1 ARITMÉTICA MODULAR
(Nível 1)

As leis da aritmética modular estão entre as melhores armas que dispomos no nosso arsenal da matemática. Nós, os cientistas da computação, freqüentemente usamos essas leis para simplificar contas. Por exemplo, se quisermos calcular o algarismo das unidades:

$$2^4 5^{12} 7^{16} - 2^2 5^{16} 7^{64}$$

Nós somos capazes de fazer isso num piscar de olhos. Para isso, precisamos lidar com equações envolvendo aritmética modular, tarefa que não é muito confortável para muitos de nós.

Dada a sequência de três números inteiros de (x_{min} x x_{max}), y (y_{min} y y_{max}) e n (n_{min} n n_{max}) você deve encontrar os números da tripla (x, y, n) que satisfazem a equação:

$$(x + y) \equiv (x - y) \pmod{n}$$

Por exemplo:

1 x 2,2 y 4,3 n 5

(x + y) mod n	=	(x - y) mod n
(1 + 2) mod 4	3	(1 - 2) mod 4
(1 + 3) mod 3	1	(1 - 3) mod 3
(1 + 4) mod 4	1	(1 - 4) mod 4
(2 + 2) mod 4	0	(2 - 2) mod 4
(2 + 3) mod 3	2	(2 - 3) mod 4
(2 + 4) mod 4	2	(2 - 4) mod 4

Entrada. A primeira linha da entrada oferece o número de casos de teste T (1 T 10). Cada uma das entradas seguintes deve conter a entrada para cada caso de teste. A entrada de cada teste contém 3 pares dos inteiros x_{min} , x_{max} , y_{min} , y_{max} e n_{min} , n_{max}. Você pode limitar em :

- $-1000 < x_{min} < x_{max} < +1000$
- $-1000 < y_{min} < y_{max} < +1000$
- $+1 < n_{min} < n_{max} < +1000$

Saída. Para cada um dos casos de teste que você precisa para imprimir o número do caso de teste. Em seguida, na mesma linha você tem que imprimir o número de triplas (x, y, n) que satisfazem nossa equação modular.

Exemplo de Input
3
1 2 2 4 3 5
-100 100 200 350 1 1000
5 9 10 12 2 9
Exemplo de Output
Caso 1: 6
Caso 2: 318384
Caso 3: 45

2 NÚMERO DE DIVISORES
(Nível 1)

Considere uma sequência de inteiros N, onde,

$N_0 = 1$

$N_j = N_j{-}1 + ND(N_j - 1)$ for $j > 0$

Tal que ND(x) = número de divisores de x.

Assim, os primeiros termos desta sequência são 1 2 4 7 9 12 18 ...
Dado dois inteiros A e B, encontre o número de inteiros da sequencia acima que se situa dentro do intervalo [A, B].

Entrada. É preciso informar o número de casos de teste T (T<1000). Cada caso de teste contém dois inteiros, A e B (1 A B 1000000).

Saída. Para cada caso, a saída do número do processo primeiro, seguido pelo resultado necessário.

Exemplo de Entrada
2
1 100
3000 4000

Exemplo de Saída
Caso 1: 20
Caso 2: 87

3 FIBONACCI E ARITMÉTICA MODULAR
(Nível 1)

Os números de Fibonacci (0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...) são definidos pela relação de recorrência:

$F_0 = 0$
$F_1 = 1$
$F_n = F_{n-1} + F_{n-2}$, para $n > 1$

Implemente um programa que calcule $M_n = F_n \bmod 2^m$ para qualquer par de inteiros n e m, tais que 0 n $(2^{31} – 1)$ e 0 m 20.

Entrada. Consiste em diversas linhas especificando o par n e m.

Saída. Consiste no correspondente M_n por linha.

Exemplo de Entrada
11 7
11 6

Exemplo de Saída
89
25

4 MÁXIMO DIVISOR COMUM
(Nível 1)

Dado o valor de n, você terá que encontrar o valor de M. A definição de M é dada abaixo:

$$M = \sum_{i=1}^{i<n} \sum_{j=i+1}^{j \leq n} mdc(i, j)$$

Naturalmente, $mdc(i, j)$ significa o máximo divisor comum dos inteiros i e j.

