

Iniciado em	domingo, 24 out 2021, 13:10
Estado	Finalizada
Concluída em	domingo, 24 out 2021, 20:21
Tempo empregado	7 horas 10 minutos
Avaliar	8,14 de um máximo de 10,00(81%)

Aviso!

?

Questão 1

Correto

Atingiu 1,00 de 1,00

Inversão de Valores

Crie um [programa](#) que lê uma [string](#) *chave* e um número inteiro, em que a [string](#) será uma chave do dicionário *verso* e o número inteiro é o valor associado a essa chave. Repita essa leitura 5 vezes, ou seja, haverá 5 chaves no dicionário.

Imprima o dicionário *verso*.

Agora crie o dicionário *reverso*, cujas chaves são os números inteiros associados às chaves de *verso*, e seus valores são as chaves de *verso*.

Por fim, imprima o dicionário *reverso*.

Obs: todas as strings e números inteiros lidos serão diferentes.

Entrada

A entrada consiste em uma [string](#) e um número inteiro, repetidos 5 vezes.

Saída

A saída consiste no dicionário formado a partir das leituras e sua versão inversa.

Observações

- No primeiro exemplo de teste, foi digitado: "primeiro", 1, "segundo", 2, "terceiro", 3, "quarto", 4, "quinto" e 5. Imprimindo: {*'primeiro': 1, 'segundo': 2, 'terceiro': 3, 'quarto': 4, 'quinto': 5*} e {*1: 'primeiro', 2: 'segundo', 3: 'terceiro', 4: 'quarto', 5: 'quinto'*}.

For example:

Aviso!

Result

?

Input	Result
primeiro 1 segundo 2 terceiro 3 quarto 4 quinto 5	{'primeiro': 1, 'segundo': 2, 'terceiro': 3, 'quarto': 4, 'quinto': 5} {1: 'primeiro', 2: 'segundo', 3: 'terceiro', 4: 'quarto', 5: 'quinto'}
numero1 382 numero2 849 numero3 1 numero4 48239 numero5 12	{'numero1': 382, 'numero2': 849, 'numero3': 1, 'numero4': 48239, 'numero5': 12} {382: 'numero1', 849: 'numero2', 1: 'numero3', 48239: 'numero4', 12: 'numero5'}
testando 382 mais 12 um 1 exemplo 58932 batuta 0	{'testando': 382, 'mais': 12, 'um': 1, 'exemplo': 58932, 'batuta': 0} {382: 'testando', 12: 'mais', 1: 'um', 58932: 'exemplo', 0: 'batuta'}

Answer: (penalty regime: 0, 0, 10, 20, ... %)

1 QTD_ITEMS = 5

2 verso = dict()

Aviso! everso = dict()

?

```

5 ▼ for i in range(QTD_ITEMS):
6     chave = input()
7     valor = int(input())
8     verso[chave] = valor
9
10 ▼ for chave in verso:
11     reverso[verso[chave]] = chave
12
13 print(verso)
14 print(reverso)

```

	Input	Expected	Got	
✓	primeiro 1 segundo 2 terceiro 3 quarto 4 quinto 5	{'primeiro': 1, 'segundo': 2, 'terceiro': 3, 'quarto': 4, 'quinto': 5} {1: 'primeiro', 2: 'segundo', 3: 'terceiro', 4: 'quarto', 5: 'quinto'}	{'primeiro': 1, 'segundo': 2, 'terceiro': 3, 'quarto': 4, 'quinto': 5} {1: 'primeiro', 2: 'segundo', 3: 'terceiro', 4: 'quarto', 5: 'quinto'}	✓
✓	numero1 382 numero2 849 numero3 1 numero4 48239 numero5 12	{'numero1': 382, 'numero2': 849, 'numero3': 1, 'numero4': 48239, 'numero5': 12} {382: 'numero1', 849: 'numero2', 1: 'numero3', 48239: 'numero4', 12: 'numero5'}	{'numero1': 382, 'numero2': 849, 'numero3': 1, 'numero4': 48239, 'numero5': 12} {382: 'numero1', 849: 'numero2', 1: 'numero3', 48239: 'numero4', 12: 'numero5'}	✓

Aviso!

?

	Input	Expected	Got	
✓	testando 382 mais 12 um 1 exemplo 58932 batuta 0	{'testando': 382, 'mais': 12, 'um': 1, 'exemplo': 58932, 'batuta': 0} {382: 'testando', 12: 'mais', 1: 'um', 58932: 'exemplo', 0: 'batuta'}	{'testando': 382, 'mais': 12, 'um': 1, 'exemplo': 58932, 'batuta': 0} {382: 'testando', 12: 'mais', 1: 'um', 58932: 'exemplo', 0: 'batuta'}	✓
✓	1 2 3 4 5 6 7 8 9 10	{'1': 2, '3': 4, '5': 6, '7': 8, '9': 10} {2: '1', 4: '3', 6: '5', 8: '7', 10: '9'}	{'1': 2, '3': 4, '5': 6, '7': 8, '9': 10} {2: '1', 4: '3', 6: '5', 8: '7', 10: '9'}	✓
✓	vamos testar 381 aqui 0 nossa -1 mais um -18232 exemplo! 381	{'vamos testar': 381, 'aqui': 0, 'nossa': -1, 'mais um': -18232, 'exemplo!': 381} {381: 'exemplo!', 0: 'aqui', -1: 'nossa', -18232: 'mais um'}	{'vamos testar': 381, 'aqui': 0, 'nossa': -1, 'mais um': -18232, 'exemplo!': 381} {381: 'exemplo!', 0: 'aqui', -1: 'nossa', -18232: 'mais um'}	✓

Aviso!

