



Instruções:

- LEIA TODO O DOCUMENTO ANTES DE RESPONDER À AVALIAÇÃO.
- A avaliação ocorre em modelo de atividade assíncrona.
- O prazo para o término da avaliação é de 200 (duzentos) minutos, ou 3 (três) horas e 20 (vinte) minutos. Haverá uma pequena tolerância de 10 minutos para a entrega. Respostas não entregues dentro deste prazo serão desconsideradas.
- As respostas devem ser entregues pelo aluno em arquivo programado e compilável, intitulado "**ESCREVA_O_SEU_NOME.cpp**". A não entrega em arquivo ".cpp" ou a não compilação do arquivo acarretará na não correção por parte do professor e, conseqüentemente, na atribuição de nota 0 (zero).
- Após a entrega das respostas por parte do aluno, o professor tem a prerrogativa de arguir o aluno a respeito da solução encontrada para as suas respostas, em modelo de atividade síncrona.
- Não serão aceitos comandos com funções nem operadores ainda não estudados no Curso.

Descrição:

Considerando o conjunto dos números naturais, o **MMC** entre dois números é o **Menor Múltiplo Comum** a esses dois números. Assim sendo, o MMC entre 12 e 9 vale 36, pois 36 é o menor múltiplo comum a 9 (9, 18, **36**, 45, ...) e a 12 (12, 24, **36**, 48, ...). Vamos supor que você precisa escrever um programa a ser compartilhado por uma equipe de programadores na linguagem C++ capaz de calcular o MMC entre dois números. **Diante da necessidade de compartilhamento do código, você precisa escrever cada função com o seu tipo de retorno, com o seu identificador e com os tipos dos seus parâmetros exatamente conforme pedido a você.** Para cada nova função, você pode – e deve – chamar outra ou outras funções já programadas anteriormente por você nesta avaliação. De tal maneira, escreva:

1) **(2,0 pontos)** Uma função para receber dois números e devolve-os ordenados de modo crescente.

Tipo de retorno: void
Nome da função: menorMaior
Tipo do primeiro parâmetro: int
Tipo do segundo parâmetro: int

2) **(1,0 ponto)** Uma função para receber um número e retornar a quantidade de divisores que este número possui.

Tipo de retorno: int
Nome da função: quantidadeDeDivisores
Tipo do parâmetro: int

3) **(1,0 ponto)** Uma função para receber um número e retornar a informação sobre a primalidade deste número, ou seja, indicar se é verdadeiro que o número é primo, ou indicar se é falso que o número é primo. Um número é considerado primo quando ele possui apenas dois divisores: o número 1 e ele próprio.

Tipo de retorno: int
Nome da função: primo
Tipo do parâmetro: int

4) **(2,0 pontos)** Uma função para receber um número e retornar o seu menor divisor primo. Um divisor primo de um número x é um número primo que divide x. Essa função deve considerar a seguinte exceção: o número 1 não possui divisores primos; portanto, excepcionalmente para o número 1, essa função deve retornar o próprio número 1.

Tipo de retorno: int
Nome da função: menorDivisorPrimo



Tipo do parâmetro: int

5) **(4,0 pontos)** Uma função para receber dois números inteiros e positivos quaisquer e retornar o MMC entre eles. Um algoritmo para o cálculo do MMC entre dois números e um exemplo de sua execução seguem após a especificação da função.

Tipo de retorno: int
Nome da função: mmc
Tipo do primeiro parâmetro: int
Tipo do segundo parâmetro: int

Algoritmo para calcular o MMC entre dois números A e B (válido para $A \leq B$):

PASSO 1: Faz-se RESULTADO igual a 1.

PASSO 2: Se A e B são ambos iguais a 1, vá para o PASSO 6.

PASSO 3: Escolhe-se como divisor o menor número entre o menor divisor primo de A e o menor divisor primo de B.

PASSO 4: O parâmetro A passa a valer o resultado da divisão de A pelo divisor escolhido no PASSO 3, somente se A for divisível por aquele divisor. O parâmetro B passa a valer o resultado da divisão de B pelo divisor escolhido no PASSO 3, somente se B for divisível por aquele divisor.

PASSO 5: O valor de RESULTADO passa a ser o resultado da multiplicação de RESULTADO pelo divisor escolhido no PASSO 3 e o algoritmo volta ao PASSO 2.

PASSO 6: Retorne RESULTADO.

Exemplo de execução do algoritmo conforme descrito acima para A = 9 e B = 12:

Passo	Execução
1	RESULTADO = 1
2	Seguir para o PASSO 3
3	DIVISOR = 2 (<i>menor entre 3 e 2</i>)
4	A = 9 B = 6 ($12/2 = 6$)
5	RESULTADO = 2 ($1 \times 2 = 2$)
2	Seguir para o PASSO 3
3	DIVISOR = 2 (<i>menor entre 3 e 2</i>)
4	A = 9 B = 3 ($6/2 = 3$)
5	RESULTADO = 4 ($2 \times 2 = 4$)
2	Seguir para o PASSO 3
3	DIVISOR = 3 (<i>menor entre 3 e 3</i>)
4	A = 3 ($9/3 = 3$) B = 1 ($3/3 = 1$)
5	RESULTADO = 12 ($4 \times 3 = 12$)
2	Seguir para o PASSO 3
3	DIVISOR = 3 (como 1 não possui divisores primos, ele não deve ser considerado)
4	A = 1 ($3/3 = 1$) B = 1
5	RESULTADO = 36
2	Seguir para o PASSO 6
6	Retorne 36

ATENÇÃO. Você não precisa entregar a função `main` programada, mas é altamente recomendado que você a programe para testar as funções implementadas. Seu código será avaliado da seguinte maneira: suas funções serão usadas por um programa já existente. Mais uma vez, é preciso que suas funções sejam compiladas. Cada



função que funcionar corretamente acrescenta a respectiva pontuação à nota da sua avaliação. Segue imagem de uma execução do programa teste:

```
"C:\Users\Rodrigo\Documents\Rodrigo - Minhas pastas\Atividades profissionais\CEFET\Aulas\Superior\Programação Estruturada\2021.1\Prova P1\_P...
Manipulacao de numeros inteiros...
Entre com o 1o. numero: 12
Entre com o 2o. numero: 9

Saida...
Menor numero entre 9 e 12: 9
Maior numero entre 9 e 12: 12
Quantidade de divisores de 12: 6
Quantidade de divisores de 9: 3
Primalidade de 12: FALSA
Primalidade de 9: FALSA
Menor divisor primo de 12: 2
Menor divisor primo de 9: 3
Menor Multiplo Comum entre 12 e 9: 36
```

Boa avaliação.