Algoritmos e Lógica de Programação

Introdução a Linguagem C

Prof. Rhafael Freitas da Costa

Conteúdo

- Componentes da Linguagem C;
- Tipos de dados;
- Declarações de variáveis e constantes;
- Atribuição de conteúdo;
- Operadores aritméticos, lógicos e relacionais;
- Comandos de entrada e saída;
- Estrutura de um programa em C.

Introdução

- Este material foi elaborado com base em uma das obras da bibliografia recomendada da disciplina:
  - EDELWEISS, Nina. Algoritmos e programação com exemplos em Pascal e C. Porto Alegre. Bookman 2014.
- Recurso on-line (Livros didáticos UFRGS 23);
- Recomenda-se fortemente o estudo de outros materiais, tais como:
  - http://linguagemc.com.br

Componentes

- **Identificadores**: As regras são as mesmas que comentamos em aula sobre pseudocódigo;
- **Palavras reservadas**: case, char, const, continue, default, break;
- **Símbolos especiais**: diferente de (!=), incremento (++), decremento (--), parênteses ( ), porcentagem (%), atribuição de valor (=);
- **Comentários**:
  - // comentário de uma linha
  - /* comentário em bloco (comentário de várias linhas) */
- **Tipos de dados**

Tipos de Dados

Números

- Se utiliza ponto decimal no lugar da vírgula;
- A notação exponencial em C utiliza a letra “E” ou “e” antes do expoente da base 10, seguindo após essa letra a potência de 10, que deve ser inteira e pode ser positiva ou negativa:
  - 1.2e3 → 1,2 x 10^3
  - -4.5E-6 → -4,5 x 10-6
  - 0.123E-5 → 0,123 x 10-5

Tipos de Dados

Valores lógicos

- Em C não existe o tipo básico lógico. Os valores lógicos estão associados a valores numéricos, ou seja, 0 (zero) é FALSO e qualquer valor diferente de zero significa o valor lógico VERDADEIRO.

Caractere

- Dígito único utilizado entre aspas duplas.

Veremos **strings** com mais cuidado nas próximas aulas

String

- Cadeia de caracteres utilizada entre aspas duplas. Em linguagem C não existe um tipo de dado string nativo.

Tipos de Dados

Tipo	Espaço em memória	Conteúdo armazenado
int	4 bytes	-214783648 a 214783647
float	4 bytes	-3.4e38 a +3.4e38
double	8 bytes	-1.7e308 a +1.7e308
char	1 byte	Caracteres do código ASCII

Declaração de Variáveis

- Todas as variáveis utilizadas em C devem ser declaradas. Recomenda-se que a declaração seja feita no início do programa, antes do primeiro comando;
- Em uma declaração, primeiro é indicado o tipo da variável, depois seu nome:
  - int codigo;
  - float salario;
  - char caract;

ATENÇÃO!

Em C se utiliza **ponto e virgula** ao final de casa comando.

Declaração de Variáveis

- Se há mais de uma variável do mesmo tipo, elas podem ser declaradas em conjunto, separadas por vírgulas:
  - int codigo, valor, ind;
  - float salario_inicial, salario_final, media;

- É possível declarar uma variável e em seguida inicializá-la:
  - int codigo, valor = 0, ind;
  - float salario_inicial, salario_final, media = 0.0;

Declaração de Constantes

- Constantes usualmente são declaradas logo no cabeçalho do arquivo com:

  #define <nome_da_constante> <valor_da-constante>

  #define VALOR_FRETE 10.50

  #define ALTURA_MAXIMA 4

Não é necessário definir o tipo de dado da constante

Atribuição de Conteúdo

- Em C, a atribuição de conteúdo para variáveis é realizada através do operador “=”
- O operador de atribuição define o valor que está à sua direita como conteúdo da variável localizada à sua esquerda, conforme a sintaxe a seguir:

<variável> **=** <conteúdo>**;**

Atribuição de Conteúdo

- O valor da expressão à direita pode ser uma constante, uma variável ou qualquer combinação válida de operandos e operadores;
- Exemplos:

```
x = 4;
val = 2.5;
y = x + 2;
y = y + 4;
tamanho = ‘M’;
altura = ALTURA_MAXIMA;
```

Operadores Aritméticos

Operador	Significado	Operandos
+	adição	inteiros e/ou reais
-	subtração	inteiros e/ou reais
*	multiplicação	inteiros e/ou reais
/	divisão	se inteiros, a divisão será inteira. se pelo menos um operando for real, a divisão será real
%	resto da divisão inteira	inteiro
++	incremento da variável	inteiros ou char
--	decremento da variável	inteiros ou char

Operadores Relacionais

Operador	Significado
==	Igual
!=	Diferente
>	Maior
<	Menor
>=	Maior ou igual
<=	Menor ou igual