?

	Input	Expected	Got	
✓	<pre> musica 123 indiozinhos 456 indiozinhos2 789 indiozinhos3 10 nao sei o resto da musica 0 </pre>	<pre> {'musica': 123, 'indiozinhos': 456, 'indiozinhos2': 789, 'indiozinhos3': 10, 'nao sei o resto da musica': 0} {123: 'musica', 456: 'indiozinhos', 789: 'indiozinhos2', 10: 'indiozinhos3', 0: 'nao sei o resto da musica'} </pre>	<pre> {'musica': 123, 'indiozinhos': 456, 'indiozinhos2': 789, 'indiozinhos3': 10, 'nao sei o resto da musica': 0} {123: 'musica', 456: 'indiozinhos', 789: 'indiozinhos2', 10: 'indiozinhos3', 0: 'nao sei o resto da musica'} </pre>	✓

Passou em todos os teste! ✓

Question author's solution (Python3):

```

1 | verso = {}
2 |
3 | for i in range(5):
4 |     chave = input()
5 |     verso[chave] = int(input())
6 |
7 | print(verso)
8 |
9 | reverso = {}
10 |
11 | for element in verso:

```

Correto

Notas para este envio: 1,00/1,00.

Aviso!

?

Questão 2

Não respondido

Atingiu 0,00 de 1,00

Correndo

Você gosta muito de praticar esportes, mas por causa da pandemia, quase todos os esportes ficaram difíceis de serem praticados. Após alguns meses triste em casa, você decidiu que tinha que voltar a se exercitar, como você é uma pessoa consciente, obviamente não irá sair de casa e, como você tem um corredor (beeeeem) longo, decidiu que irá correr nele para voltar a forma física. Porém, como você não tem saído muito de casa, algumas coisas estão faltando. Uma delas são seus cartazes contendo as distâncias em metros para descobrir quanto você correu. Então você pegou os cartazes que tinha e colocou eles a cada um metro. Como é um corredor, correr em linha reta não é a melhor das atividades mais animadoras, então você as vezes escolhe mudar de direção e correr na direção contrária. Como você está correndo, não tem tempo para ficar marcando quantos metros correu, então você só marca os cartazes em que você mudou de direção. Agora você se pergunta, qual o problema? Parece tudo certo, certo? Errado! Cabe a você, após seu treino físico, fazer seu treino mental e fazer um [algoritmo](#) que compute quantos metros você percorreu no total!

Entrada

A primeira linha da entrada consiste em dois números n, x ($1 \leq n, x \leq 10^5$), o número de cartazes que você colocou no corredor e quantas vezes você mudou de direção respectivamente. Na segunda linha, há n inteiros, cada inteiro representa um número escrito no cartaz (observe que eles não estão em [ordem](#)!). Na terceira linha, há x inteiros, cada inteiro é um número escrito no cartaz no momento em que você decidiu mudar de posição. Considere que você começa sempre no cartaz de índice 1.

Saída

Sua [função](#) deve imprimir um número inteiro, quantos metros você correu no total.

Observações

- No primeiro caso de teste, o resultado é 4 pois no início você está no cartaz 10, corre até o cartaz 7, volta para o 3 e vai para o 7 ente, totalizando 4 metros percorridos.

Aviso!

?

For example:

Input	Result
3 3 10 3 7 7 3 7	4
7 4 58295989 78587259 37903322 32562633 75256060 27613507 7077302 78587259 75256060 7077302 27613507	7
9 4 2284868 66238128 95656092 47252657 60324444 95537730 76891229 24734104 59337742 2284868 24734104 60324444 60324444	10

Answer: (penalty regime: 0, 0, 10, 20, ... %)

1 ||

Para resolver o problema, deve-se salvar a posição do número do cartaz com o índice ao qual ele pertence, e então guardar isso num
Aviso! e calcular a diferença entre sua posição atual e a posição do índice do número do cartaz.

?

Question author's solution (Python3):

```
1 n, x = [int(i) for i in input().split()]
2 listaNum = [int(i) for i in input().split()]
3 listaX = [int(i) for i in input().split()]
4 posInicial = 1
5 total = 0
6 numeros = {}
7
8 for i in range(n):
9     numeros[listaNum[i]] = i+1
10
11 for cartaz in listaX:
12     total += abs(numeros[cartaz]-posInicial)
13     posInicial = numeros[cartaz]
14
15 print(total)
```

Aviso!

