

Problema 01) Cometa

O cometa Halley é um dos cometas de menor período do Sistema Solar, completando uma volta em torno do Sol a cada 76 anos. Na última ocasião em que ele se tornou visível do planeta Terra, em 1986, várias agências espaciais enviaram sondas para coletar amostras de sua cauda e assim confirmar teorias sobre sua composição química.

Escreva um programa que, dado um ano particular ($\text{ano} > 0$), determina qual o próximo ano em que o cometa Halley será visível novamente no planeta Terra. Se o ano atual é um ano de passagem do cometa, considere que o cometa já passou nesse ano, ou seja, considere sempre o próximo ano de passagem.

Entrada: um único inteiro A, indicando um ano particular que seja maior do que zero.

Saída: uma única linha, contendo um número inteiro que indica o próximo ano em que o cometa Halley será visível novamente do planeta Terra após o Ano A.

Exemplo

Entrada	Saída
2010	2062

Problema 02) Escala numérica

João tem um irmão mais novo, chamado José, que começou a ir à escola e já está tendo problemas com números. Para ajudá-lo a “pegar o jeito” com a escala numérica, sua professora escreve três números de três dígitos cada, e pede a José para comparar esses números.

Entretanto, em vez de interpretá-los com o dígito mais significativo à esquerda, ele deve interpretá-lo com o dígito mais significativo à direita. Ele tem que dizer à professora qual o maior dos três números.

Escreva um programa que verificará as respostas de José.

Entrada: uma única linha com três números de três dígitos cada – A, B e C – que devem ser maiores que 99 (ou seja, o dígito mais significativo será maior ou igual a 1).

Saída: uma linha com o maior dos números na entrada, comparados como descrito no enunciado da tarefa. O número deve ser escrito invertido, para mostrar a José como ele deve lê-lo (obs. eliminar zeros à esquerda).

Exemplo

Entrada	Saída
123 109 456	901
100 100 100	1

Problema 03) Overflow

Os computadores foram inventados para realizar cálculos muito rapidamente, e atendem a esse requisito de maneira extraordinária. Porém, nem toda “conta” pode ser feita num computador, pois ele não consegue representar todos os números dentro de sua memória.

Em um computador pessoal atual, por exemplo, o maior inteiro que é possível representar em sua memória é 4.294.967.295 ($2^{32}-1$). Caso alguma “conta” executada pelo computador dê um resultado acima desse número, ocorrerá o que chamamos de overflow, que é quando o computador faz uma “conta” e o resultado não pode ser representado, por ser maior do que o valor máximo permitido (em inglês overflow significa transbordar).

Por exemplo, se um computador só pode representar números menores do que 1023 ($2^{10}-1$) e mandarmos ele executar a conta $1022 + 5$, vai ocorrer um overflow.

Dados o maior número que um computador consegue representar e uma expressão de soma e/ou multiplicação entre três inteiros positivos, determine se ocorrerá, ou não, overflow. A ordem de precedência de operadores da linguagem C deve ser naturalmente aplicada.

Entrada: a primeira linha da entrada contém um inteiro N ($N > 0$), que determina o maior número que o computador consegue representar, segundo o intervalo $[-2^N, 2^N-1]$. A segunda linha contém um inteiro P , seguido de: um espaço em branco, um caractere C (que pode ser '+' ou '*', representando os operadores de adição e de multiplicação, respectivamente), outro espaço em branco, outro número inteiro Q , outro espaço em branco, outro operador (que pode ser '+' ou '*'), outro espaço em branco e, finalmente, outro número inteiro R .

Saída: uma linha contendo o resultado da operação, seguido de espaço em branco e das palavras em letras maiúsculas 'OVERFLOW' ou 'OK', se o resultado excedeu ou não o resultado da operação, respectivamente.

Exemplo

Entrada	Saída
5 26 + 3 * 2	32 OVERFLOW
4 12 * 0 + -11	-11 OK

Problema 04) Capicua/Palíndromo

Alan Turing está aprendendo a decompor um número em unidades, dezenas, centenas, etc., e está com grandes dificuldades. Sua professora, preocupada com o rendimento de Alan Turing decidiu ensiná-lo uma brincadeira:

Alan Turing deve pegar um número com quatro algarismos, e verificar se o reverso deste número é ele próprio. Se for, Alan Turing deve responder sim, do contrário deve responder não.

Em verdade, a professora de Alan Turing está ensinando quando um número é chamado de capicua (também conhecido por palíndromo, quando aplicado a palavras). Um número é capicua quando seu reverso é ele próprio. Implemente este jogo divertido.

Entrada: A primeira linha da entrada contém um inteiro N ($N > 0$) representando a quantidade de números inteiros que Alan Turing deve responder sim (capicua) ou não. Cada uma das N linhas seguintes será composta por um inteiro positivo de dois a dez algarismos (zeros à esquerda serão desconsiderados).

