**Exercícios – Capítulo 1 (C Como Programar) – Deitel;**

* Exercícios de autorrevisão

2.1

a) todo programa em C inicia sua execução pela função main.

b) a chave a direita inicia o corpo de cada função, e a chave a esquerda encerra o corpo de cada função.

c) toda instrução termina com um ponto e vírgula.

d) a função printf da biblioteca padrão exibe informações na tela.

e) a sequência de escape “\n” representa o caractere de nova linha, que faz com que o cursor se posicione no início da próxima linha na tela.

f) a função scanf da biblioteca padrão é usada para obter dados do teclado.

g) o especificador de conversão %d é usado em uma string de controle de formato de scanf para indicar que um inteiro será digitado, e em uma string de controle de formato de printf para indicar que um inteiro será exibido.

h) sempre que um novo valor é colocado em uma posição da memória, ele anula o valor que ocupava essa mesma posição anteriormente. Esse processo é chamado de destrutivo.

I) quando o valor é lido em uma posição da memória, o valor nessa posição é preservado; esse processo é chamado de não destrutivo.

j) a instrução if é usada na tomada de decisões.

2.2

a) a função printf sempre começa a imprimir no início de uma nova linha.

Resposta: Falso.

ERRATA: a instrução printf sempre começa a imprimir onde o cursor estiver posicionado, e isso pode ser em qualquer lugar de uma linha na tela.

b) comentários fazem com que o computador imprima na tela o texto delimitado por /\* e\*/ quando o programa é executado.

Resposta: Falso.

ERRATA: os comentários não causam qualquer ação quando o programa é executado. Eles são usados para documentar programas e melhorar sua legibilidade.

c) a sequência de escape \n, quando usada em uma string de controle printf, faz com que o cursor se posicione no início da próxima linha na tela.

Resposta: Verdadeiro.

d) todas as variáveis precisam ser declaradas antes de serem usadas.

Resposta: Verdadeiro.

e) todas as variáveis precisam receber um tipo ao serem declaradas.

Resposta: Verdadeiro.

f) C considera as variáveis número e “NúMeRo” Idênticas.

Resposta: Falso.

ERRATA: a linguagem C diferencia maiúsculas de minúsculas, de modo que essas variáveis são diferentes.

g) declarações podem aparecer em qualquer parte do corpo da função.

Resposta: Falso.

ERATA: as declarações precisam aparecer após a chave à esquerda do corpo de uma função e antes de quaisquer instruções executáveis.

h) todos os argumentos após a string de controle de formato de uma função printf precisam ser precedidas por (&).

Resposta: Falso.

ERRATA: os argumentos em uma função printf normalmente não devem ser precedidos por um (&). Os argumentos após a string de controle de formato em uma função scanf normalmente deve ser precedido por um (&).

I) o operador módulo (%) só pode ser utilizado com operando inteiros.

Resposta: Verdadeiro.

j) os operadores aritméticos \*, /, %, + e – têm o mesmo nível de precedência.

Resposta: Falso.

ERRATA: os operadores \*, /, e % estão no mesmo nível de precedência, e os operadores + e – estão em um nível de precedência mais baixo.

k) os nomes das variáveis a seguir são idênticos em todos os sistemas C padrão.

*1 - Esteéumnúmerosuperhiperlongo1234567*

*2 - Esteéumnúmerosuperhiperlongo1234568*

Resposta: Falso.

ERRATA: alguns sistemas podem distinguir os identificadores com mais de 31 caracteres.

l) um programa que exibe três linhas de saída precisa conter três instruções printf.

Resposta: Falso.

ERRATA: uma instrução printf com várias sequências de escape “\n” pode exibir várias linhas.

