*Exercícios de revisão*

2.1

Preencha as lacunas de cada uma das frases seguintes:

a) Todos os programas em C começam sua execução com a função main.

b) A {} começa o corpo de todas as funções e a o termina.

c) Todas as instruções terminam com um ; .

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 fornecido ao programa e em uma string de controle de formato de printf para indicar a impressão (saída) de um inteiro pelo programa.

h) Sempre que um valor novo é colocado em uma posição da memória, ele substitui o valor anterior ali presente. Esse processo é conhecido como leitura Leitura destrutiva (destructive reading).

i) Quando um valor é lido de uma posição na memória, ele é preservado; isso é chamado leitura Leitura não destrutiva.

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

2.2

Diga se cada uma das afirmações seguintes é verdadeira ou falsa. Se for falsa, explique por quê.

a) Quando a função printf é chamada, ela sempre começa a imprimir no início de uma nova linha. Não, para ela imprimir numa próxima linha deve-se colocar \n ao fim do texto inserido dentro da função.

b) Os comentários fazem com que o computador imprima na tela o texto situado entre /\* e \*/ quando o programa é executado. Não, o compilador ignora os caracteres dentro de /\* e \*/ pois esses são comentários.

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

Sim, verdadeiro.

d) Todas as variáveis devem ser declaradas antes de serem usadas.

Sim, verdadeiro.

e) Todas as variáveis devem receber a atribuição de um tipo ao serem declaradas.

Sim, verdadeiro.

f) O C considera idênticas as variáveis numero e NuMeRo.

Não, pois a linguagem C é case sensitve, ou seja, ela difere as letras maiúsculas de minúsculas logo numero e NuMeRo não são iguais.

g) As declarações podem aparecer em qualquer lugar do corpo de uma função.

Falso. As declarações devem aparecer depois da chave esquerda do corpo de uma função e antes de qualquer instrução executável.

h) Todos os argumentos após a string de controle de formato em uma função printf devem ser precedidos por um e-comercial (&).

Falso. Normalmente, os argumentos em uma função printf não devem ser precedidos por um e-comercial (&). Normalmente, os argumentos após a string de controle de formato em uma função scanf devem ser precedidos por um e-comercial. Analisaremos as exceções nos Capítulos 6 e 7.

i) O operador resto (%) só pode ser usado com operadores inteiros.

Sim, verdadeiro

j) Os operadores aritméticos \*, /, % e - possuem o mesmo nível de precedência.

Não, o operador de subtração – está em um nível abaixo de precedência.

k) Verdadeiro ou falso: Os nomes de variáveis a seguir são idênticos em todos os sistemas ANSI C:

**vejaumnomesuperhiperlongol234567**

**vejaumnomesuperhiperlongol234568**

Falso. Alguns sistemas podem fazer distinção entre identificadores com mais de 31 caracteres

1. Verdadeiro ou falso: Um programa em C que imprime três linhas de saída deve conter três instruções **printf.**

Falso, podemos imprimir 3 linhas dentro de um mesmo print usando o \n

2.6 Identifique e corrija os erros de cada uma das seguintes instruções:

a) printf("O valor e %d\n", inúmero);

b)scanf("%â%ã", inumerol, numero2); Falta o &

c) if (c < 7);

printf("C e menor do que 7\n");

d) if (c => 7)

printf("C e igual ou menor do que 7\n");

2.7 Identifique e corrija os erros em cada uma das instruções a seguir (Nota: pode haver mais de um erro por instrução):

a) scanf("d", valor); Falta o % e &

b) printf("O produto de %d e %â e %d"\n, x, y) ; \n fora das aspas

c) primeiroNumero + segundoNumero = somaDosNumeros Erro de sintaxe

d) if (numero => maior)

maior == numero;

e) \*/ Programa para determinar o maior de tres inteiros /\* /\* \*/

f) Scanf ("%á", umlnteiro) ; Falta &

g) printf("0 resto de %d dividido por %â e\n", x, y, x % y); Faltou o terceiro conversor para printar o resto da divisão

h) if (x = y);

printf(%d e igual a %d\n", x, y);

i) print("A soma e %d\n, " x + y) ; printf

j) Printf("O valor fornecido e: %d\n, &valor); printf

2.8 Preencha as lacunas em cada uma das expressões a seguir:

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

b) A função usada para imprimir informações na tela é printf.