?

Questão 3

Correto

Atingiu 1,00 de 1,00

Pares Bonitos

Ângela gosta de coisas bonitas, incluindo números. Ela considera que pares de números são bonitos se um número for exatamente a metade do outro.

Por exemplo os pares (4, 2), (4, 8), (3, 6), (20, 10) são bonitos. Os pares (2, 5), (10, 40) não são.

Ângela considera que (4, 2) e (2, 4) são o mesmo par.

Dado um vetor de números, Ângela quer saber quantos pares bonitos ele contém. Como você é muito esperto, ela decidiu perguntar para você.

Entrada

A primeira linha da entrada é um número n ($2 \leq n \leq 10^5$), o tamanho do vetor de Ângela.

A segunda linha contém n inteiros ($1 \leq a_i \leq 10^9$) separados por espaço, que são os n itens do vetor.

Saída

Sua função deve imprimir um número inteiro que é a quantidade de pares bonitos no vetor.

Observações

- No primeiro caso de teste, o resultado é 2. Os dois pares bonitos do vetor são (1, 2) e (2, 4).
- No segundo caso de teste, o resultado é 0 porque não há pares bonitos no vetor.
- No terceiro caso de teste, o resultado é 4. Temos 2 pares (1, 2) e 2 pares (2, 4).

For example:

Input	Result
5	2

Aviso!

Input	Result
5 1 1 1 1 1	0
6 1 2 2 3 4 5	4

Answer: (penalty regime: 0, 0, 10, 20, ... %)

```

1 tamanho = int(input())
2 vetor = tuple(map(int, input().strip().split(' ')))
3 qtd_bonitos = 0
4
5 def getPairs(vetor):
6     duplas = []
7     for x in range(len(vetor)):
8         for y in range(x+1, len(vetor)):
9             duplas.append((vetor[x], vetor[y]))
10    return duplas
11
12 def isParBonito(par):
13     return (par[0] == par[1]/2) or (par[1] == par[0]/2)
14
15 for par in getPairs(vetor):

```

	Input	Expected	Got	
✓	5 1 2 3 4 5	2	2	✓
✓	5 1 1 1 1 1	0	0	✓
Aviso!	6 1 2 2 3 4 5	4	4	✓

?

	Input	Expected	Got	
✓	10 100 10 1 3 3 4 6 7 14 2	5	5	✓
✓	100 59341 56326 22737 98912 20041 8300 69144 79062 37242 2254 36789 44013 59717 54529 90180 15062 51577 91357 23139 81363 87925 14147 40997 18056 57035 3345 125 42561 89578 9494 70521 87078 77700 7101 64885 53955 96027 8534 62053 78810 14062 8512 40281 72735 25518 29188 44509 61420 68025 48215 45974 40720 86035 37666 22037 88134 1407 53973 19447 28502 4222 95563 81663 90304 29577 9802 96773 37594 3544 33438 1981 21736 88283 44927 78854 48092 75720 97598 33312 77517 53294 90992 89874 85516 77512 88205 113 88316 6768 51191 71023 11345 76493 15983 16969 79857 36254 69550 21427 45769	0	0	✓
✓	200 30862 15431 15431 15431 15431 15431 75698 37849 9394 4697 4697 4697 34844 17422 99948 49974 49974 49974 49974 49974 94122 47061 31764 15882 15882 15882 15882 46218 23109 23109 23109 23109 15586 7793 7793 63902 87134 43567 43567 43567 43567 43567 9396 4698 4698 4698 4698 71788 35894 35894 35894 35894 57506 28753 28753 60792 30396 30396 49086 98536 72746 36373 36373 36373 36373 5434 2717 13932 6966 6966 2254 88716 44358 41522 20761 20761 78710 68796 34398 34398 55526 27763 27763 27763 27763 27763 92064 46032 44318 7680 3840 21236 10618 10618 10618 10618 60546 30273 30273 30273 30273 30273 81230 40615 40615 40615 40642 69268 7692 3846 3846 57956 28978 28978 28978 19578 59902 29951 29951 29951 29951 29951 66896 33448 33448 33448 63492 31746 31746 31746 31746 31746 7696 3848 3848 81586 40793 41096 20548 92008 46004 46004 84780 42390 42390 42390 99016 49508 98338 49169 49169 49169 49169 78654 39327 39327 39327 76840 38420 38420 15424 7712 7712 7712 7712 73666 36833 36833 36833 36833 95736 47868 47868 47868 47868 84838 42419 71440 35720 35720 35720 35720 3036 86182 43091 43091 83098 41549 14026 7013 86028 43014 43014 43014 73092 85146 42573	139	139	✓

Passou em todos os teste! ✓

Para resolver o problema, devemos contar a frequência de cada número. Podemos usar um dicionário para isso.

Ao analisar um novo número a_i , olhamos no dicionário quantas vezes $\frac{a_i}{2}$ e $a_i * 2$ já apareceram. Essa é a quantidade de novos pares que podemos fazer com o novo número a_i considerando apenas os números já vistos. Somamos esses valores a uma variável que armazena o total de pares e essa será a resposta no final.

