Exercícios Propostos.

1. Indique quais afirmações são **falsas** e quais são **verdadeiras**:
   1. . A condição dentro de um laço *while* *e do..while* tem que ser colocada sempre dentro de parênteses.

Verdadeiro

* 1. . Os laços *while* e *for* executam **SEMPRE**, pelo menos uma vez o corpo do laço.

Falso

* 1. . O laço *do..while* executa **SEMPRE**, pelo menos uma vez, o corpo do laço.

Verdadeiro

1.4. As três componentes, dentro de parênteses, do laço *for* são todas obrigatórias.

Falso

1.5. As três componentes, dentro de parênteses, do laço *for* são todas facultativas.

Verdadeiro

1.6. Quando em um laço *for* se tem que realizar mais do que uma carga inicial ou mais do que um incremento, as diversas instruções, em cada uma das componentes, devem ser separadas por virgula e não ponto-e-vírgula, de forma a manter o formato do laço for.

Verdadeiro

1.7. A instrução de um laço é sempre executada.

Falsa

1.8. No laço *for*, o número de vezes que as cargas iniciais são executadas é sempre igual ao número de iterações do laço.

Verdadeiro

1.9. No laço *for* ou *while, o número de vezes que a condição e testada e sempre igual ao número de iteração do laço*.

Verdadeiro

1.10. No laço *for,* o número de vezes que a instrução e executada e sempre igual ao número de vezes que é executada a pós-instrução.

... Falsa

1.11. Um laço, quando está dentro de outro laço, necessita de chaves.

Verdadeiro

1.12. Não há limite para o número de laços dentro de outros laços.

Verdadeiro

1.13. No laço *do..while*, a instrução é sempre executada pelo menos uma vez.

Verdadeiro

1.14. Em todos os laços, a condição é sempre testada pelo menos uma vez.

Falso

1.15. Os laços *while* e *do..while* incrementam automaticamente a variável de controle.

Falso

1.16. Os laços são executados até que a condição escrita se verifique.

......Verdadeiro

1.17. O laço *do..while* aparece escrito menos frequentemente do que qualquer dos outros.

Verdadeiro

1.18. Um *break*, quando presente dentro de um laço, termina o programa.

....Verdadeiro

1.19. No laço *while*, antes de ser executar a instrução *continue*, deve-se atualizar a variável de controle do laço.

....Verdadeiro

1.20. Qualquer dos laços pode ser sempre escrito a partir do laço while.

......Falso

1. Escreva genericamente o laço **do...while** como um laço **while**.

do {

<Instrução>

<Incremento/decremento/contador>

} (<condição>);

....................................................................................................................

while (<condição>) {

<Instrução>

<Incremento/decremento/contador>

}

1. Escreva genericamente o laço **for** como um laço **while**.

for (i=0; i<=n; ) {

<Instrução>

<Incremento/decremento/contador>

}

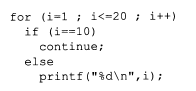
1. Escreva genericamente o laço **while** como um laço **for**.
2. Qual a diferença de execução da instrução **break** quando presente em um laço *for* e em um *while?*

A instrução **break** termina o laço, continuando o programa na próxima instrução a seguir ao laço.

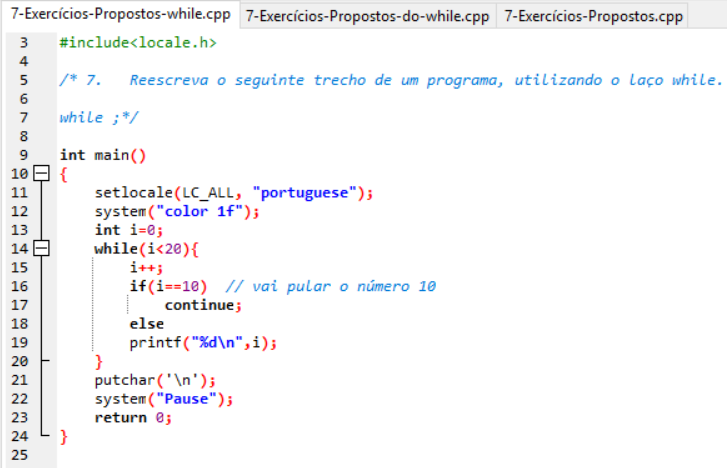
1. Qual a diferença de execução da instrução **continue** quando presente em um *for* e em um *while*?

