

# SAET 2023 - Maratona de Programação

*26 de Outubro de 2022*



## Instruções Importantes

- Use a opção **Runs** para enviar suas soluções. Os problemas podem ser resolvidos em qualquer ordem e linguagem (dentro de C, C++ e Python, independentemente do problema);
- Suas soluções serão testadas com várias entradas, além da dada como exemplo. Por isso, sua solução pode não ser aceita mesmo se funcionar para os exemplos dados. Certifique-se que ela funciona para todas as entradas possíveis;
- A saída gerada deve ser *exatamente* conforme especificada. Em particular, **não** imprima instruções (“digite um número”, “a resposta é”, etc);
- É garantido que todas as entradas usadas para teste estarão de acordo com o enunciado, não sendo necessário testar se são válidas;
- Ao enviar uma solução, o sistema irá responder uma das seguintes respostas:
  - **Not answered yet**: a solução está sendo corrigida. Aguarde um pouco e atualize a página;
  - **YES**: solução aceita. Parabéns!
  - **Wrong Answer**: a saída impressa pelo seu programa não é a saída correta esperada, para alguma entrada de teste;
  - **Presentation Error**: a saída impressa está correta, exceto por espaços em branco e/ou quebras-de-linha faltando/sobrando;
  - **Time Limit Exceeded**: o tempo de execução do seu programa ultrapassou o tempo limite estipulado para o problema (ver tabela abaixo). O tempo de execução da sua solução precisa ser menor;
  - **Runtime Error**: seu programa gerou algum erro em tempo de execução (“crashou”);
  - **Compile Error**: seu programa não compila.
- Todas as linhas, tanto na entrada quanto na saída, terminam com o caractere de fim-de-linha ( $\backslash n$ ), mesmo quando houver apenas uma única linha na entrada e/ou saída;

- Sua solução deve processar cada arquivo de entrada no tempo máximo estipulado para cada problema, dado pela seguinte tabela:

Problema	Nome	Tempo Limite (segundos)
A	Alergia	1
B	Jogo	1
C	Vasilha Errada	1
D	Drawcabackward	1
E	Empilha Copos	1
F	Falco	1
G	Gafe	1
H	Raid	1
I	Lista	1
J	Meuzamigo	1
K	Tiras	1

## A: Alergia

Arquivo: `alergia.[c|cpp|py]`

Alergia não é coisa só de humano não, a bicharada também sofre.

Maya tem um cachorrinho parceiro e super animal chamado Thomy. Ele tem o costume de cantar de galo, mandando Maya lhe dar comida. Claro que tudo em "cachorrês", afinal, ele é um cachorro! Animal é quem não entende ele.

Sempre forte como um touro, hoje Thomy amanheceu andando como uma barata tonta pela casa... deu zebra! Maya não poderia deixá-lo desamparado, pois não tem sangue de cobra e é uma mãe coruja com os seus bichinhos. Não titubeou: como uma lebre o levou para a veterinária.

O doutor prescreveu uma nova ração, já que Thomy havia desenvolvido alergia a alguns ingredientes. Ofereceram uma ração lá do Peru, que era o olho da cara... mais caro que um boi! Nem que a vaca tussa que Maya iria pagar tudo aquilo.

Ela procurou outra veterinária, e lá eles prescreveram uma ração bem mais barata, uma pechincha. Tão barata que ela desconfiou, e pediu para você confirmar se algum dos ingredientes da ração poderia causar uma reação no cachorrinho. Mostre quem é o bicho da programação, criando um programa que compare os ingredientes presentes na ração e as alergias do Thomy!

### Entrada

A primeira linha tem apenas um inteiro  $N$  ( $1 \leq N \leq 100$ ), o número de ingredientes.

A segunda linha contém  $N$  inteiros  $R_i$  ( $1 \leq i \leq N$ ). Se  $R_i = 1$ , então a ração contém o ingrediente  $i$ . Se  $R_i = 0$ , então ela não contém o ingrediente  $i$ .

A terceira linha contém  $N$  inteiros  $A_i$ , indicando se Thomy tem alergia ao ingrediente  $i$ . Se  $A_i = 1$ , então Thomy tem alergia ao ingrediente  $i$ . Se  $A_i = 0$ , então ele não tem alergia ao ingrediente  $i$ .

### Saída

Imprima "S" (maiúsculo e sem aspas) se Maya pode dar a ração para o Thomy.