Por exemplo, podemos analisar o primeiro caso:

12345

Como a tabela abaixo mostra o valor da resposta acumulada até agora, o valor que será analisado e a situação do dicionário. A última linha mostra a resposta final e a configuração final do dicionário.

Aviso!

?

resposta: 0 - valor atual: 1 - dicionario: {}

resposta: 0 - valor atual: 2 - dicionario: {1: 1}

resposta: 1 - valor atual: 3 - dicionario: {1: 1, 2: 1}

resposta: 1 - valor atual: 4 - dicionario: {1: 1, 2: 1, 3: 1}

resposta: 2 - valor atual: 5 - dicionario: {1: 1, 2: 1, 3: 1, 4: 1}

resposta: 2 - dicionario: {1: 1, 2: 1, 3: 1, 4: 1, 5: 1}

Primeiro, ao analisar o 1, a resposta está zerada, pois nenhum par foi encontrado. O dicionário também está vazio porque nenhum número foi analisado ainda. Nem a metade de 1 (0.5) nem o dobro (2) foram vistos, então não adicionamos nada, mas salvamos que 1 foi visto 1 vez.

Ao analisar o número 2, vemos que a metade de 2 (1) já foi vista 1 vez, o que quer dizer que podemos criar o par (1, 2) 1 vez. Somamos à resposta 1 nesse caso, e guardamos o 2 no dicionário.

Assim por diante.

Question author's solution (Python3):

```
1 def ans(arr):
2     seen = {}
3     resp = 0
4     for n in arr:
5         resp += seen.get((n / 2), 0)
6         resp += seen.get((n * 2), 0)
7         seen[n] = seen.get(n, 0) + 1
8     return resp
9
10 n = int(input())
11 nums = list(map(int, input().split()))
12 print(ans(nums))
13
14 ''' Resposta alternativa:
15
```

Correto

Aviso! este envio: 1,00/1,00.

?

Aviso!

?

Questão 4

Correto

Atingiu 1,00 de 1,00

Voltando para a [biblioteca](#)

Crie um [programa](#) que contém um dicionário *biblioteca*. Leia uma [string](#) *livro*, que será uma das chaves da [biblioteca](#), e um número inteiro, que é o valor associado a chave lida anteriormente. Repita as leituras cinco vezes, ou seja, haverá cinco chaves no dicionário *biblioteca*.

Por fim, leia uma [string](#) *busca* e cheque se ela é uma chave em *biblioteca*.

Se sim, imprima: "Achei! Temos *número* exemplar(es)!"

Se não, imprima: "Poxa, não temos esse livro"

Entrada

A entrada consiste em cinco strings e cinco inteiros, e uma última [string](#) utilizada para procurar.

Saída

A saída consiste em uma sentença a depender da *busca*.

Observações

- No primeiro exemplo de teste, foram digitados os livros e o número de exemplares correspondentes a cada um. Por fim, foi digitado "Senhor dos Aneis", retornando "Achei! Temos 12 exemplar(es)!"
- No terceiro exemplo de teste, foram digitados os livros e o número de exemplares correspondentes a cada um. Por fim, foi digitado "Dininho", mas como não existe livro com esse nome no dicionário, é retornado "Poxa, não temos esse livro"

For example:

Aviso!	Result
--------	--------

Input	Result
Harry Potter 30 Guia do Mochileiro das Galaxias 42 Senhor dos Aneis 12 O Hobbit 5 Duna 13 Senhor dos Aneis	Achei! Temos 12 exemplar(es)!
Testando 12 Testando 2 31 Testando 3 9 Testando 4 1 Testando 5 2 Testando	Achei! Temos 12 exemplar(es)!
Dinossauros 34 Dinossauros Simpaticos 9 Dinossauros Batutas 13 Dinossauros Gigantes 39 Dino 1 Dininho	Poxa, não temos esse livro

Answer (penalty regime: 0, 0, 10, 20, ... %)

Aviso! biblioteca = dict()

2. LOTA E TVPOS = F



?


```

2 QTD_LIVROS = 5
3
4 for x in range(QTD_LIVROS):
5     livro = input()
6     qtd_livro = int(input())
7     biblioteca[livro] = int(qtd_livro)
8
9 livro_buscado = input()
10
11 if livro_buscado in biblioteca.keys():
12     print(f'Achei! Temos {biblioteca[livro_buscado]} exemplar(es)!')
13 else:
14     print('Poxa, não temos esse livro')
15

```

	Input	Expected	Got	
✓	Harry Potter 30 Guia do Mochileiro das Galaxias 42 Senhor dos Aneis 12 O Hobbit 5 Duna 13 Senhor dos Aneis	Achei! Temos 12 exemplar(es)!	Achei! Temos 12 exemplar(es)!	✓

Aviso!

?

