



Instruções:

- LEIA TODO O DOCUMENTO ANTES DE RESPONDER À AVALIAÇÃO.
- O prazo para o término da avaliação é de 120 (cento e vinte) minutos, ou 2 (duas) horas.
- As respostas podem ser entregues a lápis, mas somente as respostas escritas a caneta (preta ou azul) terão direito à revisão.
- Você pode usar folhas em branco como rascunho.
- As respostas devem ser escritas nos versos das folhas desta prova.

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. Já existe uma solução programada e compartilhada que computa a quantidade de divisores de um número inteiro, considerando valores não negativos. Essa solução segue abaixo e pode ser usada por você:

```
int quantidadeDeDivisores (int numero)
{
    return quantidadeDeDivisoresRec(numero, numero);
}

int quantidadeDeDivisoresRec (int numero, int dividendo)
{
    switch(dividendo)
    {
        case 0: return 0;
        case 1: return 1;
        default: return (numero % dividendo == 0) ? 1 + quantidadeDeDivisoresRec(numero, dividendo - 1):
            quantidadeDeDivisoresRec(numero, dividendo - 1);
    }
}
```

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-los ordenados de modo crescente.

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

- 2) **(2,0 pontos)** 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

- 3) **(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: menoDivisorPrimo Tipo do parâmetro: int



4) **(4,0 pontos)** Uma função para receber dois números 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:

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, desde que esse não seja o número 1. Quando um desses menores divisores primos for igual a 1, escolhemos como divisor o outro número em questão.

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 do valor atual 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