Observação: O seu programa não poderá decompor os números na entrada, ou seja, não poderá ler cada número como caracteres individuais que formam o número: deve ler cada linha como um único número.

Saída: A saída consiste de N linhas, cada uma o caractere 'S' se o número for capicua, ou o caractere 'N' caso o número não seja capicua, seguido de um espaço em branco e de uma sequência de letras maiúsculas (de A até J) correspondentes ao número (por exemplo, se o número for 10389, então a sequência de letras será BADJI, pois B corresponde a posição 1, A corresponde a posição 0, D corresponde a posição 3, e assim por diante).

Exemplo

Entrada	Saída
2	N BCDEF
12345	S BCFAFCB
1250521	

Problema 05) Sistemas de numeração

A capacidade do ser humano em calcular quantidades nos mais variados modos foi um dos fatores que possibilitaram o desenvolvimento da Matemática, da Lógica e da Computação. Nos primórdios da Matemática e da Álgebra, utilizavam-se os dedos das mãos para efetuar cálculos. Por volta do século III a.C., o matemático indiano Pingala inventou o sistema de numeração binário, que ainda hoje é utilizado no processamento de todos computadores digitais: o sistema estabelece que sequências específicas de uns e zeros pode representar qualquer número, letra, imagem, etc. Entretanto, a Computação está evoluindo rapidamente e recentemente a SBC (Sociedade Brasileira de Computação) inventou um computador com as bases 3, 4 e 5. A SBC contratou você para fazer um programa que receba um número inteiro positivo, na base decimal, e converta-o para as bases 3, 4 e 5, utilizando divisões sucessivas.

Observação: Considere que os símbolos utilizados para representar as quantidades ZERO, UM, DOIS, TRÊS e QUATRO, nas bases 3, 4 e 5 são, respectivamente, 0, 1, 2, 3 e 4.

Entrada: A primeira linha contém o número de inteiros N ($N > 1$) referente a quantidade de números que devem ser convertidos. A segunda linha contém n N números inteiros positivos n_i , cada um representando um número decimal (base 10).

Saída: uma sequência de N linhas, tal que em cada linha terá as representações do número n_i , em base 3, 4 e 5, separadas por um espaço em branco.

Exemplo

Entrada	Saída
6	1 1 1
1 2 3 4 5 10	2 2 2
	10 3 3
	11 10 4
	12 11 10
	101 22 20

Problema 06) Envelopes

Kurt é um garoto muito esperto, que gosta de promoções e sorteios. Como já participou de muitas promoções da forma “para participar, envie n rótulos de produtos ...”, Kurt tem o hábito de guardar o rótulo de todos os produtos que compra. Dessa forma, sempre que uma empresa faz uma promoção, ele já tem “um monte” de rótulos para mandar.

A SBC (Super Balas e Caramelos) está fazendo uma nova promoção, e, como era de se esperar, Kurt quer participar desta promoção. Nela é preciso enviar um envelope contendo dois rótulos de cada tipo de bala que a SBC produz.

Por exemplo, se a SBC produz 3 tipos de balas (A, B, C) e uma pessoa tem 7 rótulos de A, 8 de B e 11 de C, ela pode enviar no máximo três envelopes, já que a quantidade do rótulo C é 7 e essa quantidade compõe apenas três conjuntos de dois rótulos.

Como você é o melhor amigo de Kurt, ele pediu sua ajuda para fazer o cálculo da quantidade máxima de envelopes que ele pode enviar, de modo que ele compre o número exato de envelopes. Você deve escrever um programa que, a partir da lista de rótulos de Kurt, calcula o número máximo de envelopes válidos que ele pode enviar.

Entrada: A primeira linha contém dois números inteiros N e K representando, respectivamente, a quantidade de rótulos de balas que Kurt possui e o número de tipos diferentes de bala que a SBC produz. A segunda linha contém N números inteiros n_i , $1 \leq n_i \leq K$, cada um representando um rótulo de bala que Kurt possui (ou seja, os tipos de balas são identificados por inteiros de 1 a K).

Saída: Uma única linha com o número máximo de envelopes válidos que Kurt pode enviar.

Exemplo

Entrada	Saída
10 2 1 1 1 1 1 2 2 2 2 2	2
20 5 1 2 3 4 1 2 3 4 1 2 3 4 5 1 2 3 4 5 4 4	1

Problema 07) Sapos

Sebastião Bueno Coelho, apelidado de SBC pelos familiares e amigos, passou as férias de final de ano no sítio de seus avós. Durante sua estadia, uma das suas atividades prediletas era nadar no rio que havia no fundo da casa. Uma das características do rio que mais impressionava SBC era um belo caminho, feito inteiramente com pedras brancas. Há muito tempo, o avô de SBC notara que os habitantes do sítio atravessavam o rio com grande frequência e, por isso, construiu um caminho no rio com pedras posicionadas em linha reta; ao fazê-lo, tomou muito cuidado para que o espaçamento entre o centro das pedras fosse de exatamente um metro. Hoje em dia, a única utilidade do caminho é servir de diversão para os sapos que vivem no rio, que pulam de uma pedra a outra agitadamente.

