

**RECURSÃO**

- **Submissão com data e hora de entrega disponíveis na plataforma da disciplina.**

- **Procedimento para a entrega:**

1. Siga atentamente quanto ao formato da entrada e saída de seu programa, exemplificados no enunciado.
2. Durante a correção, os programas serão submetidos a vários casos de testes, com características variadas.
3. A avaliação considerará o tempo de execução e o percentual de respostas corretas.
4. Eventualmente, serão realizadas entrevistas sobre as práticas para complementar a avaliação.
5. Considere que os dados serão fornecidos pela entrada padrão. Não utilize abertura de arquivos pelo seu programa. Se necessário, utilize o redirecionamento de entrada.
6. Os códigos fonte serão submetidos a uma ferramenta de detecção de plágios em software.
7. Códigos cuja autoria não seja do aluno, com alto nível de similaridade em relação a outros trabalhos, ou que não puder ser explicado, acarretará na perda da nota.
8. Códigos ou funções prontas específicos de algoritmos para solução dos problemas elencados não são aceitos.
9. Não serão considerados algoritmos parcialmente implementados.

- **Bom trabalho!**

A Aliança dos Reinos e o Cristal do Equilíbrio

Em uma terra encantada, três clãs protegem runas mágicas que selam o poder de uma antiga criatura chamada **Kaordus**, cuja libertação poderia trazer caos ao mundo. Cada clã possui um conjunto específico de runas que devem ser organizadas em padrões nos altares de proteção para reforçar o selo mágico.

Os clãs possuem as seguintes quantidades de runas:

- O **Clã do Vento** possui X runas;
- O **Clã da Terra** possui Y runas;
- O **Clã do Fogo** possui Z runas.

Para manter o equilíbrio entre os poderes mágicos, os líderes dos clãs decidiram que cada padrão nos altares deve utilizar a mesma quantidade de runas de cada clã. Entretanto, eles enfrentam dois grandes desafios:

1. **Quantos padrões podem ser criados?** Para responder a essa pergunta, é necessário determinar o **Máximo Divisor Comum (MDC)** entre X , Y e Z . Isso garantirá que todas as runas sejam utilizadas sem sobras.
2. **Quanto tempo até o alinhamento perfeito das runas?** Além disso, os clãs querem saber o menor período necessário para que as quantidades de runas mineradas por dia (valores de X , Y , e Z) se alinhem perfeitamente, ou seja, atinjam um valor comum. Para isso, é preciso calcular o **Mínimo Múltiplo Comum (MMC)** entre as quantidades de runas mineradas por cada clã em um dia.

Sua missão é implementar um programa que ajude os líderes a resolver esses problemas **de forma recursiva** (outras formas não serão aceitas) e, assim, garantir que o selo de Kaordus permaneça intacto. Use suas habilidades de programação para calcular:

- O **MDC** entre X , Y , e Z para determinar o número máximo de padrões mágicos que podem ser formados. O **MDC** pode ser calculado da seguinte forma:

$$MDC(A, B) = \begin{cases} A & , \text{ se } B == 0 \\ MDC(B, MOD(A, B)) & , \text{ caso contrário} \end{cases}$$

- O **MMC** entre X , Y , e Z para calcular o menor período em que as runas mineradas estarão perfeitamente alinhadas. O **MMC** pode ser calculado da seguinte forma:

$$MMC(A, B, C) = \begin{cases} C & , \text{ se } MOD(C, A) == 0 \text{ e } MOD(C, B) == 0 \\ MMC(A, B, C + MAX(A, B)) & , \text{ caso contrário} \end{cases}$$

Tanto o MDC, quanto o MMC utilizam a função MOD que corresponde a função de cálculo do resto da divisão que pode ser calculado de forma recursiva, seguindo a seguinte equação:

$$MOD(A, B) = \begin{cases} 0 & , \text{ se } A == B \\ A & , \text{ se } A < B \\ MOD(A - B, B) & , \text{ caso contrário} \end{cases}$$

A função MAX retorna o máximo entre dois números e pode ser implementada da forma tradicional.

Ajude os clãs a preservar a paz e proteger o mundo de Kaordus!

Considerações

O código-fonte deve conter três funções recursivas para efetuar as operações desejadas e deve ser modularizado corretamente conforme os arquivos de protótipo fornecidos. Especialmente nesta aula prática, **não** será permitido criar outras funções ou procedimentos. Também **não** será permitido criar variáveis adicionais no *main*.

- Não altere o nome dos arquivos.
- O arquivo .zip deve conter na sua raiz somente os arquivos-fonte.
- Há alguns casos de teste e você terá acesso (entrada e saída) a alguns deles para realizar os seus testes.

Especificação da Entrada e da saída

Em cada caso de teste de entrada haverá o valor de três inteiros: X , Y e Z . A saída é o resultado da operação de MDC seguida do MMC.

Entrada	Saída
36 60 84	MDC = 12 MMC = 1260

Diretivas de Compilação

```
$ gcc -c recursao.c -Wall
$ gcc -c pratica.c -Wall
$ gcc recursao.o pratica.o -o exe
```

Avaliação de *leaks* de memória

Uma forma de avaliar se não há *leaks* de memória é usando a ferramenta valgrind. Um exemplo de uso é:

```
gcc -g -o exe *.c -Wall; valgrind -leak-check=yes -s ./exe < casoteste.in
```

Espera-se uma saída com o fim semelhante a:

```
==38409== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

Para instalar no Linux, basta usar: `sudo apt install valgrind`.