo)
- **instruções** é qualquer conjunto de comandos válidos em C
- Se *instruções* possui apenas uma instrução, os delimitadores do bloco verdade { e } podem ser omitidos

Repetição com Teste no Início

- Exemplos:

```
#include<stdio.h>  
void main(void){  
    while (1) printf("Devo Estudar Mais!");  
}
```

```
#include<stdio.h>  
void main(void){  
    int contador = 0;  
    while (contador < 10){  
        printf("Devo Estudar Mais!\n");  
        contador++;  
    }  
}
```

Repetição com Teste no Final

- Comando **do while**

- Repete um trecho do programa enquanto uma determinada condição for verdadeira
- Como o teste é feito no final do laço, os comandos internos serão executados pelo menos uma vez !

Repetição com Teste no Final

- Sintaxe:

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

- **condição** é uma variável ou expressão com valor lógico
 - verdadeiro (valor não nulo) ou falso (valor nulo)
- **instruções** é um (conjunto de) comando(s) válido(s) em C

Repetição com Teste no Final

Exemplo:

```
#include<stdio.h>
void main(void){
    int no = 0;
    do {
        printf("Entre com um inteiro entre 1 e 10:\n");
        scanf("%d", &no);
    } while (no<1 || no>10);
    printf("A opção escolhida foi: %d", no);
}
```

Repetição com Var. de Controle

Comando **for**

- Utiliza uma ou mais variáveis de controle
- Pode fazer o papel do **para** em pseudo-código
 - repetição de laço por um número pré-estabelecido de vezes
- Porém, em C este comando é mais flexível que isso
 - pode também, por exemplo, fazer o mesmo papel de **while**
 - repetição por um número de vezes não pré-estabelecido

Repetição com Var. de Controle

Sintaxe:

```
for(inicialização ; condição ; incremento) { instruções ; }
```

- inicialização**: atribuição de valor(es) à(s) variável(is) de controle
- condição**: variável ou expressão com valor lógico
 - laço será repetido enquanto esta condição for **verdadeira** (não nula)
- incremento**: modificação do(s) valor(es) da(s) var(s). de controle
- inicialização, condição e incremento são **todos opcionais** !
- Se *instruções* possui apenas uma instrução, os delimitadores do bloco verdade { e } podem ser omitidos

Repetição com Var. de Controle

Repetição de laço por número pré-estabelecido de vezes

- Exemplo 1 (passo unitário positivo):

```
#include<stdio.h>
void main(void)
{
    int cont;
    for(cont = 1 ; cont <= 10; cont++)
        printf("Devo Estudar Mais!\n");
}
```

- Equivale a: **para** cont **de** 1 **até** 10 **passo** 1 **faça**

Repetição com Var. de Controle

- Repetição de laço por número pré-estabelecido de vezes
 - Exemplo 2 (passo negativo, não unitário):

```
#include<stdio.h>
void main(void)
{
    int cont;
    for(cont = 100 ; cont >= 0; cont--=5)
        printf("%d\n", cont);
}
```

- Equivale a: **para** cont **de** 100 **até** 0 **passo** -5 **faça**

Repetição com Var. de Controle

- Repetição por número não pré-estabelecido de vezes
 - Exemplo:

```
#include<stdio.h>
void main(void)
{
    int no = 0;
    for( ; no<1 || no>10; ){
        printf("Entre com um inteiro entre 1 e 10:\n");
        scanf("%d", &no);
    }
}
```

- Equivale a: **enquanto** (no < 1 **ou** no > 10) **faça**

Repetição com Var. de Controle

- Comando **for** permite múltiplas variáveis de controle...
 - Exemplo:

```
#include<stdio.h>
void main(void)
{
    int x, y;
    for(x=0,y=0 ; y*y <= x; x+=100,y+=1);
}
```

- No exemplo acima o laço (sem instruções) simplesmente incrementa x com passo 100 e y com passo 1 até que y^2 passe a ser maior que x (exemplo meramente ilustrativo)

Comandos **break** e **continue**

- O comando **break** interrompe imediatamente qualquer comando de repetição, independente das condições de parada do comando terem ou não sido satisfeitas
- O comando **continue** força um desvio do fluxo de execução do programa para o início do laço, "pulando" instruções internas ao laço subsequentes ao comando
- Exemplo...

Comandos **break** e **continue**

Exemplo (laço "infinito"):

```
#include<stdio.h>
void main(void){
    int x;
    for( ; ; ){
        printf("Entre com opção 1 a 5, ou 0 para sair:\n");
        scanf("%d", &x);
        if (x>=1 && x<=5){
            printf("Opção Escolhida: %d\n", x);
            continue;
        }
        if (x==0) break;
        else printf("Opção Inválida\n");
    }
}
```

Exercícios

- ◆ Sugestão de metodologia de estudos ao resolver os exercícios a seguir:
 - Resolva o problema em pseudo-código, para se preocupar inicialmente apenas com a lógica de solução, e não com detalhes de sintaxe de linguagem de programação
 - Faça um teste de mesa para se certificar que seu pseudo-código está correto
 - Transcreva o pseudo-código em linguagem C
 - Compile e rode o código nos laboratórios da graduação ou em casa para se certificar que o programa está correto

Exercícios

- Faça um programa em C que leia do teclado três valores reais (A, B e C) e escreva na tela se esses valores podem ser os comprimentos dos lados de um triângulo ou não. Caso possam, verificar e escrever no dispositivo de saída se o triângulo é equilátero, isósceles ou escaleno

Exercícios

- Exercício: escreva um programa C que leia o código de um determinado produto e mostre a sua classificação
 - Utilize seleção de múltipla escolha (comando **switch**)
 - Utilize a seguinte tabela como referência:

Código:	Classificação:
1	Alimento não perecível
2, 3 ou 4	Alimento perecível
5 ou 6	Vestuário
7	Higiene pessoal
8 a 15	Limpeza e utensílios domésticos
Qualquer outro código	Inválido

Exercícios

- Faça um programa em C, utilizando o comando **while**, que calcule a média aritmética de um conjunto de números inteiros pares fornecidos pelo usuário:
 - O valor de parada será 0
 - O usuário poderá entrar com números ímpares; porém, esses não devem ser considerados nos cálculos
 - O programa deverá retornar como saída (resposta) a quantidade de números pares digitados e a média dos mesmos
- Refaça o programa acima utilizando o comando **repita**

Exercícios

- Utilize o comando **for** e escreva um programa, em C, que calcule e escreva o valor do número S:

$$S = \frac{1}{50} + \frac{3}{49} + \frac{5}{48} + \frac{7}{47} + \dots + \frac{99}{1}$$

Exercícios

- Elabore um programa C que efetue a soma de todos os números ímpares que são múltiplos de 3 e que se encontram no intervalo de 1 a 500

Exercícios Adicionais

- ◆ Traduza para C os pseudo-códigos dos exemplos de estruturas de seleção e repetição (aula de estruturas de controle)
 - Compile e execute os programas em laboratório ou no seu computador pessoal para se certificar que funcionam!
 - Pesquise as bibliotecas C caso alguma função (por exemplo matemática) seja necessária