Um certo dia, enquanto descansava e nadava nas águas, SBC assistiu atentamente às acrobacias dos bichos e notou que cada sapo sempre pulava (zero, uma ou mais vezes) uma quantidade fixa de metros. SBC sabe que você participa da OBI (Olimpíada Brasileira de Informática) todos os anos e, chegando na escola, resolveu desafiar-lhe com o seguinte problema:

“Dado o número de pedras no rio, o número de sapos, a pedra inicial sobre a qual cada sapo está (cada pedra é identificada por sua posição na sequência de pedras – 1, 2, 3, ...) e a distância que cada sapo pula, determinar as posições onde pode existir um sapo depois que SBC chega no rio”.

Entrada: A primeira linha da entrada contém dois inteiros N ($N \geq 1$) e M ($M \geq 1$) representando, respectivamente, o número de pedras no rio e o número de sapos. Cada uma das M linhas seguintes possui dois inteiros P e D representando, respectivamente, a posição inicial de um sapo e a distância fixa de cada pulo que ele pode dar.

Saída: A saída contém uma única linha com os números das pedras (separados por um espaço em branco) em que há a possibilidade de haver um sapo. Por exemplo se existem sete pedras, e nas pedras três há essa possibilidade, então a saída será “3 5”.

Exemplo

Entrada	Saída
5 2 3 2 4 4	1 3 4 5
8 3 3 3 2 2 6 2	2 3 4 6 8

Problema 08) Números primos

No livro *A música dos números primos*, de Marcus du Saboy (2007, Editora Zahar, 471 páginas), o autor mostra que o mistério dos números primos passou a ser considerado o maior problema matemático de todos os tempos. Em meados do século XIX, o alemão Bernhard Riemann (1826 – 1866) formulou uma hipótese: é possível uma harmonia entre esses números primos, à semelhança da harmonia musical. A partir de então, as mentes mais ambiciosas da Matemática embarcaram nessa procura que parece não ter fim. Atualmente, estipulou-se o prêmio de um milhão de dólares para quem provar a hipótese. O livro relata esse verdadeiro Santo Graal da Matemática, com casos interessantes e retratos pitorescos dos personagens que, desde Euclides, se envolveram nesse estranho mistério.

Você deverá pesquisar e implementar um algoritmo que seja capaz de identificar se um dado número inteiro positivo é, ou não, um número primo.

Entrada: Uma única linha que contém: a quantidade de números ($N > 0$), seguido de um espaço em branco e dos números inteiros positivos n_i , ($n_i > 1$) separados por um espaço em branco entre si.

Saída: N linhas, cada uma com um número inteiro positivo n_i , seguido de: um espaço em branco; o caracter 'S' se o número n_i for primo, ou o caracter 'N' caso o número não seja primo; um espaço em branco e o menor número primo p tal que $p > n_i$.

Exemplo

Entrada	Saída
5 2 3 11 16 60	2 S 3 3 S 5 11 S 13 16 N 17 60 N 61

Problema 09) Área de polígonos

Um grande está cursando Arquitetura e pediu lhe auxílio para resolver o seguinte problema: o cálculo da área, em metros quadrados, de diversas figuras planas:

C círculo – o raio é dado por R ;

E elipse – os raios maior e menor são, respectivamente, R e r ;

R retângulo – os lados são a e b ;

T trapézio – as bases maior e menor e a altura são, respectivamente, B , b e H (nesta ordem).

Você elaborará um programa de computador que seja capaz de receber as informações necessárias e retornar a área da figura com precisão de três casas decimais. Obs.: use $\pi = 3,1415$.

Entrada: A primeira linha da entrada contém um inteiro N ($N > 0$) representando a quantidade de figuras planas que seu programa deve calcular a área.

Cada uma das N linhas seguintes será composta por um caractere que identifica a figura ('C', 'E', 'T' ou 'Z'), seguido de: um espaço em branco, os parâmetros necessários para calcular a área da figura, separados entre si por um espaço em branco.

Saída: N linhas, cada uma contendo o nome da figura em letras maiúsculas (CIRCULO, ELIPSE, RETANGULO OU TRAPEZIO), seguido de um espaço em branco e da área da respectiva figura (a área de cada figura plana deverá ser truncada para três casas decimais de precisão).

Exemplo

Entrada	Saída
4	CIRCULO 17.705
C 2.374	ELIPSE 39.581
E 2.657 4.742	RETANGULO 48.000
R 8.000 6.000	TRAPEZIO 25.761
T 7.328 3.752 4.650	