Estrutura de Dados

Aula 1 – Variáveis em C

Comandos de Entrada e Saída

- O objetivo de escrevermos programas é, em última análise, a obtenção de resultados (Saídas) depois da elaboração de cálculos ou pesquisas (Processamento) através do fornecimento de um conjunto de dados ou informações conhecidas (Entradas).

scanf()

- Instrução responsável por promover leitura de dados. Sua forma geral é:

  scanf(“tipo de entrada”, lista de variáveis)

Alguns “tipos de entrada”:

%c – leitura de caracter

%d – leitura de números inteiros

%f – leitura de número reais

%s – leitura de caracteres

printf()

- Instrução responsável por promover saída de dados. Sua forma geral é:

  printf(“tipo de saida”, lista de variáveis)

- Os “tipos de saída” a serem utilizados pelo printf() no momento serão os mesmos dos “tipos de entrada” do scanf().

Comando de Entrada e Saída

- Exemplo: Dado um número, calcula seu quadrado

```
#include <stdio.h>
int main(){
    int numero;
    printf("Digite um numero: ");
    scanf("%d",&numero);
    printf("%d elevado ao quadrado e' igual a %d. \n", numero,
        numero*numero);
    system("pause");
    return 0;
}
```

Variáveis

- variáveis em um programa C estão associadas a posições de memória que armazenam informações
- nomes de variáveis podem ter vários caracteres em C, apenas os 31 primeiros caracteres são considerados
- o primeiro caracter tem que ser uma letra ou *underscore "_"*
- o restante do nome pode conter letras, dígitos e sublinhados

Variáveis
Exemplos de nomes de variáveis

Corretos

- Contador
- Teste23
- Alto_Paraiso
- _sizeint

Incorretos

- 1contador
- oi!gente
- Alto..Paraíso
- _size-int

Variáveis

- Palavras-chave de C não podem ser utilizadas como nome de variáveis: ***int, for, while, etc...***

- C é *case-sensitive:*

  contador ≠ Contador ≠ CONTADOR

Exercício

- Escolha a opção que inclui somente nomes válidos para variáveis na linguagem C.

a) If, a_b_2, H789, _yes

b) i, j, int, obs

c) 9xy, a36, x*y, --j

d) 2_ou_1, \fim, *h, j

e) Nenhuma das opções anteriores

Tipos de Dados

- O ***tipo*** de uma variável define os valores que ela pode assumir e as operações que podem ser realizadas com ela
- Ex:
  - variáveis tipo ***int*** *recebem apenas valores inteiros*
  - variáveis tipo ***float*** armazenam apenas valores reais

Tipos de dados básicos em C

- **char**: um byte que armazena o código de um caracter do conjunto de caracteres local
- **int**: um inteiro cujo tamanho depende do processador, tipicamente 16 ou 32 bits
- **float**: um número real com precisão simples
- **double**: um número real com precisão dupla

Modificadores de Tipos

- Modificadores alteram algumas características dos tipos básicos para adequá-los a necessidades específicas

- Modificadores:
  - **signed**: indica número com sinal (inteiros e caracteres)
  - **unsigned**: número apenas positivo (inteiros e caracteres)
  - **long**: aumenta abrangência (inteiros e reais)
  - **short**: reduz a abrangência (inteiros)

Constantes

Constantes são valores fixos que não podem ser modificados pelo programa (Tipo/Exemplos)

- char -> ‘a’‘\n’‘9
- int -> 123 1 1000 -23
- long int -> 35000L -45L
- short int -> 10 -12 90
- unsigned int -> 1000U 234U 4365U
- float -> 123.45F 3.1415e-10F
- double -> 123.45 -0.91254’

Constantes char

Barra invertida

- \a bip
- \b backspace
- \n newline
- \t tab horizontal
- \’ apóstrofe
- \” aspa

Declaração de Variáveis

- A declaração de uma variável segue o modelo:

  TIPO_VARIÁVEL lista_de_variaveis;