	Input	Expected	Got	
✓	Testando 12 Testando 2 31 Testando 3 9 Testando 4 1 Testando 5 2 Testando	Achei! Temos 12 exemplar(es)!	Achei! Temos 12 exemplar(es)!	✓
✓	Dinossauros 34 Dinossauros Simpaticos 9 Dinossauros Batutas 13 Dinossauros Gigantes 39 Dino 1 Dininho	Poxa, não temos esse livro	Poxa, não temos esse livro	✓
✓	Como Pular Corda 12 Como dar No no Sapato 4 Como Andar de Bicicleta 43 Como Nadar 12 Como Jogar Xadrez 1 Como Comer Gelatina	Poxa, não temos esse livro	Poxa, não temos esse livro	✓

Aviso!

?

	Input	Expected	Got	
✓	Dom Casmurro 12 Quincas Borba 2 Papeis Avulsos 5 Memorias Postumas de Bras Cubas 7 Esau e Jaco 3 Quincas Borba	Achei! Temos 2 exemplar(es)!	Achei! Temos 2 exemplar(es)!	✓
✓	Oi 1 Ola 2 Como vai? 3 Tudo certo 4 Valeu, falou 5 Tchau	Poxa, não temos esse livro	Poxa, não temos esse livro	✓

Passou em todos os teste! ✓

Question author's solution (Python3):

```

1 biblioteca = {}
2 for i in range(5):
3     livro = input()
4     biblioteca[livro] = int(input())
5
6 busca = input()
7 if busca in biblioteca:
    print("Achei! Temos", biblioteca[busca], "exemplar(es)!")

```

Aviso!

?

Correto

Notas para este envio: 1,00/1,00.

Aviso!

?

Questão 5

Parcialmente correto

Atingiu 0,86 de 1,00

Amigos

Danusa, uma professora do pré, estava curiosa para saber quantos amiguinhos cada um de seus alunos tem. Ela passou a semana toda anotando quem é amigo de quem na classe, mas cuidar de crianças não é nada fácil, então Danusa ainda não conseguiu parar para tirar conclusões das suas anotações. Ela pediu ajuda a você, que é um universitário conhecedor de [algoritmos](#).

Pontos importantes:

- Se A é amigo de B , então B também é amigo de A
- Se A é amigo de B e B é amigo de C , então A também é amigo de C .

Dada a lista de alunos na classe de Danusa, e pares de amigos, descubra quantos amigos cada aluno tem.

Entrada

A primeira linha da entrada possui 2 inteiros, n e p , ($2 \leq n \leq 25$), ($1 \leq p \leq n * (n - 1)$). n é o número de alunos na sala, e p o número de pares de amigos que Danusa anotou na semana.

A segunda linha contém n strings, os nomes de cada aluno.

Após, p linhas seguem. Cada linha possui duas strings a e b ($a \neq b$), o nome de 2 alunos que são amigos.

Saída

Sua [função](#) deve imprimir n linhas. Em cada linha imprima " n_i possui x amigos" onde n_i é o nome de um dos alunos e x o número de amigos que esse aluno possui na turma.

Imprima o nome dos alunos em [ordem](#) alfabética.

Observações

- No primeiro caso de teste, todos são amigos de todos (pela transitividade da amizade), então das 3 pessoas, tirando você mesmo, sobram 2 amigos.

Aviso!

No segundo caso de teste, João e Pedro formam um círculo de amizade com 2 pessoas no total, então cada um tem 1 amigo. Não é uma coisa para Maria e Miriam.

- O terceiro caso é semelhante ao segundo, mas o círculo de amizade João, Pedro , Matheus possui 3 pessoas, então são 2 amigos para cada uma.

For example:

Input	Result
3 2 João Maria Miriam João Maria Maria Miriam	João possui 2 amigos Maria possui 2 amigos Miriam possui 2 amigos
4 2 João Pedro Maria Miriam João Pedro Maria Miriam	João possui 1 amigos Maria possui 1 amigos Miriam possui 1 amigos Pedro possui 1 amigos
5 3 João Maria Miriam Pedro Matheus João Pedro Maria Miriam Pedro Matheus	João possui 2 amigos Maria possui 1 amigos Matheus possui 2 amigos Miriam possui 1 amigos Pedro possui 2 amigos

Answer: (penalty regime: 0, 0, 10, 20, ... %)

```

1 import itertools
2
3 def getDuplicates(list1, list2):
4     return list(set(list1) & set(list2))
5
6 def mergeLists(list1, list2):
7     return list1 + list(set(list2) - set(list1))
8
9 def getSublists(lista, size):
10    return list(itertools.combinations(set(lista), size))
11
12

```

Aviso! qtd_alunos, qtd_pares = map(int, input().strip().split(' '))
 nomes = input().strip().split(' ')

15

	Input	Expected	Got	
✓	3 2 João Maria Miriam João Maria Maria Miriam	João possui 2 amigos Maria possui 2 amigos Miriam possui 2 amigos	João possui 2 amigos Maria possui 2 amigos Miriam possui 2 amigos	✓
✓	4 2 João Pedro Maria Miriam João Pedro Maria Miriam	João possui 1 amigos Maria possui 1 amigos Miriam possui 1 amigos Pedro possui 1 amigos	João possui 1 amigos Maria possui 1 amigos Miriam possui 1 amigos Pedro possui 1 amigos	✓
✓	5 3 João Maria Miriam Pedro Matheus João Pedro Maria Miriam Pedro Matheus	João possui 2 amigos Maria possui 1 amigos Matheus possui 2 amigos Miriam possui 1 amigos Pedro possui 2 amigos	João possui 2 amigos Maria possui 1 amigos Matheus possui 2 amigos Miriam possui 1 amigos Pedro possui 2 amigos	✓

Aviso!