2.3

a) int c, estavariavel, q76354, numero;

b) printf(“Digite um inteiro: ”);

c) scanf (“%d”, &a);

d) if (numero != 7) {

printf (“A variável numero não é igual a 7.”); }

e) printf (“Este é um programa em C.\n”);

f) printf (“Este é um programa\nem C.\n”);

g) printf (“Este\né\num\nprograma\nem\nC.\n”);

f) printf (“Este\té\tum\tprograma\tem\tC.\n”);

2.4

a) /\* calcula o produto de três inteiros\*/

b) int x, y, z, resultado;

c) printf (“Digite três resultados:\n”);

d) scanf (“%d %d %d”, &x, &y, &z);

e) resultado = x\*y\*z;

f) printf (“O produto é %d.\n\n”, resultado);

2.5

#include <stdio.h>

int main () {

/\* calcula o produto de três inteiros\*/

int x, y, z, resultado;

printf (“Digite três resultados:\n”);

scanf (“%d %d %d”, &x, &y, &z);

resultado = x\*y\*z;

printf (“O produto é %d.\n\n”, resultado);

return 0;

}

2.6

a) o erro é &numero.

b) ausência de (&) em numero2.

c) o ponto e vírgula após a condição do if.

d) o correto seria >=.

* Exercícios

2.7

a) scanf("%d", &valor");

b) ausência de uma variável;

c) somaDosNumeros = primeiroNumero + segundoNumero;

d) if (numero >= maior) {

maior == numero; }

e) /\*txt\*/

f) scanf("%d", &umInteiro);

g) ausência de um identificador que printe a última variável na tela;

h) if(x == y) {

printf("%d é igual a %d\n", x, y); }

i) printf ("A soma é %d\n",x+y );

j) printf("O valor que você digitou é: %d\n", valor);

2.8

a) comentários são usados para documentar um programa e melhorar a sua legibilidade.

b) a função usada para exibir informações na tela é printf.

c) uma instrução C responsável pela tomada de decisão é if.

d) os cálculos normalmente são realizados por instruções operadores.

e) a função scanf lê os valores do teclado.

2.9

a) printf("Digite dois números: ");

b) a = b \* c;

c) /\* O objetivo desse programa é informar o salário líquido do

Trabalhador, após ser descontadas as inúmeras variáveis de

Desconto. \*/

d) scanf("%d, %d, e %d.\n\n", &a, &b, &c);

2.10

a) os operadores em C são avaliados da esquerda para a direita.

Resposta: Verdadeiro.

b) os seguintes nomes de variáveis são válidos: (...).

Resposta: Verdadeiro.

c) a instrução printf(“a = 5”); é um exemplo típico de instrução de atribuição.

Resposta: Falso.

ERRATA: printf não é uma função de atribuição, mas sim uma função responsável por imprimir valores na tela.

d) uma expressão aritmética válida sem parênteses é avaliada da esquerda para a direita.

Resposta: Verdadeiro.

e) os seguintes nomes de variáveis são válidos: (...)

Resposta: Falso.

ERRATA: uma variável não pode ser carácter de inicialização caso ela comece com algum número, ou seja, uma palavra reservada as palavras chaves da programação em C.

2.11

a) que operações aritméticas estão no mesmo nível de procedência da multiplicação?

Resposta: % e /.

b) quando os parênteses são aninhados, qual conjunto de parênteses é avaliado em primeiro lugar em uma expressão aritmética?

Resposta: da esquerda para a direita, respeitando a precedência dos operadores aritméticos.

c) uma posição na memória do computador que pode conter diferentes valores em vários momentos durante a execução de um programa é chamada destrutiva.

2.12

a) 2.

b) 4.

c) x=.

d) x=2.

e) 5=5.

f) nada.

g) nada.

h) nada.

i) \_\_

\_\_

2.13

a) b, c, d, e, f.

b) p.

c) nada.

d) nada.

2.14

Resposta: letra e.

2.15

Para todos os casos a ordem é parêntesis, agrupamentos prévios,

Potência, radiciação, multiplicação, divisão, adição e subtração.

2.28

Uma tentativa de dividir por zero normalmente é indefinida em sistemas de computador, e geralmente resulta em um erro fatal que faz com que o programa seja encerrado imediatamente. Erros não fatais permitem que os programas sejam executados até o fim, e frequentemente produzem resultados incorretos.