Entrada. É preciso informar o número de casos de teste T (T < 100). Na linha sequente, é inserido cada caso de teste contendo um inteiro, N (1 < N < 2048).

Saída. Para cada caso, a saída do número do processo primeiro, seguido pelo resultado necessário M. O valor de M deve caber numa representação sinalizada inteira de 64-bits.

Exemplo de Entrada
3
10
100
200000

Exemplo de Saída
Caso 1: 67
Caso 2: 13015
Caso 3: 143295493160

5 CONGRUÊNCIA LINEAR E DISPOSITIVOS MÓVEIS
(Nível 2)

Implemente um algoritmo para resolver uma congruência linear, por exemplo, como esta:

$$39x \equiv 3 \pmod 7$$

Para isso é preciso implementar basicamente o algoritmo de Euclides para mdc, e calcular o inverso multiplicativo.

Entrada. A entrada é fornecida para cada caso de teste. Um caso de teste consiste em três inteiros **a**, **b**, e **m**, tais que $ax \equiv b \pmod m$.

Saída. Para cada caso de teste da entrada seu programa deve produzir uma única linha, contendo a resolução do sistema de congruência, caso exista.

Exemplo de Entrada
980 68 288
10 14 24

Exemplo de Saída
13
11

6 SISTEMA DE CONGRUÊNCIA E DISPOSITIVOS MÓVEIS
(Nível 2)

Implemente um algoritmo para resolver sistemas de congruência modular com até 5 relações de congruência, por exemplo, como esta:

$$x \equiv 1 \pmod 3$$
$$x \equiv 2 \pmod 5$$
$$x \equiv 3 \pmod 7$$

Isto, naturalmente, envolve a implementação do teorema chinês dos restos.

Entrada. A entrada é dada por um um número N, tal que 1 < N 5, e correspondendo a quantidade de relações de congruência modular. Em seguida, são fornecidos os pares de inteiros $<a_i,m_i>$, tais que $x \equiv a_i \pmod{m_i}$, para cada i N.

Exemplo de Entrada
4
5 7
0 3
1 5
2 4

Exemplo de Saída
306

7 GRANDES NÚMEROS E DISPOSITIVOS MÓVEIS
(Nível 2)

Implemente um algoritmo para calcular os **dois últimos dígitos** de grandes operações, por exemplo, como estas:

$$a^{1266616} - b^{2164545464}$$
$$a^{62622321} + b^{8664}$$
$$a^{9994456} * b^{694454994}$$

Entrada. A entrada é fornecida para cada caso de teste. Um caso de teste consiste em dois pares de inteiros <**a**, exp_a >, <b,exp_b**>** e na operação desejada (-, + ou * -- subtração, soma ou multiplicação), tal que $a^{expa} - b^{expb}$, $a^{expa} + b^{expb}$ ou $a^{expa} * b^{expb}$. Para $2 \leq a, exp_a, b, exp_b \leq (2^{31} - 1)$.

Saída. Para cada um dos casos de teste que você precisa para imprimir o número do caso de teste. Em seguida, na mesma linha você tem que imprimir o respectivo resultado dos dois últimos dígitos do resultado da operação.

Exemplo de Entrada
2
3 41216 7 21664 -
3 41216 7 21664 *

Exemplo de Saída
Caso 1: 20
Caso 2: 21

8 RECUPERANDO DADOS CORROMPIDOS
(Nível 2)

Diversos tipos de dados possuem um digito verificador para garantir a consistência dos dados informados. Exemplos de dados que possuem esta especificação são números de cartão de crédito, certidão de nascimento/casamento/óbito, cnpj, cpf, processos judiciais, registro de encomendas (ect), registro de identidade civil (ric) e título eleitoral.

De forma semelhante a tais dados, livros são identificados por um número ISBN (International Standard Book Number), um código de 10 dígitos x_1, x_2,...x_{10}. Esses números identificam a linguagem, a editora, o número do livro e um dígito de verificação. Esse dígito de verificação é selecionado de forma que

$$\sum_{i=1}^{10} i \cdot x_i \equiv 0 \pmod{11}$$

Implemente um algoritmo para recuperar uma lista de ISBN que possui um digito ausente.