?

	Input	Expected	Got	
✓	10 5 Bruni Batatinha Brad Chef Tom Felix João Maria Mia Shawn Chef Brad Maria Tom Bruni Maria Bruni Mia Chef Shawn	Batatinha possui 0 amigos Brad possui 2 amigos Bruni possui 3 amigos Chef possui 2 amigos Felix possui 0 amigos João possui 0 amigos Maria possui 3 amigos Mia possui 3 amigos Shawn possui 2 amigos Tom possui 3 amigos	Batatinha possui 0 amigos Brad possui 2 amigos Bruni possui 3 amigos Chef possui 2 amigos Felix possui 0 amigos João possui 0 amigos Maria possui 3 amigos Mia possui 3 amigos Shawn possui 2 amigos Tom possui 3 amigos	✓
✓	10 8 Mango Banana Guerreira Dune João Frajola Greg Xena Fifi Criss Mango Frajola Fifi Dune Xena Mango Xena Banana Banana Criss Mango Greg Criss Mango Fifi João	Banana possui 5 amigos Criss possui 5 amigos Dune possui 2 amigos Fifi possui 2 amigos Frajola possui 5 amigos Greg possui 5 amigos Guerreira possui 0 amigos João possui 2 amigos Mango possui 5 amigos Xena possui 5 amigos	Banana possui 5 amigos Criss possui 5 amigos Dune possui 2 amigos Fifi possui 2 amigos Frajola possui 5 amigos Greg possui 5 amigos Guerreira possui 0 amigos João possui 2 amigos Mango possui 5 amigos Xena possui 5 amigos	✓

Aviso!

?

	Input	Expected	Got	
✓	15 10 Kiwi Greg Bruna Hunter Khali Mandachuva Luna Didi Matheus Maria João Lary Gatinho Bob Xanim Mandachuva Xanim Didi Greg João Bruna Gatinho Khali Kiwi Bruna Gatinho Luna Gatinho Mandachuva Matheus Mandachuva Hunter Bruna Kiwi Khali	Bob possui 0 amigos Bruna possui 9 amigos Didi possui 1 amigos Gatinho possui 9 amigos Greg possui 1 amigos Hunter possui 9 amigos João possui 9 amigos Khali possui 9 amigos Kiwi possui 9 amigos Lary possui 0 amigos Luna possui 9 amigos Mandachuva possui 9 amigos Maria possui 0 amigos Matheus possui 9 amigos Xanim possui 9 amigos	Bob possui 0 amigos Bruna possui 9 amigos Didi possui 1 amigos Gatinho possui 9 amigos Greg possui 1 amigos Hunter possui 9 amigos João possui 9 amigos Khali possui 9 amigos Kiwi possui 9 amigos Lary possui 0 amigos Luna possui 9 amigos Mandachuva possui 9 amigos Maria possui 0 amigos Matheus possui 9 amigos Xanim possui 9 amigos	✓

Your code failed one or more hidden tests.

Algo a se perceber antes de começar a resolver o problema é que a amizade é transitiva, isto é, se a possui um círculo de amizade, e b outro, caso vejamos que a é amigo de b , então ele não ganha apenas mais 1 amigo, e sim a quantidade de pessoas que estiverem no círculo de amizade de b até o momento. Os dois círculos de amizade efetivamente se fundem em um único círculo de amizade maior.

Uma maneira de resolver a questão é representar o círculo de amizades de cada aluno por um dicionário. Como dicionários não têm elementos repetidos, não precisamos nos preocupar com a possibilidade de adicionar amigos repetidos a um círculo de amizade.

Inicialmente os alunos não têm amigos, então podemos representar os círculos de amizades por dicionários cuja única entrada é o próprio aluno.

Aviso! cada nova amizade, adicionamos os amigos de um ao círculo de amizades do outro. Em seguida igualamos os círculos de
os alunos que são amigos desse.

Ao final da execução o número de amigos de um determinado aluno será o tamanho do seu círculo de amizades menos 1, pois o próprio aluno faz parte desse círculo.

Question author's solution (Python3):

```
1 n_alunos, n_amizades = map(int, input().split())
2 alunos = input().split()
3
4 amizades = dict()
5 for aluno in alunos:
6     amizades[aluno] = dict({aluno : None}) #dicionário de dicionários
7
8 for _ in range(n_amizades):
9     amigo1, amigo2 = input().split()
10    amizades[amigo1].update(amizades[amigo2]) #bota as chaves de amizades[amigo2] que não
11    for aluno in amizades[amigo1]:
12        amizades[amigo1].update(amizades[aluno]) #o circulo de amizade de a1 é atualizado
13    for aluno in amizades[amigo1]:
14
```

Parcialmente correto

Notas para este envio: 0,86/1,00.