Imprima "N" (maiúsculo e sem aspas) se Maya não deve dar a ração para o Thomy.

Exemplo de entrada	Exemplo de saída
6 1 0 0 1 1 0 0 0 1 0 0 1	S

Exemplo de entrada	Exemplo de saída
4 1 1 1 1 0 1 0 0	N

## B: Jogo

Arquivo: `jogo.[c|cpp|py]`

Carlos e Pedro gostam muito de futebol e querem jogar uma partida um contra o outro. Como ambos são goleiros, eles precisam de jogadores no ataque e na defesa para completar o seu time.

Para isso, eles têm uma lista de jogadores interessados em participar da partida, contendo os pontos de ataque e de defesa de todos. Um jogador com pontos de defesa maior que os pontos de ataque é um defensor; um jogador com pontos de ataque maiores ou iguais aos de defesa é um atacante; o *talento* de um jogador é dado pela soma de seus pontos de ataque com seus pontos de defesa.

Para a divisão ser justa, a escolha dos jogadores será feitas em rodadas. Em cada rodada, Carlos e Pedro escolherão um jogador cada para seus times. Na primeira rodada Carlos escolhe um jogador primeiro; na segunda rodada Pedro escolhe um jogador primeiro; e assim por diante. Caso o número de jogadores seja ímpar, um deles não será escolhido.

A escolha de um jogador é feita pelos seguintes critérios:

- Se um time já tiver mais atacantes do que defensores, será escolhido para o time o defensor mais talentoso ainda não escolhido. Caso não haja mais defensores disponíveis, será escolhido o atacante mais talentoso;
- Se um time já tiver mais defensores do que atacantes, será escolhido para o time o atacante mais talentoso ainda não escolhido. Caso não haja mais atacantes disponíveis, será escolhido o defensor mais talentoso;
- Se um time tiver a mesma quantidade de atacantes e defensores, será escolhido o jogador mais talentoso ainda não escolhido. Se houver mais de um jogador ainda não escolhido com o maior talento, será escolhido um atacante.

Sua tarefa é determinar o talento médio do time mais talentoso.

### Entrada

A primeira linha da entrada contém o número  $N$  de jogadores a serem escolhidos ( $2 \leq N \leq 10^5$ ). Cada uma das  $N$  linhas seguintes contém 2 inteiros:  $K_i$  e  $M_i$  ( $0 \leq K_i \leq 99, 0 \leq M_i \leq 99$ ), sendo, respectivamente, os pontos de ataque e de defesa do jogador  $i$ .

### Saída

A saída deverá conter uma linha com o talento médio do time mais talentoso, com exatamente duas casas decimais.

Exemplo de entrada	Exemplo de saída
4 10 20 30 10 10 50 10 5	37.50

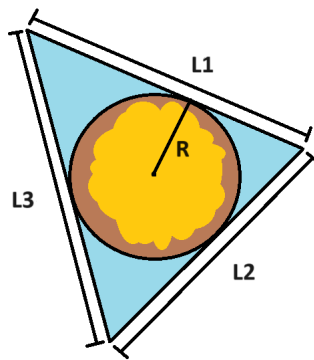
Exemplo de entrada	Exemplo de saída
6 62 1 79 65 71 71 8 91 71 99 20 24	125.00

Exemplo de entrada	Exemplo de saída
10 61 55 49 81 96 23 2 59 56 41 33 20 93 16 72 20 65 66 58 94	117.40

## C: Vasilha Errada

Arquivo: `vasilhaerrada.[c|cpp|py]`

Nathan é um menino apaixonado por triângulos, por conta disso em sua casa só tem vasilhas triangulares. Contudo, sua mãe Ana sempre faz panquecas circulares para o seu filho levar para escola. Como Nathan é ruim em matemática, ele nunca sabe qual o melhor recipiente para levar. Por sorte no seu material escolar há uma régua, conseguindo dessa forma medir os lados de todas as vasilhas da casa. Sendo assim, o seu trabalho será criar um programa que lê as medidas de Nathan e mostrá-lo quais são os raios limites que a panqueca pode ter para cada recipiente.



### Entrada

A primeira linha da entrada contém o número  $N$  ( $1 \leq N \leq 1000$ ) de vasilhas.

