

Objetivo

O objetivo desta atividade é que o aluno implemente o Algoritmo Euclidiano (o simples, não o estendido) visto em sala de aula. Podemos executar este algoritmo manualmente construindo uma tabela com apenas uma coluna que se inicia com os valores dos dois números fornecidos como entrada e, em seguida, contém os valores dos restos das sucessivas divisões realizadas pelo algoritmo. O valor presente na última linha será zero (o algoritmo termina quando produz um resto zero) e o valor presente na penúltima linha será o MDC dos dois números. Por exemplo, o cálculo do MDC entre 1234 e 54 irá gerar a seguinte tabela:

R
1234
54
46
8
6
2
0

O objetivo do programa que será realizado é ler pares de números inteiros positivos, executar o Algoritmo Euclidiano e imprimir na tela para o usuário a réplica das tabelas geradas, como a tabela acima.

Entrada

Inicialmente, o programa deverá ler um número inteiro n . Este número irá indicar quantos pares de números inteiros positivos o programa deverá ler na sequência. Isto é, se $n = 6$, o programa deverá ler, em seguida, seis pares de números inteiros positivos. Cada par de números será lido de uma vez, estando os dois números do par separados por uma vírgula.

Abaixo, são apresentados dois exemplos de possíveis entrada para o programa.

Saída

Para cada par de inteiros lido, o programa deverá imprimir uma réplica da tabela gerada pelo Algoritmo Euclidiano para o cálculo do MDC dos dois números do par. Nesta réplica, nas duas primeiras linhas devem ser impressos os dois números do par fornecido como entrada e, em seguida, em cada linha deve ser impresso o valor do último resto produzido pela sequência de divisões. Ao final da réplica de uma tabela, o programa deverá imprimir uma linha com apenas três traços: ---.

Após a entrada de cada par de inteiros, o programa deve imprimir a réplica da tabela correspondente. Isto é, **o funcionamento do programa é “lê um par, imprime uma tabela, lê outro par, imprime outra tabela...”**. O programa não deve ler todos os pares da entrada e só então começar a imprimir as tabelas, pois isto exigiria um uso de memória (e, conseqüentemente, de variáveis) muito maior.

Abaixo, são apresentados dois exemplos de saídas para o programa. Estas são justamente as saídas que devem ser produzidas caso o programa receba as entradas fornecida no exemplo.

Observações Importantes

- Fique atento para que o seu arquivo com o código siga a convenção de nomenclatura solicitada: `<DRE>.py`. Ex: `106329638.py`
- Não utilize caracteres acentuados no seu código, nem mesmo nos comentários. Eles muitas vezes geram erros de execução do programa. Com a grande quantidade de alunos, eu não tenho como ficar analisando cada caso individualmente para descobrir se o erro de execução foi apenas por um caractere acentuado ou por um erro real de programação do aluno. Evite problemas, não use acentos, til, cedilha, etc.
- O seu programa deve realizar a entrada dos dados exatamente no formato descrito no enunciado e deve imprimir a saída também exatamente no formato descrito no enunciado.
- O seu programa não deve imprimir **nada** além do que é pedido no enunciado. Desta forma, mensagens de “ajuda” para o usuário, como “Digite um par de números” não devem ser impressas. Assuma sempre que o usuário do programa terá o mesmo conhecimento que você sobre o enunciado da atividade, isto é, ele sabe em qual formato deve fornecer a entrada dos dados.
- Você deve assumir também que o usuário sempre irá respeitar o enunciado da atividade, isto é, ele nunca digitará um número negativo quando o enunciado pede um número positivo, etc. Desta forma, não é necessário “gastar” código fazendo tratamento de possíveis erros de entrada por parte do usuário.

Exemplo 1

Este exemplo é o mesmo descrito no início do enunciado.

Entrada

1
1234,54

Saída

1234
54
46
8
6
2
0

Exemplo 2

Entrada

3
484,22
760,84
10026,168

Saída

484
22
0

760
84
4
0

10026
168
114
54
6
0