Aviso!

?

Questão 6

Correto

Atingiu 1,00 de 1,00

Frequência Fácil

Dada uma cadeia de caracteres contendo apenas letras minúsculas e números, conte a frequência de cada caractere e imprima as seguindo a [ordem](#) dada no exemplo.

Entrada

Contem uma unica [string](#) *s*.

Saída

Imprima a frequência dos caracteres em [ordem](#) alfabética e depois numérica. Observe os exemplos.

Observações

For example:

Input	Result
-------	--------

Aviso!

?

Input	Result
abc12	0 caractere a aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere b aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere c aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere d aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere e aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere f aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere g aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere h aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere i aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere j aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere k aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere l aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere m aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere n aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere o aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere p aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere q aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere r aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere s aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere t aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere u aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere v aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere w aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere x aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere y aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere z aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 0 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 1 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 2 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 3 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 4 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 5 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 6 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 7 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 8 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 9 aparece 0 vez(es) na cadeia lida pelo programa :)

Aviso!

?

Input	Result
adg23	0 caractere a aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere b aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere c aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere d aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere e aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere f aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere g aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere h aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere i aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere j aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere k aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere l aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere m aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere n aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere o aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere p aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere q aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere r aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere s aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere t aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere u aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere v aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere w aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere x aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere y aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere z aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 0 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 1 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 2 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 3 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 4 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 5 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 6 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 7 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 8 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 9 aparece 0 vez(es) na cadeia lida pelo programa :)

Aviso!

?

Input	Result
abcdefghijkl123	0 caractere a aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere b aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere c aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere d aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere e aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere f aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere g aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere h aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere i aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere j aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere k aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere l aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere m aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere n aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere o aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere p aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere q aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere r aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere s aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere t aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere u aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere v aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere w aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere x aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere y aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere z aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 0 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 1 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 2 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 3 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 4 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 5 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 6 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 7 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 8 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 9 aparece 0 vez(es) na cadeia lida pelo programa :)

Aviso! penalty regime: 0, 0, 10, 20, ... %)

```

1 | CARACTERES = tuple('abcdefghijklmnopqrstuvwxyz0123456789')
2 | dicionario_chars = dict()
3 |
4 | texto = input().strip()
5 |
6 | for char in CARACTERES:
7 |     dicionario_chars[char] = texto.count(char)
8 |
9 | for char in dicionario_chars:
10 |     print(f'0 caractere {char} aparece {dicionario_chars[char]} vez(es) na cadeia lida pelo
11 |
12 |
13 |
14 |

```

	Input	Expected	Got	
✓	abc12	0 caractere a aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere b aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere c aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere	0 caractere a aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere b aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere c aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere	✓

Aviso!

?

	Input	Expected	Got	
		d aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere e aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere f aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere g aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere h aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere i aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere j aparece 0 vez(es) na cadeia lida pelo programa :)	d aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere e aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere f aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere g aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere h aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere i aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere j aparece 0 vez(es) na cadeia lida pelo programa :)	

Aviso!

?

	Input	Expected	Got	
		0 caractere k aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere l aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere m aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere n aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere o aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere p aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere q aparece 0 vez(es) na cadeia lida pelo	0 caractere k aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere l aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere m aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere n aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere o aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere p aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere q aparece 0 vez(es) na cadeia lida pelo	

Aviso!

?

	Input	Expected	Got	
		programa :) 0 caractere r aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere s aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere t aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere u aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere v aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere w aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere x aparece 0 vez(es) na cadeia lida	programa :) 0 caractere r aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere s aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere t aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere u aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere v aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere w aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere x aparece 0 vez(es) na cadeia lida	

Aviso!

?

	Input	Expected	Got	
		pelo programa :) 0 caractere y aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere z aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 0 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 1 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 2 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 3 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 4 aparece 0 vez(es) na	pelo programa :) 0 caractere y aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere z aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 0 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 1 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 2 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 3 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 4 aparece 0 vez(es) na	

Aviso!

?

	Input	Expected	Got	
		cadeia lida pelo programa :) 0 caractere 5 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 6 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 7 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 8 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 9 aparece 0 vez(es) na cadeia lida pelo programa :)	cadeia lida pelo programa :) 0 caractere 5 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 6 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 7 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 8 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 9 aparece 0 vez(es) na cadeia lida pelo programa :)	
✓	adg23	0 caractere a aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere	0 caractere a aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere	✓

Aviso!

