

09/09/2019

Lista de exercícios
LP1-ADS

Prof. Luciano Bernardes de Paula

(Lista adaptada do livro “Treinamento em linguagem C” – Victorine Viviane Mizrahi, ec. Pearson).

Exercícios para resolver com laço *for*

1 – Escreva um programa que receba um valor n positivo e escreva esse valor n vezes na tela. Seu programa deve testar e só aceitar valores positivos.

Exemplo de uso:

Entre com um valor positivo: 3

3
3
3

2 – Escreva um programa que receba um valor n positivo e escreva uma contagem regressiva do valor até 0. Seu programa deve testar e só aceitar valores positivos.

Exemplo de uso:

Entre com um valor positivo: 5

5 4 3 2 1 0

3 – Escreva um programa que receba um valor n positivo e apresente uma contagem regressiva do valor até $-n$.

Exemplo de uso:

Entre com um valor positivo: 5

5 4 3 2 1 0 -1 -2 -3 -4 -5

4 – Escreva um programa que receba dois valores n_1 e n_2 . Se n_2 for igual a 1, o programa imprime n_1 em uma contagem regressiva do valor até 0, se n_2 for igual a 0, o programa imprime n_1 em uma contagem progressiva do valor 0 até n_1 . Faça testes para certificar que n_2 tenha como valor recebido somente 0 ou 1.

Exemplo 1

Entre com n_1 : 5

Entre com n_2 : 1

5 4 3 2 1 0

Exemplo 2

Entre com n_1 : 5

Entre com n_2 : 0

0 1 2 3 4 5

5 (c4ex19) – Escreva um programa que receba três números a , b e c , sendo $a > 1$ e $b < c$. O programa deve a quantidade de números entre b e c que são divisíveis por a .

Exemplo

Entre com a : 2

Entre com b: 2

Entre com c: 10

Entre b e c existem 3 numeros divisiveis por 2.

Exercícios para resolver com *while* ou *do-while*

6 – Escreva um programa que receba um valor. Se esse valor for par, ele escreve uma mensagem na tela e pede um novo valor. Se o valor for ímpar, ele escreve uma mensagem na tela e termina.

Exemplo

Entre com um numero: 4

O numero 4 é par.

Entre com um numero: 6

O numero 6 é par.

Entre com um numero: 8

O numero 8 é par.

Entre com um numero: 5

O numero 5 é ímpar.

Finalizando...

7 – Escreva um programa que receba 2 valores: x e y. Os valores devem seguir a seguinte regra: $x < y$. Faça testes para certificar que esse critério está sendo atendidos. O programa deve, após isso, executar um laço no qual é computado $x = x + y$ e testar novamente se $x < y$, terminando quando essa expressão não for mais verdadeira. No final, o programa deve apresentar quantas vezes o cálculo foi feito para que $x \geq y$.

8 – Faça um programa que use o comando switch. O usuário deve entrar com um valor entre 0 e 2 e, para cada valor, uma frase diferente deve ser impressa na tela. Faça também uma opção *default*. O programa deve se repetir enquanto o usuário não digitar um valor negativo.

9 – Escreva um programa que recebe um valor x. Em seguida recebe um novo valor. Enquanto o segundo valor entrado não for maior que x, ele continua pedindo um novo valor, caso contrário, o programa termina.

10 - Implemente um programa que forneça quantos números devem existir em sequência a partir do 1 (1, 2, 3, 4, ...) para que a sua soma ultrapasse 250. O programa deve utilizar apenas a operação while ou do-while.

Para os próximos exercícios, escolha o melhor laço para resolver os problemas. Alguns, inclusive, devem ser resolvidos utilizando mais de um tipo de laço de repetição.

11 – Escreva um programa que receba um valor n e este imprima na tela o mesmo valor n, n vezes. Em seguida é pedido outro valor para o usuário e a operação é repetida. O programa finaliza se o usuário entrar com um valor maior que 500.

12 – Faça um programa que recebe um valor de 2 a 9 e retorna na tela a tabuada desse valor.

13 – Repita o exercício anterior, porém, para cada conta da tabuada do número inserido pelo usuário, o usuário que deve responder com o resultado. O programa então deve apresentar se o valor inserido pelo usuário foi correto ou não e passar para a próxima conta. No final, o programa apresenta quantas respostas certas o usuário teve.

14 – Repita o Exercício 14, porém, no momento que o usuário errar uma das contas, o programa deve reiniciar aquela tabuada desde o início. O programa só termina quando o usuário acertou todas as contas em sequência, sem errar nenhuma.

15 – Escreva um programa que recebe uma letra do teclado até que o usuário digite a letra 'z'. Dica: use a função `getche()`.

16 - Escreva um programa que recebe uma letra do teclado até que o usuário digite a letra 'z' ou 'Z'. Dica: use a função `getche()`.

17 (c3ex19) – Escreva um program que receba um valor n e um valor r e calcule:

$$\text{Res} = n! / (r! * (n - r)!)$$
 (equação que calcula a combinação de n objetos dentro de um conjunto de r).

Sendo n! o fatorial de n que é calculado fazendo:

$$n! = n * (n-1) * (n-2) * \dots * 1$$

18 (c3ex20) – Escreva um programa que receba um valor inteiro n e apresente o fator enésimo da sequência de Fibonacci.

A sequência de Fibonacci é dada por:

1, 1, 2, 3, 5, 8, 13, 21, ...

Ou seja, o primeiro e o segundo fatores são iguais a 1, a partir do terceiro fator, o valor é a soma dos dois anteriores.

19 (c3ex18) – Escreva um programa que receba um valor entre 3 e 18 e indique:

a) quantas são as maneiras de que o valor possa ser representado com 3 dados;

b) qual a probabilidade de que o valor indicado seja sorteado com 3 dados.

Fórmula:

$$\text{probabilidade} = (n1 / n2) * 100$$

onde n1 é a quantidade de combinações que o valor indicado pode ser representado e n2 é o número total de combinações.

20 – Faça um programa que simule dois semáforos de trânsito. O primeiro semáforo muda de estado (vermelho para verde ou vice-versa) a cada 10 segundos. O segundo semáforo muda de estado a cada 48 segundos. Indique, para um período de 1000 segundos, quantas vezes os dois semáforos estarão no mesmo estado (vermelho ou verde) ao mesmo tempo. Considere os semáforos iniciando iguais. Dica: você pode

representar os semáforos como uma variável **int** sendo que se o valor for 0, representa uma das cores e 1, representa a outra cor. Ou mesmo uma variável **char** que recebe duas letras distintas para cada cor ('v' para verde e 'm' para vermelho, etc).