**Exercícios – Capítulo 3(C Como Programar) – Deitel;**

* Exercícios de autorrevisão

3.1

a) o procedimento para resolver um problema em termos de ações a serem executadas e da ordem em que essas ações devem ser executadas é chamado de algoritmo.

b) especificar a ordem de execução das instruções pelo computador é chamado de controle do programa.

c) todos os programas podem ser escritos nos termos de três tipos de estrutura de controle sequência, seleção e repetição.

d) a estrutura de seleção if/else é usada para executar uma ação quando uma condição seja verdadeira e outra ação quando essa condição seja falsa.

e) várias instruções agrupadas com chaves ({e}) são chamadas de instrução composta.

f) a estrutura de repetição while especifica que uma instrução ou grupo de instruções deve ser executada repetidamente enquanto alguma condição permanece verdadeira.

g) a repetição de um conjunto de instruções por um número específico de vezes é chamada de repetição controlada por contador.

h) quando não se sabe com antecedência quantas vezes um conjunto de instruções será repetido, um valor de sentinela pode ser usado para encerrar a repetição.

3.2

X = X + 1;

X += 1;

++X;

X++;

3.3

a) z= x++ + y;

b) produto \*= 2;

c) produto = produto \* 2;

d) if (contador > 10)

printf (“Contador é maior que 10.\n”);

e) total -= --x;

f) total += x--;

g) q %= divisor;

q = q % divisor;

h) printf (“%.2f”, 123.456);

123.45 é exibido

i) printf (“%.3f”, 123.456);

123.456 é exibido

3.4 / 3.5

/\*calcula a soma dois inteiros de 1 a 10\*/

#include <stdio.h>

int main (void)

{

int soma, x;

x = 1;

soma = 0;

while (x<= 10) {

soma +=x

++x;

}

printf("\nA soma é: %d.\n\n", soma);

return 0;

}

3.6

produto = 25, x = 6;

3.7 / 3.8

/\*Eleva x à potência y\*/

#include <stdio.h>

int main (void)

{

int x, y, i, potencia;

i = 1;

potencia = 1;

scanf("%d", &x);

scanf("%d", &y);

while (i <= y){

potencia \*= x;

++i;

}

printf("%d", potencia);

return 0;

}

3.9

a) erro: falta a chave à direita do while.

Correção: adicione a chave à direita depois da estrutura c++.

b) erro: precisão usada na especificação de conversão scanf.

Correção: retire .4 da especificação de conversão.

c) erro: ponto e vírgula depois do else em uma estrutura if...else resulta em um erro lógico. O segundo printf continuará a ser executado.

Correção: retire o ponto e vírgula que parece depois do else.

3.10

O valor da variável z nunca muda da estrutura while. Portanto, cria-se um loop infinito. Para impedir que isso aconteça z precisa ser decrementado, de modo que seu valor chegue a 0.

* Exercícios

3.11

a) erro: ponto e vírgula após o if, executando seu conteúdo antes da verificação.

Correção: remover o ponto e vírgula após o if.

b) erro: o valor da variável total não inicia com zero, ela contém valor armazenado, resultado um erro lógico não fatal.

Correção: atribua o valor total, inicialmente, como zero.

c) erro: ausência de chaves para atribuir o conteúdo do bloco, logo será executado sem que ocorra a verificação.

Correção: adicione as chaves para fechar o bloco da estrutura de repetição while.

d) erro: a estrutura de parada dentro do comando de repetição while está incorreta, fazendo com que seja executado um loop sem tempo de parada.

Correção: enquanto y for maior que algum número natural diferente de 0, executar o comando, é uma possibilidade de correção.

3.12

a) a solução para qualquer problema envolve realizar uma série de ações em um (a) ordem específico (a).

b) um sinônimo para procedimento é algoritmo.

c) uma variável que acumula a soma de vários números é um (a) contador.

d) a definição de certas variáveis com valores específicos no início de um programa é chamada de atribuição.

e) um valor especial usado para indicar o fim da entrada de dados é chamado de valor sentinela, valor de sinal, valor fictício, ou valor de chave.

f) um (a) fluxograma é uma representação gráfica de um algoritmo.

g) em um fluxograma, a ordem em que as etapas devem ser realizadas é indicada por símbolos gráficos.

h) o símbolo de finalização indica o (a) parada e fim da sequência de cada algoritmo.

i) retângulos correspondem a cálculos que normalmente são realizados por instruções e operações de entrada/saída que normalmente são realizadas por chamadas às funções pow (x, y) e sqrt () da biblioteca-padrão.

j) o item escrito dentro de um símbolo de decisão é chamado de parada.

3.13

O programa imprime a soma dos quadrados dos números naturais de 1 a 10.

3.14

a) printf (“Digite dois números: ”);

b) z = x + y + z;

c) if (m > 2v)

d) scanf (“%d %d %d”, &s, &r, &t);

3.15

a)

//declare as variáveis soma, numero1 e numero2

//peça ao usuário para atribuir os valores de numero1 e numero2

//atribua a soma de numero1 e numero2 a variável soma

//imprima na tela a variável soma;

b)

//declare as variáveis numero1 e numero2

//peça ao usuário para atribuir os valores de numero1 e numero2

//se numero1 é maior que numero2’

// imprima “Número 1 é maior. ”

//se não

// se numero1 é menor que numero2

// imprima “Número 2 é maior. “

// se não

// imprima “Número 1 é igual a Número 2. ”

c)

//declare as variáveis numero e soma

//atribua a variável soma o valor zero

//enquanto numero for diferente de -1

//peça que o usuário insira um número inteiro par

//atribua a variável soma ela mesma mais o número digitado

//se o usuário digitou –1

//fim da estrutura de repetição

//mostrar na tela a variável soma;;

3.16

a) a experiência tem mostrado que a parte mais difícil da solução de um programa em um computador é a produção de um programa em C funcional.

Resposta: Verdadeiro.

b) um valor de sentinela deve ser um valor que não possa ser confundido com um valor de dados legítimos.

Resposta: Verdadeiro.

c) linhas de fluxo indicam as ações a serem realizadas.

Resposta: Verdadeiro.

d) condições escritas dentro de símbolos de decisão sempre contêm operadores aritméticos.

Resposta: Falsa.

ERRATA: não necessariamente precisa ter operadores aritméticos, mais sim de desigualdade, ou só o valor da variável em caso de variáveis do tipo booleanos.

e) nos refinamentos sucessivos top-down, cada refinamento é uma representação é uma representação completa do algoritmo.

Resposta: Falsa.

ERRATA: representa somente uma parte do algoritmo.