

# 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 das dadas 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
D	Drawkcabackward	1
K	Tiras	1

## A: Alergia

Arquivo: `jogo.[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 tem uma lista de jogadores interessados a participar da partida, contendo os pontos de ataque e de defesa de todos. Quem tem seus pontos de defesa maior que os de ataque, joga na defesa, caso contrário, joga no ataque.

Para a divisão ser justa, a escolha dos jogadores será feitas em rodadas. Carlos e Pedro escolherão um jogador por rodada. Na primeira rodada Carlos escolhe um jogador primeiro, na segunda Pedro escolhe primeiro e vão intercalando até acabarem os jogadores. Caso o numero de jogadores seja impar, um não sera escolhido.

A cada rodada a escolha é feita de forma bem simples:

- Quem tiver a maior quantidade de pontos somados é escolhido;
- Se alguém do ataque e da defesa tem a mesma quantidade de pontos, ambos tem preferência pelo atacante;
- Caso tenha mais jogadores do ataque escolhidos, se possivel, um jogador da defesa será escolhido;
- Caso tenha mais jogadores da defesa escolhidos, se possivel, um jogador do ataque será escolhido;

## Entrada

A primeira linha da entrada contém o número de jogadores a serem escolhidos  $N$  ( $2 \leq N \leq 10^5$ ). 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 a média do time com a maior quantidade de pontos somados, com truncamento em duas casas decimais.

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

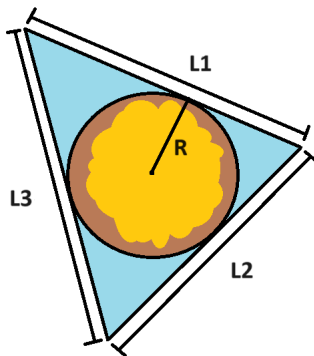
Exemplo de entrada	Exemplo de saída
6 62 1 79 65 71 71 8 91 71 99 20 24	125.00

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  $L1, L2, L3$  ( $1 \leq L1, L2, L3 \leq 500$ ) 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 apresentado 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: Drawkbackward

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 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

## 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  $\text{MDC}$  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