As próximas  $N$  linhas conterá os valores de  $L_1$ ,  $L_2$ ,  $L_3$  ( $1 \leq L_1, L_2, L_3 \leq 500$ ) em cm representando as medidas realizadas por Nathan em cada recipiente. Considere que Nathan não erre as medidas e que a precisão pode alcançar duas casas decimais.

### Saída

A saída deverá conter  $N$  números, sendo um valor por linha, que serão os raios  $R$  máximos, com duas casas decimais, que a panqueca pode ter em cada vasilha apresentada por Nathan.

Exemplo de entrada	Exemplo de saída
6	1.00
3 4 5	3.46
12 12 12	3.00
10 12 10	0.45
2.08 1.82 1.30	13.61
33.07 96.2 103.86	33.24
131.27 316.33 216.7	



## D: Drawkcabbackward

Arquivo: drawkcab.[c|cpp|py]

Uma das mais divertidas tarefas de um maratonista de programação é a escolha do nome da sua equipe. Depois de muito debate, sua equipe decidiu escolher o nome da seguinte maneira:

- O nome deverá ser uma substring não vazia de uma dada string  $s$ ;
- O nome deverá ser palíndromo (isto é, deve ser igual quando lido da esquerda para a direita e da direita para a esquerda);
- Cada letra do alfabeto  $a, b, c, \dots, z$  tem um *valor*  $V_a, V_b, V_c, \dots, V_z$ . O *valor total* do nome escolhido é a soma dos valores de suas letras. O nome escolhido deverá ter o maior valor total possível.

Como exemplo, considere a string  $s = \text{xabaydcbbcdyz}$  e os valores  $V_a = 20$ ,  $V_b = -10$ ,  $V_c = 15$ ,  $V_d = 11$ ,  $V_x = 20$ ,  $V_y = -20$  e  $V_z = 20$ . Alguns nomes que poderiam ser escolhidos são:

- $\text{aba}$ , com valor total  $V_a + V_b + V_a = 20 - 10 + 20 = 30$ ;
- $\text{dcbbcd}$ , com valor total  $11 + 15 - 10 - 10 + 15 + 11 = 32$ ;
- $\text{ydcbbcdy}$ , com valor total  $-20 + 11 + 15 - 10 - 10 + 15 + 11 - 20 = -8$ ;
- Outras substrings palíndromes de  $s$ .

Neste exemplo, a substring palíndromo  $\text{dcbbcd}$  tem o maior valor total possível (32), e portanto este será o nome da equipe.

Dada a string  $s$  e os valores de cada letra, ajude a escolher o nome da sua equipe!

### Entrada

A primeira linha contém 26 valores inteiros  $V_a, V_b, V_c, \dots, V_z$  (entre  $-1000$  e  $1000$  cada) indicando o valor de cada letra do alfabeto.

A segunda linha contém a string  $s$  ( $1 \leq |s| \leq 5 \times 10^5$ ), contendo apenas letras minúsculas.

### Saída

Imprima uma única linha contendo o valor total do nome escolhido pela equipe.

<b>Exemplo de entrada</b> 20 -10 15 11 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 20 0 -20 20 xabaydcbbcdyz	<b>Exemplo de saída</b> 32
<b>Exemplo de entrada</b> 1 -1 0 bbbaaaaaabbabbbbaaa	<b>Exemplo de saída</b> 6
<b>Exemplo de entrada</b> 7 1 1 8 1 -5 1 1 3 1 -10 1 71 -42 1 1 1 1 1 1 1 1 1 1 1 1 lazafakeekafdrawkcabackwardlaza	<b>Exemplo de saída</b> 31
<b>Exemplo de entrada</b> -1 0 aaa	<b>Exemplo de saída</b> -1

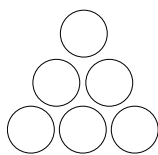
## E: Empilha Copos

Arquivo: `empilha.[c|cpp|py]`

Um professor de matemática que é fascinado por competições resolveu fazer uma dinâmica com a sua turma. Ele percebeu que em sua casa havia uma grande quantidade de copos, e optou por utilizá-los em sua próxima aula.

Ele explicou da seguinte maneira para os seus alunos: Dado  $N$  copos ( $1 \leq N \leq 10000$ ), eles devem determinar qual o maior valor de  $B$  tal que é possível formar um triângulo de base  $B$  utilizando no máximo  $N$  copos.

