

Variáveis

Prof. Debora Medeiros

Baseado nos slides dos Professores: Roseli Romero, Renata Fortes e Fernando Osório

1

Exemplo de potência

- Quadro...

2

Algoritmos => Programas

- Computador:
 - Uso de dados armazenados na memória (variáveis)
 - Instruções bem definidas: os comandos da linguagem
- Ciclo:
 - Entrada de Dados: Ler os dados
 - Processamento: manipular os dados
 - Saída de Dados: Escrever os resultados

3

Algoritmos => Programas

- Programa de Computador: Comandos
- Comandos são ordens para que o computador manipule os dados de sua memória...
- Exemplos de Comandos:

4

Algoritmos => Programas

- Programa de Computador: Comandos
- Comandos são ordens para que o computador manipule os dados de sua memória...
- Exemplos de Comandos:
 - Realizar operações com os dados: somar, subtrair, ...
 - Ler novos dados pelo teclado: entrada de dados
 - Escrever resultados na tela: saída de dados

5

Algoritmos => Programas

- Programa de Computador: Comandos
- Exemplo de Programa: LINGUAGEM "C"
 - Hello world
 - Algoritmo?
 - Comando?

6

Algoritmos => Programas

- Programa de Computador: Comandos
- Exemplo de Programa: LINGUAGEM "C"
 - Hello world
 - Algoritmo?
 - Comando?
 - Saída: **printf()**
 - Biblioteca de comandos: **stdio.h**

7

Saída

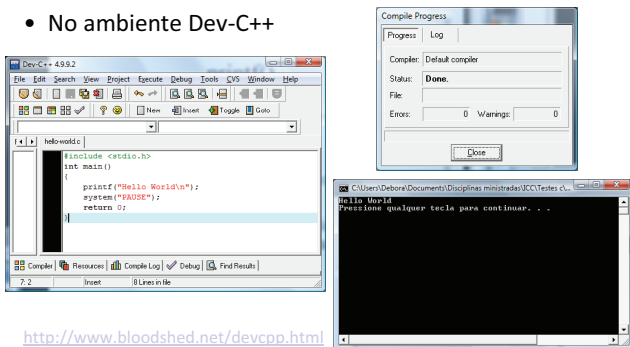
- Programa de Computador: Comandos
- Exemplo de Programa: LINGUAGEM "C"

```
#include <stdio.h>
int main()
{
    printf("Hello World\n");
    system("PAUSE");
    return 0;
}
```

8

Saída

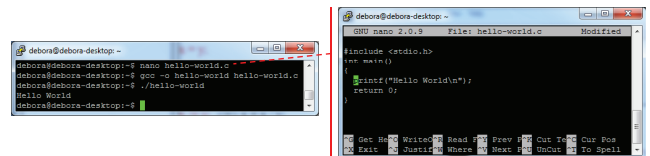
- No ambiente Dev-C++



11

Saída

- Em Linux...



10

Entrada e Saída

- Programa que cumprimenta o usuário
 - Saber o número usp do usuário
 - Exibir: "Hello <número usp>"
 - Entrada?
 - Saída?
 - Armazenar o número usp?

Entrada e Saída

- Programa que cumprimenta o usuário
 - Saber o número usp do usuário
 - Exibir: "Hello <número usp>"
 - Entrada?
 - Comando para receber dados de entrada: `scanf()`
 - Saída?
 - `printf()` de uma forma diferente...
 - Armazenar o número usp?
 - Criar uma variável
 - Posições na memória
 - Tipos

12

Comando de entrada: scanf()

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

scanf("tipo de variável", nome da variável)

– Alguns tipos:

- %c – caractere
- %d – números inteiros
- %f – número reais

13

Entrada e Saída

```
#include <stdio.h>
int main()
{
    int nusp;
    printf("Qual o seu numero usp? ");
    scanf ("%d",&nusp);
    printf("Hello ");
    printf("%d\n",nusp);
    system("PAUSE");
    return 0;
}
```

14

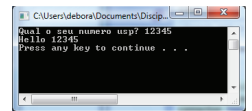
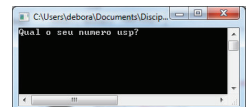
Entrada e Saída

```
#include <stdio.h>
int main()
{
    int nusp;
    printf("Qual o seu numero usp? ");
    scanf ("%d",&nusp);
    printf("Hello ");
    printf("%d\n",nusp); } printf("Hello %d\n",nusp);
    system("PAUSE");
    return 0;
}
```

15

Entrada e Saída

```
#include <stdio.h>
int main()
{
    int nusp;
    printf("Qual o seu numero usp? ");
    scanf ("%d",&nusp);
    printf("Hello %d\n",nusp);
    system("PAUSE");
    return 0;
}
```



- scanf: recebe o endereço da variável (&var), pois altera seu valor
- printf: recebe o valor da variável

16

Processamento

- Soma de 2 números...

17

Processamento

- Soma de 2 números...

