Capítulo 5 – Funções em C

Exercícios de Autorrevisão

5.1 - ) Preencha os espaços com as seguintes sentenças:

1. Um módulo de progama em C é chamado de função.
2. Uma função é invocada com um(a) chamada de função.
3. Uma variável conhecida apenas dentro da função que é definida é chamada de variável local.
4. O comando return em uma função chamada é usada para passar o valor de uma expressão de volta a função chamadora.
5. A palavra-chave void é usada em um cabeçalho de função para indicar que uma função não retorna um valor ou para indicar que uma função não contém parâmetros.
6. O(A) escopo de um indentificador é a parte do progama em que o identificador pode ser usado.
7. As três maneiras de retornar o controle de uma função chamada à chamadora são return, return expressão e econtrando a chave direta de fechamento de uma função.
8. Um(a) prótotipo de uma função permite que o compilador verifique o número, os tipos e a ordem dos argumentos passados a uma função.
9. A função rand é usada para produzir números aleatórios.
10. A função srand é usada para definir a semente do número aleatório para randomizar um progama.
11. Os especificadores de classe de armazenamento são auto, register, extern e static.
12. Supõe-se que as variáveis declaradas em um bloco ou na lista de parâmetros de uma função pertençam à classe de armazenamento auto, a menos que especificadas de outras maneiras.
13. O especificador de classe de armazenamento register é uma recomendação ao compilador para que armazene uma variável em um dos regitradores do computador.
14. Uma variável não static definida fora de qualquer bloco ou função é uma variável externa ou global.
15. Para que uma variável local em uma função retenha seu valor entre as chamadas a uma função, ela precisa ser declarada com o especificador de classe de armazenamento static.
16. Os quatro escopos possíveis de uma função são escopo de uma função, escopo do arquivo, escopo do bloco e escopo de protótipo de função.
17. Uma função que chama a si mesma, direta ou indiretamento, é uma função recursiva.
18. Uma função recursiva normalmente tem dois componentes: aquele que oferece um meio para a recursão terminar ao testar um caso básico e um que expressa o problema como uma chamada recursiva para um problema ligeiramente mais simples que a chamada original.

5.2 -)

a) Escopo do bloco.

b) Escopo do bloco.

c) Escopo de arquivo.

d) Escopo de arquivo.

e) Escopo de arquivo.

f) Escopo de protótipo de função.

5.4-)

a) double hypotenuse (double side1, double side2)

b) int smallest (int x, int y, int z)

c) void instructions (void)

d) float intToFloat (int number)

5.5-)

a) double hypotenuse (double side1, double side2)

b) int smallest (int x, int y, int z)

c) void instructions (void)

d) float intToFloat (int number)

5.6-)

a) register int count = 0;

b) static float lastVal;

c) static int number;

5.7)

a) Erro: A função h é definida na função g.

Correção: Mova a definição de h para fora da definição de g.

b) Erro: O corpo da função deveria retornar um inteiro, mas não o faz.

Correção: Exclua a variável result e coloque o seguinte comando na função: return x + y;

c) Erro: O resultado de n + sum(n – 1) não é retornado; sum retorna um resultado imprópio.

Correção: Rescreva o comando na claúsula else como: return n + sum(n – 1).

d) Erro: Ponto e vírgula após o parênteses à direita que delimita a lista de parâmetros, e redefinição do parâmetro a na definição daa função.

Correção: Retire o ponto e vírgula após o parêntese à direita na lista da parâmetros e exclua a declaração float a: no corpo da função.

e) Erro: A função retorna um valor quando não deveria.

Correção: Elimine o comando return.