Como exemplo, a figura abaixo mostra que, com no máximo  $N = 8$  copos, é possível formar um triângulo de base  $B = 3$ :



Neste exemplo, 6 copos são usados e 2 são descartados.

### Entrada

A entrada contém apenas um inteiro  $N$  ( $1 \leq N \leq 10000$ ).

### Saída

Imprima o maior valor de  $B$  tal que é possível formar um triângulo de base  $B$  usando até  $N$  copos.

Exemplo de entrada	Exemplo de saída
10	4

Exemplo de entrada	Exemplo de saída
8	3

Exemplo de entrada	Exemplo de saída
1000	44

## F: Falco

Arquivo: `falco.[c|cpp|py]`

Agente Falco é um experiente espião da Nlogonia, especializado em infiltrações. Para se comunicar com seus aliados ele utiliza mensagens codificadas na cifra de César, uma técnica de criptografia simples, mas eficaz.

A cifra de César é uma técnica de criptografia que envolve a substituição de cada letra do alfabeto por outra letra, deslocada um número fixo de posições, de maneira circular. Por exemplo, com um deslocamento de 3 posições, a letra "A" se torna "D", a letra "B" se torna "E", a letra "Z" se torna "C", e assim por diante. O deslocamento é um segredo compartilhado com o destinatário da mensagem.

Sua tarefa é desenvolver um programa que codifica uma mensagem de texto usando a cifra de César com um deslocamento especificado para o Agente Falco.

### Entrada

A primeira linha da entrada contém 2 inteiros  $t$  ( $0 < t \leq 100$ ) e  $k$  ( $0 < k \leq 10^{18}$ ), sendo respectivamente o tamanho da mensagem e o deslocamento.

A segunda linha contém a mensagem  $s$ , composta apenas de caracteres minúsculos.

### Saída

A saída deverá ser a mensagem codificada.

Exemplo de entrada	Exemplo de saída
4 1 aaaa	bbbb

Exemplo de entrada	Exemplo de saída
4 3 saet	vdhw

Exemplo de entrada	Exemplo de saída
1 27 a	b

## G: Gafe

Arquivo: `gafe.[c|cpp|py]`

Gabriel fez aniversário semana retrasada, e tudo o que ele queria de presente era um fone de ouvido. Não tinha dúvidas sobre o que desejar no momento de apagar as velas do seu bolinho de aniversário.

A festa ia começar, quando a SAET (Serviço de Aplicação e Execução Tática), um dos departamentos da polícia federal, invadiu equivocadamente a sua casa, pensando que se tratava do domicílio de um dos criminosos mais procurados do país. Tudo ocorreu por causa do cabo Paradella, que confundiu o número 66 por 50 nas evidências. Este erro desencadeou uma série de eventos que culminaram em uma incursão à casa de Gabriel. Que gafe!

Após perceberem o erro, a comandante do esquadrão deu de presente um dos fones de ouvido táticos usados pela polícia como pedido de desculpas ao Gabriel. Apesar de super tecnológico, não tem um botão de desligar. Ao invés disso, tem três botões capazes de diminuir o nível do volume de formas muito inconventionais:

- **Botão E-**: Diminui o nível do lado esquerdo em 1. Não faz nada se o nível já for 0;
- **Botão D-**: Diminui o nível do lado direito em 1. Não faz nada se o nível já for 0;
- **Botão S**: Subtrai o nível do lado mais alto pelo nível do lado mais baixo. Se os volumes forem iguais, um lado qualquer é subtraído do outro.

Para exemplificar, se o volume do fone estiver nos níveis  $[880, 980]$ , o botão **E-** o fará mudar para os níveis  $[879, 980]$ , o botão **D-** para  $[880, 979]$ , e o botão **S** para  $[880, 100]$ .

Acontece que o nível de cada lado é *independente* e vai até 1000, e Gabriel não quer ficar apertando os botões milhares de vezes até desligar o fone. Dado os níveis do fone no momento, crie um programa que ajude Gabriel, dizendo a quantidade mínima de vezes que ele terá de apertar os botões para desligar completamente o seu novo fone de ouvido.

## Entrada

A entrada contém apenas uma linha com dois inteiros  $E$  e  $D$  ( $0 \leq E, D \leq 1000$ ), o volume do lado esquerdo e direito do fone, respectivamente.

