



II Maratona de Programação UCDB

Nível Superior

28 de Maio de 2019

Este caderno contém 10 problemas; as páginas estão numeradas de 1 a 11.

Informações Gerais

Estas condições são válidas para todos os problemas, exceto quando explicitamente dito.

Nome do programa

1. Sua solução deve ser chamada *problema.c*, *problema.cpp* ou *problema.java*; onde *problema* é a letra maiúscula que identifica o problema.

Entrada

1. A entrada de seu programa deve ser lida da entrada padrão.
2. A entrada consiste de um único caso de teste, que é descrito com uma quantidade de linhas definida no problema. Não há outras informações na entrada.
3. Quando uma linha contém diversos valores, eles são separados por *um único espaço*. Não há outros espaços na entrada. Não há linhas vazias.
4. Apenas o alfabeto em Inglês é utilizado. Não há caracteres com til, acento, trema ou outros símbolos (ñ, Ã, é, Ì, ô, Ü, ç, etcetera).
5. Todas as linhas, incluindo a última, contém o tradicional caractere de quebra-de-linha.

Saída

1. A saída de seu programa deve ser escrita na saída padrão.
2. O resultado de um caso de teste deve ser escrito com uma quantidade de linhas definida no problema. Não deve haver outras informações na saída.
3. Quando uma linha contém diversos valores, eles devem ser separados por *um único espaço*. Não deve haver outros espaços na entrada. Não deve haver linhas vazias.
4. Apenas o alfabeto em Inglês deve ser utilizado. Não deve haver caracteres com til, acento, trema ou outros símbolos (ñ, Ã, é, Ì, ô, Ü, ç, etcetera).
5. Todas as linhas, incluindo a última, devem conter o tradicional caractere de quebra-de-linha.
6. Para resultados de números reais, arredonde para o número racional mais próximo com a quantidade equivalente de dígitos de precisão. O caso de teste é feito de modo que não haja ambiguidade neste tipo de arredondamento.

Problema A - Jornada nas Estrelas

Após comprar vários sítios adjacentes na região do oeste catarinense, a família Estrela construiu uma única estrada que passa por todos os sítios em sequência. O primeiro sítio da sequência foi batizado de Estrela 1, o segundo de Estrela 2, e assim por diante. Porém, o irmão que vive em Estrela 1 acabou enlouquecendo e resolveu fazer uma Jornada nas Estrelas para roubar carneiros das propriedades de seus irmãos. Mas ele está definitivamente pirado. Quando passa pelo sítio Estrela i , ele rouba apenas um carneiro daquele sítio (se o sítio tem algum) e segue ou para Estrela $i + 1$ ou para Estrela $i - 1$, dependendo se o número de carneiros em Estrela i era, respectivamente, ímpar ou par. Se não existe a Estrela para a qual ele deseja seguir, ele interrompe sua jornada. O irmão louco começa sua Jornada em Estrela 1, roubando um carneiro do seu próprio sítio.

Entrada

A primeira linha da entrada consiste de um único inteiro N ($1 \leq N \leq 10^6$), o qual representa o número de Estrelas. A segunda linha da entrada consiste de N inteiros, de modo que o i -ésimo inteiro, X_i ($1 \leq X_i \leq 10^6$), representa o número inicial de carneiros em Estrela i .

Saída

Imprima uma linha contendo dois inteiros, de modo que o primeiro represente o número de Estrelas atacadas pelo irmão louco e o segundo represente o número total de carneiros não roubados.

Entrada 1 8 1 3 5 7 11 13 17 19	Saída 1 8 68
Entrada 2 8 1 3 5 7 11 13 16 19	Saída 2 7 63

Problema B - Motoboy

José é um motoboy e trabalha fazendo entregas para uma pizzaria. Seu salário é baseado no número de pizzas entregues. Só que esta pizzaria está com muito movimento e ele pediu auxílio a seu amigo Roberto para que o ajudasse nas entregas. Como Roberto é camarada e está sem trabalho no momento, ele concordou em pegar aqueles pedidos cujas entregas serão mais demoradas.

