Atividade de Laboratório **2.1**

Números Inteiros e Criptografia - Prof. Luis Menasché Schechter

Objetivo

O objetivo desta atividade é que o aluno implemente o algoritmo "ingênuo" de divisão visto em sala de aula. Podemos executar este algoritmo manualmente construindo uma tabela em que na primeira coluna colocamos os sucessivos valores da variável que irá armazenar o quociente e na segunda coluna colocamos os sucessivos valores da variável que irá armazenar o resto. Os valores presentes na última linha da tabela são o quociente e o resto da divisão. Por exemplo, realizando a divisão de 120 por 14 com o algoritmo "ingênuo", construímos a seguinte tabela:

Q	R
0	120
1	106
2	92
3	78
4	64
5	50
6	36
7	22
8	8

O objetivo do programa que será realizado é ler pares de números inteiros positivos, realizar a divisão do primeiro número do par pelo segundo através do algoritmo "ingênuo" 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 "ingênuo" para a divisão entre os dois números do par. Nesta réplica, em cada linha deve ser impresso o valor atual da variável do quociente seguido do valor atual da variável do resto, sendo estes dois valores separados por um espaço. 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, consequentemente, 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
120,14
```

Saída

```
0 120
1 106
2 92
3 78
4 64
5 50
6 36
7 22
8 8
---
```

Exemplo 2

Entrada

```
3
5,2
17,3
42,4
```

Saída

```
0 5
1 3
2 1
---
0 17
1 14
2 11
3 8
4 5
5 2
---
0 42
1 38
2 34
3 30
4 26
5 22
6 18
7 14
8 10
9 6
10 2
---
```