## Saída

Imprima uma única linha contendo a quantidade mínima de vezes que Gabriel terá de apertar os botões para desligar ambos lados do fone.

Exemplo de entrada	Exemplo de saída
2 2	3

Exemplo de entrada	Exemplo de saída
11 15	8

Exemplo de entrada	Exemplo de saída
1000 976	48

## H: Raid

Arquivo: `raid.[c|cpp|py]`

Seu melhor amigo, Gabriel, te pediu ajuda para criar a melhor estratégia para a raid que ele irá participar em LIT (Lost In Time), o jogo favorito dele. Ele te explicou como a raid funciona da seguinte forma:

- Para causar dano nos monstros você deve usar magias.
- Para cada magia usada você causa exatamente o dano dela no monstro e o monstro é eliminado quando ele tiver 0 HP.
- Caso o dano proporcionado pela magia exceda o HP atual do monstro, você recebe uma nova magia cujo dano é dado pela diferença entre o HP atual do monstro e o dano total da magia.

Para se preparar para a raid ele deve comprar as magias usando as LIT'coins.

- As magias disponíveis estão dispostas linearmente na loja.
- Cada magia custa exatamente seu valor em dano de LIT'coins.
- Você só pode comprar todas as magias dentro de um único intervalo  $[i; j]$  ( $a_i, a_{i+1}, a_{i+2}, \dots, a_j$ ).

O pedido de Gabriel é simples, ajude ele determinar qual a menor quantidade possível de LIT'coins para vencer a raid.

### Entrada

A primeira linha contém dois inteiros  $N$  e  $M$  ( $1 \leq M \leq N \leq 10^5$ ).

A segunda linha contém  $N$  inteiros ( $1 \leq a_i \leq 100$ ) que representam as magias disponíveis para a raid.

A terceira linha contém  $M$  inteiros ( $1 \leq b_i \leq 100$ ) que representam o HP dos monstros da raid.

### Saída

Imprima a quantidade mínima de LIT'coins para vencer a raid. Caso não seja possível completá-la imprima  $-1$ .

Exemplo de entrada	Exemplo de saída
4 3 2 3 4 6 3 1 2	6

Exemplo de entrada	Exemplo de saída
4 2 3 1 2 6 5 1	6

Exemplo de entrada	Exemplo de saída
4 4 2 3 3 4 5 2 1 8	-1



## I: Lista

Arquivo: `lista.[c|cpp|py]`

Pedro tem uma clínica que funciona 7 dias por semana, porém ele não gosta de trabalhar muito aos finais de semana, e por isso pediu para sua secretária distribuir os seus  $N$  pacientes de acordo com a primeira letra de seus nomes em certos dias da semana.

Ele atende as pessoas que têm os nomes que começam com as letras A, B, C e D no domingo; já na segunda-feira ele atende as pessoas que têm os nomes que começam com as letras E, F, G e H; e assim por diante até a sexta-feira, quando só atende as pessoas com as letras U, V e W, e no sábado X, Y e Z.

Entretanto, ele pediu para sua secretária que não organizasse os pacientes, dentro de um mesmo dia, em ordem alfabética, pois assim seria injusto com os pacientes que marcaram suas consultas antes. Dentro de um mesmo dia, a lista de pacientes deve seguir a ordem em que marcaram as consultas.

Assim, por exemplo, se Caio, Felipe, Diego, Ana, Bianca, Alice e Tiago marcaram consultas (nessa ordem), seriam atendidos no domingo, nessa ordem: Caio, Diego, Ana, Bianca e Alice. Já Felipe seria atendido na segunda-feira, portanto após todos estes. Tiago seria atendido por último.

### Entrada

A primeira linha contém  $N$  ( $1 \leq N \leq 1000$ ), o número de pacientes. As próximas  $N$  linhas contém um nome cada, na ordem em que marcaram as consultas. Os nomes serão strings de até 20 letras minúsculas e/ou maiúsculas cada.

### Saída

Imprima um nome de paciente por linha, na ordem em que serão atendidos.

Exemplo de entrada	Exemplo de saída
5 bruna Gabi ana Caio bianca	bruna ana Caio bianca Gabi

## J: Meuzamigo

Arquivo: meuzamigo.[c|cpp|py]

A universidade Blupen está oferecendo uma gincana com várias brincadeiras de premiações interessantes. Uma delas é a Meuzamigo e nosso amigo Manoel Gomes está decidido a ganhar. A brincadeira consiste em em  $N$  pessoas conectadas a  $N - 1$  fios. O participante é o primeiro a se conectar a rede, em seguida os ajudantes da gincana se conectam para dar vida a brincadeira.