A instrução **continue** faz com que uma determinada iteração seja terminada, passando-se automaticamente a próxima iteração.

1. Reescreva o seguinte trecho de um programa, utilizando o laço *while*.

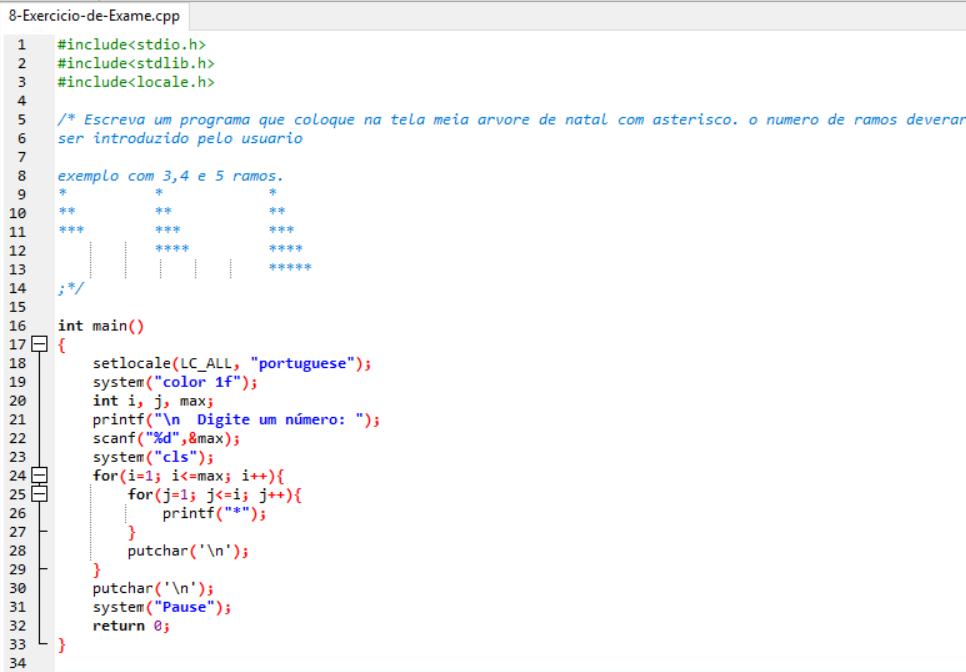


....................................................................................................................

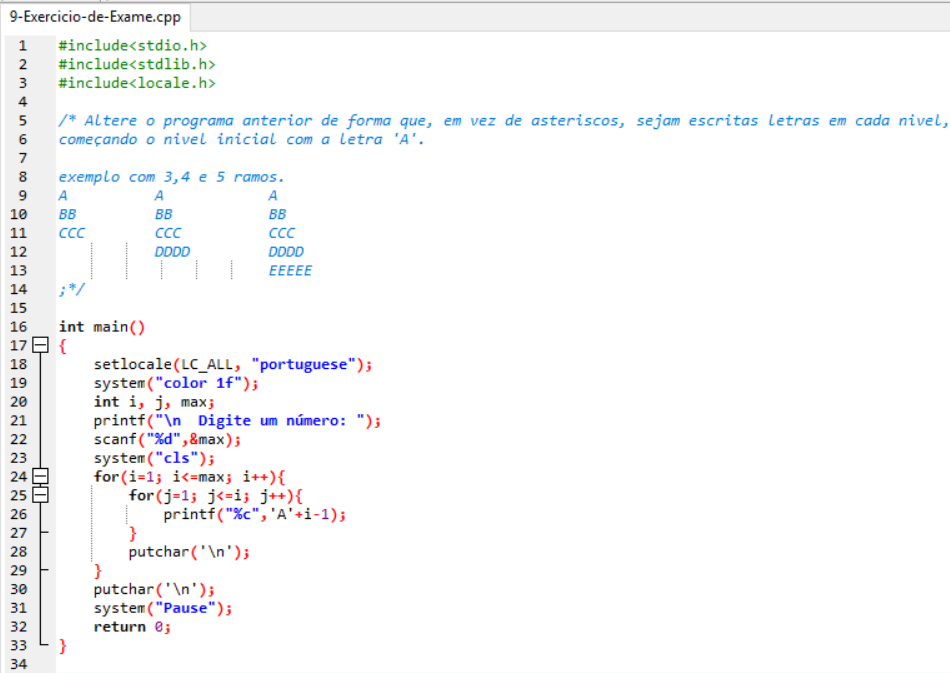
1. **[Exercício de Exame]**

Escreva um programa que coloque na tela meia arvore de natal com asterisco. O número de ramos deverá ser introduzido pelo usuário exemplo com 3,4 e 5 ramos.