Assim, sempre que chegam à pizzaria, antes de partirem para novas entregas José determina a quantidade de pizzas que Roberto deverá entregar e seleciona para ele os pedidos mais demorados. Por exemplo, se há 22 pizzas para serem entregues e José determinar que Roberto entregue no máximo 10 destas pizzas (pode ser menos), estas devem estar obrigatoriamente entre os pedidos que levarão mais tempo para serem entregues. Isso é ilustrado no primeiro caso de teste, onde Roberto deverá fazer a entrega do segundo, terceiro e sexto pedido, somando 8 pizzas e 62 minutos ($23 + 21 + 18$). Se Roberto fosse realmente entregar 10 pizzas, ele teria que atender o segundo, terceiro e quarto pedido e isto levaria 59 minutos ($23 + 21 + 16$), o que não é o objetivo de José, pois levaria menos tempo do que a primeira opção, ou seja, a relação pizzas/tempo não importa muito para José (isso pode ser observado no segundo caso de teste do exemplo abaixo).

Para poder fazer a divisão do trabalho, José pediu a um amigo acadêmico em Ciência da Computação que desenvolvesse um programa que determinasse quanto tempo seu amigo Roberto irá levar para entregar estes pedidos mais demorados.

Entrada

A entrada contém vários casos de teste. Cada caso de teste contém na primeira linha um valor inteiro N ($1 \leq N \leq 20$) que indica o número de pedidos. A linha seguinte contém um valor inteiro P ($1 \leq P \leq 30$) indicando o número máximo de pizzas que podem ser entregues por Roberto. Cada uma das próximas N linhas contém um pedido com o tempo total para ser entregue e a quantidade de pizzas do pedido, respectivamente. A final da entrada é determinado por $N = 0$, e não deverá ser processado.

Saída

Para cada caso de teste de entrada deve ser impresso um valor inteiro que determina o tempo que Roberto irá levar para entregar as suas pizzas seguido de um espaço em branco e do texto “min.”, conforme exemplo abaixo.

Entrada 1	Saída 1
6	62 min.
10	47 min.
15 5	77 min.
23 4	39 min.
21 2	
16 4	
19 5	
18 2	
2	
15	
47 12	
39 4	
5	
23	
43 9	
4 1	
17 2	
13 5	
54 17	
6	
7	
14 4	
21 2	
26 7	
18 4	
30 13	
10 2	
0	

Problema C - Máquina de Café

O novo prédio da Sociedade Brasileira de Computação (SBC) possui 3 andares. Em determinadas épocas do ano, os funcionários da SBC bebem muito café. Por conta disso, a presidência da SBC decidiu presentear os funcionários com uma nova máquina de expresso. Esta máquina deve ser instalada em um dos 3 andares, mas a instalação deve ser feita de forma que as pessoas não percam muito tempo subindo e descendo escadas.

Cada funcionário da SBC bebe 1 café expresso por dia. Ele precisa ir do andar onde trabalha até o andar onde está a máquina e voltar para seu posto de trabalho. Todo funcionário leva 1 minuto para subir ou descer um andar. Como a SBC se importa muito com a eficiência, ela quer posicionar a máquina de forma a minimizar o tempo total gasto subindo e descendo escadas.

Sua tarefa é ajudar a diretoria a posicionar a máquina de forma a minimizar o tempo total gasto pelos funcionários subindo e descendo escadas.

Entrada

A entrada consiste em 3 números, $A1$, $A2$, $A3$ ($0 \leq A1, A2, A3 \leq 1000$), um por linha, onde Ai representa o número de pessoas que trabalham no i -ésimo andar.

Saída

Seu programa deve imprimir uma única linha, contendo o número total de minutos a serem gastos com o melhor posicionamento possível da máquina.