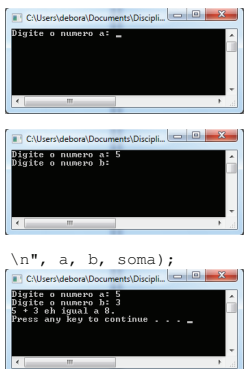
```
#include <stdio.h>
int main()
{
    int a, b, soma;
    printf("Digite o numero a: ");
    scanf ("%d",&a);
    printf("Digite o numero b: ");
    scanf ("%d",&b);
    soma=a+b;
    printf("%d + %d e' igual a %d. \n", a, b, soma);
    system("PAUSE");
    return 0;
}
```

18

Processamento

- Soma de 2 números...

```
#include <stdio.h>
int main()
{
    int a, b, soma;
    printf("Digite o numero a: ");
    scanf("%d",&a);
    printf("Digite o numero b: ");
    scanf("%d",&b);
    soma=a+b;
    printf("%d + %d e' igual a %d. \n", a, b, soma);
    system("PAUSE");
    return 0;
}
```



Tomada de decisão

- Sua forma geral é:

```
if <condicao>
{
    <comandos>;
}
else
{
    <comandos>;
}
```

20

Tomada de decisão

- Exemplo: Programa Menor ou Maior de Idade

21

Tomada de decisão

- Exemplo: Programa Menor ou Maior de Idade

```
#include <stdio.h>
int main()
{
    int idade;
    printf("Digite sua idade: ");
    scanf("%d",&idade);
    if (idade < 18)
        printf("\nMenor de idade\n");
    else
        printf("\nMaior de idade\n");
    system("PAUSE");
    return 0;
}
```

22

Nomes de variáveis

- Palavras-chave de C não podem ser utilizadas como nome de variáveis: **int, if, else, etc...**
- C é *case-sensitive*:
 - contador ≠ Contador ≠ CONTADOR

23

Nomes de variáveis

- o primeiro caracter tem que ser uma letra ou *underscore* “_”
- o restante do nome pode conter letras, dígitos e sublinhados

24

Nomes de variáveis

- o primeiro caracter tem que ser uma letra ou *underscore* “_”
- o restante do nome pode conter letras, dígitos e sublinhados

Contador	1contador
Teste23	int
Alto_Paraiso	Alto..Paraiso
__sizeint	_size-int

25

Nomes de variáveis

- o primeiro caracter tem que ser uma letra ou *underscore* “_”
- o restante do nome pode conter letras, dígitos e sublinhados

<u>Corretos</u>	<u>Incorretos</u>
Contador	1contador
Teste23	int
Alto_Paraiso	Alto..Paraiso
__sizeint	_size-int

26

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

27

Tipos 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

28

Modificadores de tipos

- modificadores alteram algumas características dos tipos básicos para adequa-los a necessidades específicas
- Modificadores:
 - signed**: indica número com sinal
 - unsigned**: número apenas positivo
 - long**: aumenta abrangência
 - short**: reduz a abrangência

29

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 */

30

Operadores

- Correspondem a símbolos simples ou combinados que representam operações de natureza: aritmética, relacional ou lógica.
- Podem ser classificados também quanto a quantidade de elementos sob os quais incidem, i.e., unários, binários ou ternários
- Exemplo:
 - unário Valor++
 - binário Valor1 + Valor2

31

Operadores aritméticos

- Representam as operações aritméticas básicas

Operação	Operador
Adição	+
Subtração	-
Multipliação	*
Divisão	/
Resto da Divisão	%
Incremento (+1)	++
Decremento (-1)	--
Sinal Negativo	-

32

Operadores relacionais

- Estabelecem relações/comparações

Operação	Operador
Igualdade	==
Diferença	!=
Maior	>
Maior ou igual	>=
Menor	<
Menor ou igual	<=

33

Operadores lógicos

- Representam as operações básica dada na lógica matemática

Operação	Operador
Negação	!
Conjunção (E)	&&
Disjunção (OU)	

34

Operadores de atribuição

- Forma geral:
variavel = expressão ou constante
- Armazena o conteúdo dado a direita no elemento dado à esquerda
Salario_Minimo = 465.00;
- Múltiplas atribuições
 - C permite a atribuição de mais de uma variável em um mesmo comando:

x = y = z = 0;

35

Operadores de atribuição

= atribui	x = y;
+= soma e atribui	x += y; <=> x = x + y;
-= subtrai e atribui	x -= y; <=> x = x - y;
*= multiplica e atribui	x *= y; <=> x = x * y;
/= divide e atribui quociente	x /= y; <=> x = x / y;
%= divide e atribui resto	x %= y; <=> x = x % y;

36

Expressões

- Expressões são compostas por:
 - Operandos: a, b, x, Meu_dado, 2, ...
 - Operadores: +, -, %, ...
 - Pontuação: ()
 - Funções da biblioteca do "C": sin(), abs(), sqrt(), ...
- Exemplos:
 - x
 - 14
 - x + y
 - (x + y)*z + w - v
 - (-b + sqrt(delta)) / 2*a

37

Expressões

- expressões retornam um valor:
 - $x = 5 + 4$ /* retorna 9 */
 - esta expressão atribui 9 a x e retorna 9 como resultado da expressão
 - $((x = 5 + 4) == 9)$ /* retorna true */
 - na expressão acima, além de atribuir 9 a x, o valor retornado é utilizado em uma comparação

38

Expressões

- expressões podem aparecer em diversos pontos de um programa
 - Comandos
 - x = y+z;
 - parâmetros de funções
 - sqrt(x + y);
 - condições de teste
 - if (x == (y+z));

39

Expressões

- a ordem em que uma expressão é avaliada depende da prioridade dos operadores e da pontuação

40

Expressões

Maior precedência

()
! ++ --
* / %
+ -
> < <= >=
== !=
&&
||
= += -= *= /=

Menor precedência

41