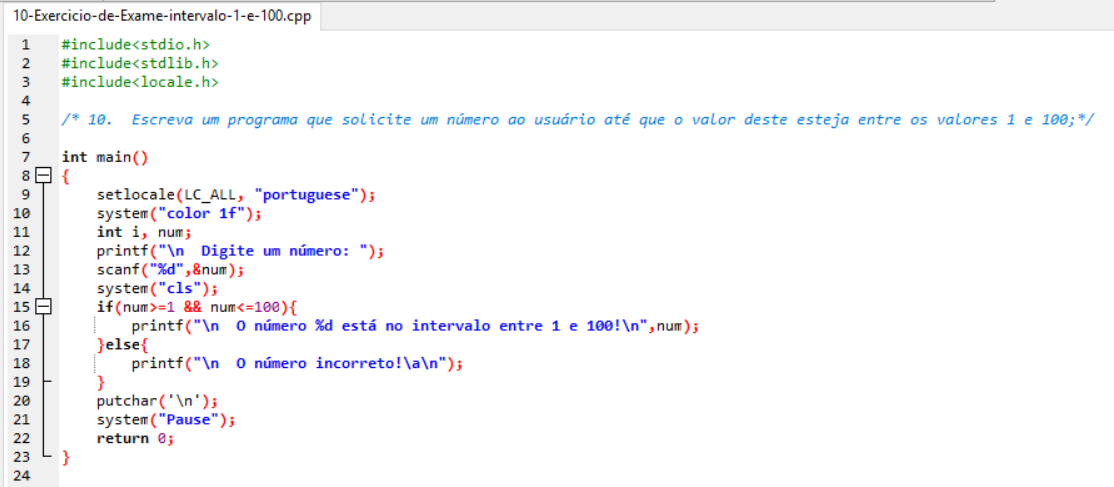


1. **[Exercício de Exame]**

Altere o programa anterior de forma que, em vez de asteriscos, sejam escritas letras em cada nível, começando o nível inicial com a letra ‘A’.



1. Escreva um programa que solicite um número ao usuário até que o valor deste esteja entre os valores 1 e 100.



1. **[Exercício de Exame]**

Escreva um programa em C que escreva na tela toda a tabela ASCII (0..255 chars), escrevendo em cada linha o código ASCII e o caractere correspondente.

Exemplo:

...

65 --> A

66 --> B

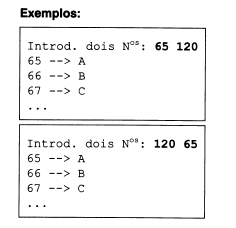
67 --> C

...

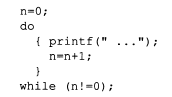
A fim de que o usuário possa ver todos os caracteres escritos, a tela deve ser parada de 20 em 20 linhas, até que o usuário pressione a tecla ‘c’ ou ‘C’ seguida de <ENTER> para continuar a mostra os próximos 20 caracteres (20 linhas).

1. **[Exercício de Exame]**

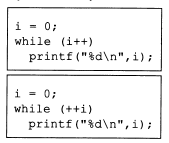
Escreva um programa completo em C que solicite ao usuário dois números inteiros entre 0 e 255 e escreva na tela todos os caracteres da tabela ASCII cujos códigos variem entre os dois números introduzidos, escrevendo em cada linha o código ASCII e o caractere correspondente.



1. Qual a diferença (se existir) entre as duas seguintes instruções: = ++i e x=i++? Dê exemplos.
2. Se uma variável X se encontra isolada, ++X e X++ são equivalentes?
3. Pode-se utilizar os operadores ++ e -- em constantes?
4. Qual a diferença entre a=-2 e a-=?
5. Qual o resultado do seguinte programa:



1. Qual a diferença entre os dois trechos de código:



1. O que faz o seguinte código:



1. Quantas vezes são executadas as instruções dos seguintes laços:
   1. for (i=1; i<=20; i++)
   2. for (i=1; i<=20; i+=2)
   3. for (i=-20; i<=20; i++)
   4. for (i=1; i<=10; i++)
   5. for (j=1; j<=5; j++)
2. Preencha os espaços em branco com o valor das respectivas colunas, depois de executada a instrução a sua esquerda.