Para vencer, é necessário que o participante diga qual é a distância mais longa entre ele mesmo e um ajudante qualquer. É garantido que todos as pessoas estejam conectadas na rede.

### Entrada

A primeira linha da entrada contém o número de pessoas  $N$  ( $2 \leq n \leq 10^5$ ).

As próximas  $N - 1$  linhas contém  $A$  e  $B$ , as conexões entre cada participante.

### Saída

A saída deverá conter um número inteiro correspondente a distância mais longa entre Manoel Gomes e algum participante.

Exemplo de entrada	Exemplo de saída
9 1 9 1 4 9 5 4 3 5 6 1 2 1 7 5 8	3

Exemplo de entrada	Exemplo de saída
2 1 2	1

Exemplo de entrada	Exemplo de saída
19 1 7 7 12 12 6 1 14 6 13 13 17 14 9 12 2 14 18 1 10 10 8 13 11 1 15 13 16 1 4 6 5 7 19 2 3	5

## K: Tiras

Arquivo: `tiras.[c|cpp|py]`

Sila trabalha em um pequeno ateliê de costura próprio onde é confeccionado de tudo, desde camisetas e uniformes até o vestuário infantil. Depois de uma semana cheia de entregas, a maior cooperativa da cidade, SAET (Sociedade de Agropecuária e Empreendedorismo de Tumangapurubá), pediu para que ela produzisse bodies para bebês, pois o departamento de Recursos Humanos da cooperativa queria dar um de presente para toda funcionária ou cooperada gestante que entrasse de licença maternidade.

Para confeccionar os bodies, existem  $n$  caixas com tiras de comprimento  $c_i$ . Sila pediu para que seu marido, Amário, cortasse essas tiras em tamanhos iguais, sem deixar sobras e procurando deixar as tiras o mais compridas possível. Para fazer isso, Amário criou o seguinte processo para juntar duas caixas adjacentes:

- Pegar duas caixas adjacentes  $i$  e  $i + 1$  com tiras de comprimentos  $c_i$  e  $c_{i+1}$ ;
- Cortar todas as tiras das duas caixas para que os pedaços fiquem com tamanho  $\text{MDC}(c_i, c_{i+1})$ , onde MDC é a operação Máximo Divisor Comum;
- Substituir as duas caixas  $i$  e  $i + 1$  por uma caixa com as tiras de tamanho  $\text{MDC}(c_i, c_{i+1})$  recém cortadas.

Para ter uma noção de quanto Amário irá demorar, Sila mandou que você determinasse qual é o mínimo de vezes que o processo de junção terá de ser realizado para que, ao final, todas as tiras das caixas remanescentes tenham o mesmo tamanho. Note que a quantidade de tiras não importa, mas sim o tamanho delas em cada caixa.

Por exemplo, se houverem  $n = 5$  caixas, cada uma com tiras de comprimento  $c = [10, 30, 12, 42, 2]$ , o processo pode ser realizado um mínimo de 3 vezes até que, ao final, todas as caixas tenham tiras de mesmo comprimento e o processo não possa mais ser realizado. Uma das formas de juntar as caixas 3 vezes, satisfazendo o pedido de Sila, é:

$$[10, 30, 12, 42, 2] \rightarrow [10, 12, 42, 2] \rightarrow [10, 6, 2] \rightarrow [2, 2]$$

Em vermelho, estão o comprimento das tiras das duas caixas escolhidas para realizar a junção.

### Entrada

A primeira linha da entrada contém o número de caixas  $n$  ( $2 \leq n \leq 10^5$ ).

A próxima linha consiste em  $n$  inteiros  $c_i$  ( $1 \leq c_i \leq 10^5$ ), o tamanho das tiras dentro da caixa  $i$ .

### Saída

A saída deverá conter uma linha com o número de processos de junção realizados, atendendo à ordem de Sila.

Exemplo de entrada	Exemplo de saída
5 10 30 12 42 2	3

Exemplo de entrada	Exemplo de saída
5 16 4 4 2 8	4

Exemplo de entrada	Exemplo de saída
3 1 1 100000	1