Atividade de Laboratório **9.2**

Números Inteiros e Criptografia - Prof. Luis Menasché Schechter

Objetivo

O objetivo desta atividade é que o aluno implemente o Algoritmo Chinês do Resto para simplificar o cálculo de potências modulares em alguns casos em que o módulo é composto, como visto em sala de aula. Por exemplo, para calcular $2^{6754} \pmod{1155}$, começamos fatorando $1155 = 3 * 5 * 7 * 11$. Em seguida, considerando que $3, 5, 7$ e 11 são primos, calculamos, com o auxílio do Teorema de Fermat, 2^{6754} módulo 3, módulo 5, módulo 7 e módulo 11, obtendo, respectivamente, 1, 4, 2 e 5. Construímos então o sistema

$$\begin{cases} x \equiv 1 \pmod{3} \\ x \equiv 4 \pmod{5} \\ x \equiv 2 \pmod{7} \\ x \equiv 5 \pmod{11} \end{cases}$$

e o resolvemos com o Algoritmo Chinês do Resto, obtendo a solução $2^{6754} \equiv 709 \pmod{1155}$.

O objetivo do programa que será realizado é ler triplas de números inteiros e, considerando o primeiro número como a base, o segundo como o expoente e o terceiro como o módulo, utilizar o Algoritmo Ingênuo de Fatoração, o Teorema de Fermat e o Algoritmo Chinês do Resto para calcular estas potências.

Entrada

Inicialmente, o programa deverá ler um número inteiro n. Este número irá indicar quantas **triplas** de números inteiros o programa deverá ler na sequência. Isto é, se $n = 6$, o programa deverá ler, em seguida, seis **triplas** de números inteiros. Cada tripla será lida de uma vez, estando os três números separados por vírgula.

Abaixo, é apresentado um exemplo de possível entrada para o programa.

Saída

Para cada tripla lida, considerando considerando o primeiro número como a base, o segundo como o expoente e o terceiro como o módulo, o programa deverá imprimir a seguinte sequência de informações:

1. Uma réplica da tabela do Algoritmo Ingênuo de Fatoração (conforme atividade 4.1) aplicado ao módulo;
2. Para cada fator primo da fatoração acima, o programa deve testar se a base é ou não congruente a zero módulo o fator primo em questão (pois, para utilizar o Teorema de Fermat, a base não pode ser congruente a zero).
    2.1. Se for, o programa deve imprimir uma linha apenas com o valor 0;
    2.2. Se não for, o programa deverá imprimir tabela análoga à da atividade 7.2, começando com uma linha com 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 (atividade 6.2) com este novo expoente;
3. O programa deve então resolver o sistema de congruências obtido a partir dos resultados acima utilizando o Algoritmo Chinês do Resto com o mesmo formato de impressão da atividade 9.1:
    3.1. Uma réplica da tabela do Algoritmo Euclidiano Estendido (conforme atividade 3.1) entre os módulos das duas primeiras congruências;

3.2. Em seguida, em uma única linha, separados por um espaço em branco, os valores calculados para α e β;

3.3. Na linha seguinte, separados por um espaço em branco, o valor e o módulo da solução encontrada para as duas primeiras congruências;

3.4. Caso haja mais de duas congruências no sistema dado, o programa deverá então imprimir uma réplica da tabela do Algoritmo Euclidiano Estendido entre o último módulo calculado acima e o módulo da próxima congruência que ainda não foi utilizada;

3.5. O programa então repete os passos 3.2, 3.3 e 3.4 acima até obter a solução final (valor e módulo) para o sistema dado. Após a impressão da solução final, o programa deverá imprimir uma linha com apenas três traços: `---`.

Abaixo, é apresentado um exemplo de saída para o programa. Esta é justamente a saída que deve ser produzida caso o programa receba a entrada fornecida no exemplo.

Exemplo

Entrada

```
3
2,6754,1155
3,3290,14007
5,10043,30305
```

Saída

```
3 1
5 1
7 1
11 1
0
1 2 0 N
2
1 2 2 N
1 4 1 S
4 1 0 N
4
1 2 4 N
1 4 2 N
1 2 1 S
2 4 0 N
4
1 2 4 N
1 4 2 N
1 5 1 S
5 3 0 N
3 - 1 0
5 - 0 1
3 0 1 0
2 1 -1 1
1 1 2 -1
0 2 - -
2 -1
4 15
15 - 1 0
7 - 0 1
1 2 1 -2
0 7 - -
1 -2
79 105
105 - 1 0
11 - 0 1
6 9 1 -9
5 1 -1 10
1 1 2 -19
0 5 - -
2 -19
709 1155
---
3 1
7 1
23 1
29 1
0
2
1 3 2 N
1 2 1 S
2 4 0 N
12
1 3 12 N
1 9 6 N
1 12 3 S
12 6 1 S
3 13 0 N
14
1 3 14 N
1 9 7 S
9 23 3 S
4 7 1 S
28 20 0 N
3 - 1 0
7 - 0 1
3 0 1 0
1 2 -2 1
0 3 - -
-2 1
9 21
21 - 1 0
23 - 0 1
21 0 1 0
2 1 -1 1
1 10 11 -10
0 2 - -
11 -10
72 483
483 - 1 0
29 - 0 1
19 16 1 -16
10 1 -1 17
9 1 2 -33
1 1 -3 50
0 9 - -
-3 50
7800 14007
---
5 1
11 1
19 1
29 1
0
3
1 5 3 S
5 3 1 S
4 9 0 N
17
1 5 17 S
5 6 8 N
5 17 4 N
5 4 2 N
5 16 1 S
4 9 0 N
19
1 5 19 S
5 25 9 S
9 16 4 N
9 24 2 N
9 25 1 S
22 16 0 N
5 - 1 0
11 - 0 1
5 0 1 0
1 2 -2 1
0 5 - -
-2 1
15 55
55 - 1 0
19 - 0 1
17 2 1 -2
2 1 -1 3
1 8 9 -26
0 2 - -
9 -26
840 1045
1045 - 1 0
29 - 0 1
1 36 1 -36
0 29 - -
1 -36
24875 30305
---
```