Operadores Lógicos

Operador	Significado	Resultado
!	negação	Verdadeiro se o operando for falso, falso se o operando for verdadeiro
&&	e	Verdadeiro somente se ambos os operandos são verdadeiros
||	ou	Verdadeiro se pelo menos um dos operandos for verdadeiro

- !(5 > 3)
- (altura <= altura_maxima) && (largura <= largura_maxima)
- (altura > altura_maxima) || (largura > largura_maxima)

Saída de Dados – printf

A forma geral dessa função é a seguinte:

printf("<string de controle>", <expressões>);

- A <string de controle> contém texto e / ou especificações de formato e literais, ou seja, indicam o tipo de valor a ser apresentado

printf("Mensagem so com caracteres, nenhuma expressao necessaria!");

Saída de dados – printf

Operador	Formato
%c	Um caractere (**char**)
%d ou %i	Um número inteiro decimal (**int**)
%f	Ponto flutuante com precisão simples (**float**)
%lf	Ponto flutuante com precisão dupla (**double**)

Saída de Dados – printf

- printf com especificação de formato na string de controle, exigindo uma expressão após a vírgula:
  - printf("Numero de itens %d", qtde_itens); // inteiro
  - printf("Peso do item %f", peso_item); // float
  - printf("Peso do item %lf com double", peso_item); // double
  - printf("Tamanho do item %c", tamanho_item); // char

- O conjunto de caracteres “\n” dentro da string de controle provoca uma quebra de linha
  - printf("Numero de itens %d \n", qtde_itens);

Entrada de Dados – scanf

A forma geral dessa função é a seguinte:

scanf("<string de controle>" , <lista de argumentos>);

- A <string de controle> contém especificações do formato que indicam o tipo de valor a ser lido e que devem estar de acordo com o tipo da variável;
- Para cada especificação de formato deverá existir na <lista de argumentos> a indicação de uma variável;
- Na função scanf os nomes das variáveis devem ser precedidos por um “e comercial” (&).

Entrada de Dados – scanf

- Leitura de um valor inteiro:

  scanf("**%d**", **&**valorInteiro);

- Leitura de um valor float:

  scanf("**%f**", **&**valorFloat);

- Leitura de um valor double:

  scanf("**%lf**", **&**valorDouble);

- Leitura de um caractere:

  scanf(“ **%c**", **&**soumchar);

ATENÇÃO

Para o uso de **%** e **&**

scanf de **char** deve ter um espaço antes de **%c**

Estrutura de um Programa em C

- Definição do objetivo do programa;
- Inclusão de bibliotecas e funções predefinidas:
  - Um grande número de funções de entrada, saída, aritméticas, etc. podem ser utilizadas em forma de bibliotecas (repositórios de códigos prontos para uso). Para que essas funções possam ser utilizadas, suas bibliotecas devem ser incluídas logo no início do código do programa com **#include<biblioteca.h>**

- Declaração de constantes;
- Função **main()**
  - Função principal presente em todos os programas

Estrutura de um Programa em C

Biblioteca	Função
stdio.h	Manipulação de entrada e saída de dados
stdlib.h	Diversas operações, incluindo conversão, geração de números pseudo-aleatórios, alocação de memória, controle de processo, sinais, busca e ordenação
math.h	Funções matemáticas comuns em computação

Estrutura de um Programa em C

Função	Biblioteca	Argumentos	Resultado	Ação
abs(X)	stdlib.h	integer	integer	Valor absoluto
atan(X)	math.h	double	double	Arco tangente
cos(X)	math.h	double	double	Cosseno
sin(X)	math.h	double	double	Seno
exp(X)	math.h	double	double	e^x
pow(x,y)	math.h	double	double	Eleva x à potência y
sqrt(X)	math.h	double	double	Raiz quadrada
log(X)	math.h	double	double	Raiz quadrada

Estrutura de um programa em C

```
// comentário onde é definido o objetivo do programa
#include ... // inclusão de bibliotecas de funções predefinidas
#include ...
#define... // declarações de constantes, se houver
#define...
int main ( ) // função principal
{
        // declarações de variáveis
        int variavel1;
        float variaveln;
        // comandos
        <comando1>;
        ...
        <comandon>;
        return 0; // valor retornado
}
```

Estrutura de um programa em C

```c
// O programa informa a soma de dois valores lidos
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
int main ( ) {  // função principal
    float valor1, valor2, soma; // declaração de variáveis
    printf ("\nForneca o primeiro valor: ");
    scanf ("%f", &valor1);
    printf ("\nForneca o segundo valor: ");
    scanf ("%f", &valor2);
    soma = valor1 + valor2; // cálculo da soma
    printf ("\nSoma dos dois valores: %f\n", soma); // saída
    return 0; // valor retornado
}
```

```
> ./main

Forneca o primeiro valor: 3.0

Forneca o segundo valor: 2.7

Soma dos dois valores: 5.700000
>
```

Dúvidas

Obrigado!

Prof. Rhafael Freitas da Costa