Atividade de Laboratório **8.1**

Números Inteiros e Criptografia - Prof. Luis Menasché Schechter

Objetivo

O objetivo desta atividade é que o aluno implemente o Algoritmo de Potenciação Modular auxiliado pelo Teorema de Fermat como visto em sala de aula. Como vimos, o Teorema de Fermat pode auxiliar o cálculo de potências quando o módulo é primo, reduzindo significativamente o tamanho do expoente fornecido para o algoritmo de potenciação. Por exemplo, com o módulo 31, que é primo, temos $3^{1057} \equiv 3^7 \pmod{31}$. Assim, para calcularmos o valor de 3^{1057}, podemos utilizar o algoritmo de potenciação com expoente 7 ao invés de 1057:

R	A	E	E é ímpar?
1	3	7	S
3	9	3	S
27	19	1	S
17	20	0	N

A partir da última linha da tabela, se obtém o resultado: $3^{1057} \equiv 3^7 \equiv 17 \pmod{31}$.

O objetivo do programa que será realizado é ler triplas de números inteiros positivos, executar o Algoritmo de Potenciação Modular, considerando o primeiro valor da tripla como a base, o segundo como o expoente e o terceiro como o módulo (que será primo), e imprimir na tela para o usuário os expoentes que o Teorema de Fermat permite utilizar no lugar do expoente originais, seguidos da réplica das tabelas geradas com estes novos expoentes, como a tabela acima.

Entrada

Inicialmente, o programa deverá ler um número inteiro n. Este número irá indicar quantas **triplas** de números inteiros positivos o programa deverá ler na sequência. Isto é, se $n = 6$, o programa deverá ler, em seguida, seis **triplas** de números inteiros positivos.

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

Saída

Para cada tripla lida, considerando o primeiro valor da tripla como a base, o segundo como o expoente e o terceiro como o módulo (que será primo), o programa deverá imprimir o novo expoente que o Teorema de Fermat permite utilizar no lugar do expoente original seguido, na linha abaixo, por uma réplica da tabela gerada pelo Algoritmo de Potenciação Modular com este novo expoente. Após a impressão de uma tabela, o programa deverá imprimir uma linha com apenas três traços: `---`.

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 fornecidas no exemplo.

Exemplo 1

Este exemplo é o mesmo descrito no início do enunciado.

Entrada

```
1
3,1057,31
```

Saída

```
7
1 3 7 S
3 9 3 S
27 19 1 S
17 20 0 N
---
```

Exemplo 2

Entrada

```
3
2,125,7
6,290,101
3,11413,103
```

Saída

```
5
1 2 5 S
2 4 2 N
2 2 1 S
4 4 0 N
---
90
1 6 90 N
1 36 45 S
36 84 22 N
36 87 11 S
1 95 5 S
95 36 2 N
95 84 1 S
1 87 0 N
---
91
1 3 91 S
3 9 45 S
27 81 22 N
27 72 11 S
90 34 5 S
73 23 2 N
73 14 1 S
95 93 0 N
---
```