c) Uma instrução do C para a tomada de decisões é if.

d) Normalmente, os cálculos são realizados por instruções matemáticas??

e) A função scanf(); fornece ao programa os valores digitados no teclado.

2.10 Diga se cada uma das expressões a seguir é verdadeira ou falsa. Explique suas respostas,

a) Os operadores em C são calculados da esquerda para a direita.

Sim, seguindo a ordem de precedência

b) Todos os nomes de variáveis a seguir são válidos: \_barra\_inf erior, m928134, t5, j7, suas\_vendas, total\_sua\_conta, a, b, c, z e z2.

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

Não, não é possível fazer atribuição em prompt

d) Uma expressão aritmética válida em C e sem parênteses é calculada da esquerda para a direita.

Sim

e) Todos os nomes de variáveis a seguir são válidos: 3g, 87, 67h2, h22 e 2h.

Somente h22, nomes de variáveis não se iniciam por números

2.11 Preencha as lacunas de cada uma das expressões a seguir:

a) Que operações aritméticas estão no mesmo nível de precedência que a multiplicação? . Divisão e resto da divisão

b) Quando os parênteses são aninhados, que conjunto de parênteses de uma expressão aritmética é calculado em primeiro lugar? . O conjunto mais interno

c) O local da memória do computador que pode conter valores diferentes em várias ocasiões ao longo da execução de um programa é chamado. Variável

2.12 O que é impresso quando cada uma das instruções seguintes da linguagem C é executada? Se nada for impresso, responda "nada". **Admita x = 2ey = 3.**

a) printf("%d", x); 2

b) printf("%d", x + x); 4

c) printf ("x=") ; Nada

d) printf("x=%d", x) ; 2

e) printf("%ã - %â", x + y, y + x); 5 - 5

f) z = x + y; Nada

g) scanf("%ã%ã", &x, &y); Nada

h) /\* printf("x + y = %ã", x + y); \*/ Nada

i) printf("\n"); Nada

2.13 Quais das instruções seguintes em C, se houver alguma, contêm variáveis envolvidas com leitura destrutiva?

a) scanf("%d%d%d%d%d", &b, &c, &d, &e, &f);

b)p = i + j+ k + 7;

c) printf("Leitura destrutiva");

d) printf("a = 5");

2.14 Dada a equação y = ax 3 + 7, qual das instruções em C a seguir, se houver alguma, são corretas para ela?

a)y = a\*x\*x\*x + 7;

b)y = a\*x\*x\* (x+7);

c) y = (a \* x) \* x \* (x + 7) ;

d) y = (a \* x) \* x \* x + 7;

e) y = a \* (x \* x \* x) + 7;

f)y = a\*x\* (x \* x + 7);

2.15 Diga a ordem de cálculo dos operadores em cada uma das instruções em C a seguir e mostre o valor de x depois que cada instrução for executada.

a)x = 7 + 3\*6/2-l 1: 3\*6 2: /2 3: 7 + 4: -l X=15

b)x = 2%2 + 2\*2-2/2; 1: 2%2 2: 2\*2 3: 2/2 4: + 5: - X=3

c) x = (3 \* 9 \* (3 + (9\*3/ (3) ) ) ); 1: (9\*3/ (3) ) 2: 27/3 3: (3 + (9\*3/ (3) ) ) 4: 9 \* (3 + (9\*3/ (3) ) ) 5: 3 \* X= 324