- Ex:

```
int x, y, z;
float f;
unsigned int u;
long double df;
char c = 'A'; /* variavel definida e iniciada */
char s[ ] = "vetor de caracteres";
```

Operadores

- Operador de Atribuição

  nome_da_variável = expressão

Ex:

```
void main(){
  int x=5;
  int y;

  y = x+3;
}
```

Operadores de Atribuição

Conversão de tipos em Atribuições

- Regra de conversão de tipos:
  - O valor do lado direto é convertido no tipo do lado esquerdo;
  - Quando se converte de inteiros para caracteres, inteiros longo para inteiros e inteiros para inteiros curtos, a regra básica que a quantidade apropriada de bits mais significativos será ignorada.

Operador de Atribuição
Exemplo

```
intx;
char ch;
float f;
void func(void){
    ch = x; /* linha 1 -o valor de ch reflete apenas os 8 bits menos
    significativos; */
    x = f; /* linha 2 –x recebe a parte inteira de f */
    f = ch; /* linha 3 –f converte ch (8 bits) no mesmo valor em formato ponto
    flutuante; */
    f = x; /* linha 4 –f converte x (16 bits) no mesmo valor em formato ponto
    flutuante */
}
```

Operador de Atribuição

Múltiplas atribuições

- C permite a atribuição de mais de uma variável em um mesmo comando:

```
x = y = z = 0;
```

Operadores Aritméticos

- Os operadores aritméticos são usados para desenvolver operações matemáticas. A seguir apresentamos a lista dos operadores aritméticos do C:

- **+** Soma (inteira e ponto flutuante)
- **-** Subtração ou Troca de sinal (inteira e ponto flutuante)
- ***** Multiplicação (inteira e ponto flutuante)
- **/** Divisão (inteira e ponto flutuante)
- **%** Resto de divisão (de inteiros)
- **++** Incremento (inteiro e ponto flutuante)
- **--** Decremento (inteiro e ponto flutuante)

Operadores Relacionais

- Os **Operadores Relacionais** do C realizam comparações entre variáveis, e retornam *verdadeiro* (1) ou *falso* (0). São eles:

Operador	**Ação**
>	Maior do que
>=	Maior ou igual a
<	Menor do que
<=	Menor ou igual a
==	Igual a
!=	Diferente de

Operadores Lógicos

- Operam com valores lógicos e retornam um valor lógico verdadeiro (1) ou falso (0)

Operador	Ação	Exemplo
&&	AND (E)	(c>='0' && c<='0'
||	OR (OU)	(a=='F' || b!=32
!	NOT (NÃO)	(!var)

- Tabela Verdade

a	b	!a	!b	a&&b	a||b
0	0	1	1	0	0
0	1	1	0	0	1
1	0	0	1	0	1
1	1	0	0	1	1

Expressões

- Expressões são combinações de variáveis, constantes e operadores.

- Exemplos:

```
Anos = Dias/365.25;
i = i+3;
c = a*b + d/e;
c = a*(b+d)/e
```

Expressões

- Expressões que podem ser abreviadas

  O C admite as seguintes equivalência, que podem ser usadas para simplificar expressões ou para facilitar o entendimento

Expressão Original	Expressão Equivalente
x = x+k;	x+=k;
x = x-k;	x-=k;
x = x*k;	x*=k;
x = x/k;	x/=k;

Expressões

- **Modeladores (Casts)**

Um modelador é aplicado a uma expressão. Ele força a mesma a ser de um tipo especificado. Sua forma geral é:

(tipo) expressão

```
#include <stdio.h>
Int main(){
        int num;
        float f;

        num = 10;
        f = (float) num/7;
        printf("%f",f);
        return(0);
}
```

Se não tivéssemos usado o modelador no exemplo, o C faria uma divisão inteira entre 10 e 7. O resultado seria 1 (um) e este seria depois convertido para float mas continuaria a ser 1.0. Com o modelador, temos o resultado correto.

Exercícios

Lista 1