Entrada 1 10 20 30	Saída 1 80
Entrada 2 10 30 20	Saída 2 60
Entrada 3 30 10 20	Saída 3 100

Problema D - Corvo Contador

Como se sabe, existe um corvo com três olhos. O que não se sabia é que o corvo com três olhos pode prever o resultado da loteria de Westeros. Enquanto todos os outros corvos coletam as apostas, o corvo de três olhos já sabe o resultado, e quando Bran sonha com o corvo, o corvo conta o resultado. O problema é que Bran apesar de lembrar do sonho, não consegue interpretá-lo sozinho em tempo hábil. A sua tarefa é fazer um programa para interpretar o sonho de Bran e calcular o resultado da loteria.

Durante o sonho, o corvo pisca diversas vezes e grita apenas 3 vezes. A cada grito um número do resultado da loteria é calculado.

Cada piscada do corvo comunica um número em binário. Um olho aberto significa *1* e um olho fechado significa *0*. O olho da esquerda é o mais significativo e o da direita é o menos significativo. A cada piscada, este número deve ser somado, e quando o corvo grita, essa soma é um resultado.

Entrada

A entrada descreve, em cada linha, em sequência, ou um grito ou uma piscada do corvo.

Um grito é representado pela string *caw caw*

Uma piscada é representada por três caracteres *** ou *-*, representando, respectivamente, um olho aberto ou um olho fechado, da esquerda para a direita.

Lembre-se que o corvo tem 3 olhos.

Os números sorteados na loteria não excedem 1000.

Saída

Seu programa deve imprimir uma única linha, contendo o número total de minutos a serem gastos com o melhor posicionamento possível da máquina.

Entrada 1	Saída 1
--*	1
caw caw	4
*--	0
caw caw	
caw caw	
Entrada 2	Saída 2
--*	3
--*	8
--*	5
caw caw	
*--	
*--	
caw caw	
--*	
*--	
caw caw	

Problema E - Crescimento Populacional

Mariazinha quer resolver um problema interessante. Dadas as informações de população e a taxa de crescimento de duas cidades quaisquer (A e B), ela gostaria de saber quantos anos levará para que a cidade menor (sempre é a cidade A) ultrapasse a cidade B em população. Claro que ela quer saber apenas para as cidades cuja taxa de crescimento da cidade A é maior do que a taxa de crescimento da cidade B, portanto, previamente já separou para você apenas os casos de teste que tem a taxa de crescimento maior para a cidade A. Sua tarefa é construir um programa que apresente o tempo em anos para cada caso de teste.

Porém, em alguns casos o tempo pode ser muito grande, e Mariazinha não se interessa em saber exatamente o tempo para estes casos. Basta que você informe, nesta situação, a mensagem "Mais de 1 seculo."

Entrada

A primeira linha da entrada contém um único inteiro T , indicando o número de casos de teste ($1 \leq T \leq 3000$). Cada caso de teste contém 4 números: dois inteiros PA e PB ($100 \leq PA \leq 1000000$, $PA < PB \leq 1000000$) indicando respectivamente a população de A e B, e dois valores $G1$ e $G2$ ($0.1 \leq G1 \leq 10.0$, $0.0 \leq G2 \leq 10.0$, $G2 < G1$) com um dígito após o ponto decimal cada, indicando respectivamente o crescimento populacional de A e B (em percentual).

Atenção: A população é sempre um valor inteiro, portanto, um crescimento de 2.5 % sobre uma população de 100 pessoas resultará em 102 pessoas, e não 102.5 pessoas, enquanto um crescimento de 2.5% sobre uma população de 1000 pessoas resultará em 1025 pessoas. Além disso, não utilize variáveis de precisão simples para as taxas de crescimento.