Entrada. É preciso informar o número de casos de teste T (T<100). Cada caso de teste contém uma sequencia com 11 digitos N, contendo um digito ausente, para representar o digito ausente insira o caractere Y.

Saída. Para cada caso, a saída do número do caso primeiro, seguido pelo digito recuperado D.

Exemplo de Entrada
2
020157Y891
032123Y072

Exemplo de Saída
Caso 1: 8
Caso 2: 7

9 RELAÇÕES DE RECORRÊNCIA E MOEDAS
(Nível 2)

Seja P(n,x,y) a propriedade “n centavos podem ser obtidos usando moedas de x-centavos e y-centavos.”

Implemente um programa que calcula o n mínimo a partir dos x e y fornecidos de forma qualquer m inteiro pode ser representado tal que n < m.

Entrada. É preciso informar o número de casos de teste T (T<100). Cada caso de teste contém uma tupla de inteiros, <*n,x,y*> tal que “n centavos podem ser obtidos usando moedas de *x*-centavos e *y*-centavos.

Saída. Para cada caso, a saída será uma tupla de inteiros, <*m, s, t*> sendo *m* o menor valor, *s* o coeficiente para *x*-centavos e *t* o coeficiente para *y*-centavos.
Caso o valor *n* não seja representável, o algoritmo deverá retornar a mensagem:
“*n* não pode ser representado com *x* e *y* centavos”

Exemplo de Entrada
3
12 3 5
13 4 5
7 3 5

Exemplo de Saída
Caso 1: 8 4 0
Caso 2: 12 2 1
Caso 3: “7 não pode ser representado com 3 e 5 centavos”

10 RSA
(Nível 3)

A RSA é um dos algoritmos criptográficos mais utilizado e é considerado como sendo uma das alternativas mais seguras existentes. A sua operação básica é descrita abaixo.

Dois números primos **p** e **q** são escolhidos e **n = pq** é calculado. Em seguida, a função totient **φ(n) = (p−1)(q−1)** e é calculado um número inteiro **e** para encriptar, satisfazendo **1 < e < φ(n)** é escolhido de forma que **mdc(φ(n), e) = 1**.

Finalmente o inteiro **d** é computado para descriptografar a mensagem, sendo **d** o inverso multiplicativo de **e** modulo **φ(n),** ou seja, o inteiro **d** que satisfaz **d * e ≡ 1 mod φ(n)**.

Desse modo, obtemos a chave pública, que consiste no par de inteiros **<n,e>**, e a chave privada são os inteiros **<n,d>**.

Para encriptar uma mensagem **m**, com **0 < m < n**, nós calculamos $\mathbf{c = m^e \bmod n}$, e **c** é a mensagem criptografada. Para descriptografar a mensagem, basta computar $\mathbf{m = c^d \bmod n}$.

Observe que, a fim de fazer isso, a chave secreta deve ser usada; conhecer a chave pública não é suficiente para descriptografar a mensagem. Neste problema a sua tarefa é quebrar a criptografia RSA.

Entrada. A primeira linha da entrada oferece o número de casos de teste T (1 T 10). Cada uma das linhas seguintes deve conter a entrada para cada caso de teste. Um caso de teste consiste de uma linha, que contém três inteiros **N**, **E**, e **C**, tais que $\mathbf{15 \le N \le 10^8}$, **1 ≤ E < N** e **1 ≤ C < N** , de tal forma que **N** e **E** constituem o RSA chave pública descrita acima, e **C** é uma mensagem encriptada com a chave pública.

Saída. Para cada um dos casos de teste que você precisa para imprimir o número do caso de teste. Em seguida, na mesma linha você tem que imprimir a mensagem descriptografada em **ASCII,** de forma seja representável por um inteiro M, 1 ≤ M < N, sendo este a mensagem original.

Exemplo de Entrada
1
91 43 19

Exemplo de Saída
Caso 1: !