Saída

Imprima, para cada caso de teste, quantos anos levará para que a cidade A ultrapasse a cidade B em número de habitantes. Obs.: se o tempo for mais do que 100 anos o programa deve apresentar a mensagem: Mais de 1 seculo. Neste caso, acredito que seja melhor interromper o programa imediatamente após passar de 100 anos, caso contrário você poderá receber como resposta da submissão deste problema "Time Limit Exceeded".

Entrada 1	Saída 1
6	51 anos.
100 150 1.0 0	16 anos.
90000 120000 5.5 3.5	12 anos.
56700 72000 5.2 3.0	Mais de 1 seculo.
123 2000 3.0 2.0	10 anos.
100000 110000 1.5 0.5	100 anos.
62422 484317 3.1 1.0	

Problema F - Cigarras Periódicas

As “cigarras periódicas” americanas têm o ciclo de vida mais longo de todos os insetos conhecidos. A cada 17 anos, estas cigarras periódicas amadurecem, se acasalam, depositam ovos e morrem. Suas crias se refugiam debaixo da terra, a 20 centímetros de profundidade, onde elas se alimentarão da seiva de raízes por 17 anos, até que chegue seu dia de buscar um lugar ao sol.

Acredita-se que esse número não aconteceu por acaso, outras espécies de cigarras da região tem ciclos de 13 anos, assim essas duas espécies emergem ao mesmo tempo apenas a cada 221 anos. Isso é desejável pois dessa forma a chance de que as duas espécies se misturem diminui consideravelmente e características indesejáveis de uma população não são introduzidas na outra.

Inspirado por esse fenômeno, uma nova variação de algoritmo evolutivo foi criada. Na última etapa desse algoritmo as melhores possíveis soluções são divididas em populações de modo que cada população i tem um ciclo de vida C_i . Além disso uma população extra também é adicionada, de modo que a quantidade de iterações até que o ciclo de vida de todas as populações coincida seja a maior possível. Essas populações são então avaliadas até que o ciclo de vida de todas coincida e a melhor solução ao final do processo é escolhida. Como não é interessante esperar demais até que o algoritmo gere uma resposta, um limite superior L no número de iterações também deve ser respeitado.

Dados os ciclos de vida das populações criadas e o limite na quantidade de iterações L , sua tarefa é computar qual o período ótimo para a população extra que será adicionada

Entrada

A primeira linha da entrada contém dois inteiros N e L , respectivamente, a quantidade de populações geradas pelas etapas anteriores do algoritmo e o limite da quantidade de iterações, $2 \leq N \leq 10^4$, $1 \leq L \leq 10^6$. A linha seguinte contém os N valores C_i representando a quantidade de iterações no ciclo de vida de cada população, onde $1 \leq C_i$. Você pode assumir que os ciclos de vida das populações atuais coincidem em menos de L iterações.

Saída

Seu programa deve produzir uma única linha com um inteiro representando o período da população extra que maximiza a quantidade T de iterações até que os ciclos de vida de todas as populações coincidam, respeitando a restrição de que $T \leq L$. Caso exista mais de um valor possível imprima o menor deles.

Entrada 1 2 5000 105 55	Saída 1 4
Entrada 2 2 512 3 14	Saída 2 72
Entrada 3 3 80 6 10 15	Saída 3 4

Problema G - Pares e Ímpares

Considerando a entrada de valores inteiros não negativos, ordene estes valores segundo o seguinte critério:

- Primeiro os Pares - Depois os Ímpares

Sendo que deverão ser apresentados os pares em ordem crescente e depois os ímpares em ordem decrescente.

Entrada

A primeira linha de entrada contém um único inteiro positivo N ($1 < N < 10^5$) Este é o número de linhas de entrada que vem logo a seguir. As próximas N linhas conterão, cada uma delas, um valor inteiro não negativo.

Saída

Apresente todos os valores lidos na entrada segundo a ordem apresentada acima. Cada número deve ser impresso em uma linha, conforme exemplo abaixo.

Entrada 1	Saída 1
10	4
4	32
32	34
34	98
543	654
3456	3456
654	6789
567	567
87	543
6789	87
98	

Problema H - Despojados

Todo inteiro positivo pode ser escrito como um produto de potências de primos. Por exemplo, $252 = 2^2 \cdot 3^2 \cdot 7$. Um inteiro é despojado se pode ser escrito como um produto de dois ou mais primos distintos, sem repetição. Por exemplo, $6 = 2 \cdot 3$ e $14 = 2 \cdot 7$ são despojados, mas $28 = 2^2 \cdot 7$, 1, 17 não são despojados.

Entrada

A entrada consiste de uma única linha que contém um inteiro N ($1 \leq N \leq 10^{12}$).

Saída

Seu programa deve produzir uma única linha com um inteiro representando o número de divisores despojados de N .

Entrada 1 252	Saída 1 4
Entrada 2 6469693230	Saída 2 1013
Entrada 3 8	Saída 3 0
Entrada 4 1	Saída 4 0
Entrada 5 88290298627	Saída 5 0

Problema I - Andando no Tempo

Imagine que você tenha uma máquina do tempo que pode ser usada no máximo três vezes, e a cada uso da máquina você pode escolher voltar para o passado ou ir para o futuro. A máquina possui três créditos fixos; cada crédito representa uma certa quantidade de anos, e pode ser usado para ir essa quantidade de anos para o passado ou para o futuro. Você pode fazer uma, duas ou três viagens, e cada um desses três créditos pode ser usado uma vez apenas. Por exemplo, se os créditos forem 5, 12 e 9, você poderia decidir fazer duas viagens: ir 5 anos para o futuro e, depois, voltar 9 anos para o passado. Dessa forma, você terminaria quatro anos no passado, em 2012. Também poderia fazer três viagens, todas indo para o futuro, usando os créditos em qualquer ordem, terminando em 2042.

Neste problema, dados os valores dos três créditos da máquina, seu programa deve dizer se é ou não possível viajar no tempo e voltar para o presente, fazendo pelo menos uma viagem e, no máximo, três viagens; sempre usando cada um dos três créditos no máximo uma vez.

Entrada

A entrada consiste de uma linha contendo os valores dos três créditos A , B e C ($1 \leq A, B, C \leq 1000$).

Saída

Seu programa deve imprimir uma linha contendo o caracter “S” se é possível viajar e voltar para o presente, ou “N” caso contrário.

Entrada 1 22 5 22	Saída 1 S
Entrada 2 31 110 79	Saída 2 S
Entrada 3 45 8 7	Saída 3 N

Problema J - Fatorial

Joãozinho é um garoto esperto da sexta série. Ele gosta muito de matemática, e descobriu que sua professora é muito preguiçosa. Nas provas da matéria a professora pede que as crianças circulem a resposta com um quadrado colorido, e que façam o primeiro dígito diferente de zero (da direita para esquerda) do número especialmente grande com caneta. Joãozinho desconfiou que a professora olhava apenas para aquele dígito para corrigir a questão.

A turma aprendeu a calcular o fatorial de um número, e isso será cobrado na próxima prova. Joãozinho está convencido de que não precisa escrever de fato o número correto, desde que o primeiro dígito (olhando da direita para esquerda) seja o correto. Sua tarefa neste problema é ajudar Joãozinho a calcular para um número inteiro n da entrada, o primeiro dígito (da direita para esquerda) de $n!$ que seja diferente de zero.

Entrada

A entrada é composta de diversas instâncias. A primeira linha de cada instância consiste um inteiro N ($1 \leq N \leq 1000000$).

A entrada termina com final de arquivo.

Saída

Para cada instância, você deverá imprimir um identificador Instancia K , onde K é o número da instância atual. Na linha seguinte imprima o primeiro dígito (da direita para esquerda) diferente de zero.

Após cada instância imprima uma linha em branco.

Entrada 1	Saída 1
5	Instancia 1 2