?

	Input	Expected	Got	
		b aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere c aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere d aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere e aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere f aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere g aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere h aparece 0 vez(es) na cadeia lida pelo programa :)	b aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere c aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere d aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere e aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere f aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere g aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere h aparece 0 vez(es) na cadeia lida pelo programa :)	

Aviso!

?

	Input	Expected	Got	
		0 caractere i aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere j aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere k aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere l aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere m aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere n aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere o aparece 0 vez(es) na cadeia lida pelo	0 caractere i aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere j aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere k aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere l aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere m aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere n aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere o aparece 0 vez(es) na cadeia lida pelo	

Aviso!

?

	Input	Expected	Got	
<div data-bbox="248 1362 367 1433">Aviso!</div>		programa :) 0 caractere p aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere q aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere r aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere s aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere t aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere u aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere v aparece 0 vez(es) na cadeia lida	programa :) 0 caractere p aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere q aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere r aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere s aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere t aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere u aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere v aparece 0 vez(es) na cadeia lida	

	Input	Expected	Got	
		pelo programa :) 0 caractere w aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere x aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere y aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere z aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 0 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 1 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 2 aparece 1 vez(es) na	pelo programa :) 0 caractere w aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere x aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere y aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere z aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 0 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 1 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 2 aparece 1 vez(es) na	

Aviso!

?

	Input	Expected	Got	
		cadeia lida pelo programa :) 0 caractere 3 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 4 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 5 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 6 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 7 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 8 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 9 aparece 0	cadeia lida pelo programa :) 0 caractere 3 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 4 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 5 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 6 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 7 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 8 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 9 aparece 0	

Aviso!

?

	Input	Expected	Got	
		vez(es) na cadeia lida pelo programa :)	vez(es) na cadeia lida pelo programa :)	
✓	abcdefghijkl123	0 caractere a aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere b aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere c aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere d aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere e aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere f aparece 1 vez(es) na cadeia lida pelo programa :)	0 caractere a aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere b aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere c aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere d aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere e aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere f aparece 1 vez(es) na cadeia lida pelo programa :)	✓

Aviso!

?

	Input	Expected	Got	
		0 caractere g aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere h aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere i aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere j aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere k aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere l aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere m aparece 0 vez(es) na cadeia lida pelo	0 caractere g aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere h aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere i aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere j aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere k aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere l aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere m aparece 0 vez(es) na cadeia lida pelo	

Aviso!

?

	Input	Expected	Got	
		programa :) 0 caractere n aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere o aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere p aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere q aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere r aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere s aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere t aparece 0 vez(es) na cadeia lida	programa :) 0 caractere n aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere o aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere p aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere q aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere r aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere s aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere t aparece 0 vez(es) na cadeia lida	

Aviso!

?

	Input	Expected	Got	
		pelo programa :) 0 caractere u aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere v aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere w aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere x aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere y aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere z aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 0 aparece 0 vez(es) na	pelo programa :) 0 caractere u aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere v aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere w aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere x aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere y aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere z aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 0 aparece 0 vez(es) na	

Aviso!

?

	Input	Expected	Got	
		cadeia lida pelo programa :) 0 caractere 1 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 2 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 3 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 4 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 5 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 6 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 7 aparece 0	cadeia lida pelo programa :) 0 caractere 1 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 2 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 3 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 4 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 5 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 6 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 7 aparece 0	

Aviso!

?

	Input	Expected	Got	
		vez(es) na cadeia lida pelo programa :) 0 caractere 8 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 9 aparece 0 vez(es) na cadeia lida pelo programa :)	vez(es) na cadeia lida pelo programa :) 0 caractere 8 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 9 aparece 0 vez(es) na cadeia lida pelo programa :)	
✓	abcdefghijklmnopqrstuvwxyz012345678910942iugasjidhfkjasdhfjqwhqoiuehfq93yf93hefqwihqiu	0 caractere a aparece 3 vez(es) na cadeia lida pelo programa :) 0 caractere b aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere c aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere d aparece 3 vez(es) na cadeia lida pelo programa :)	0 caractere a aparece 3 vez(es) na cadeia lida pelo programa :) 0 caractere b aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere c aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere d aparece 3 vez(es) na cadeia lida pelo programa :)	✓

Aviso!

?

	Input	Expected	Got	
		0 caractere e aparece 3 vez(es) na cadeia lida pelo programa :) 0 caractere f aparece 6 vez(es) na cadeia lida pelo programa :) 0 caractere g aparece 2 vez(es) na cadeia lida pelo programa :) 0 caractere h aparece 7 vez(es) na cadeia lida pelo programa :) 0 caractere i aparece 6 vez(es) na cadeia lida pelo programa :) 0 caractere j aparece 4 vez(es) na cadeia lida pelo programa :) 0 caractere k aparece 2 vez(es) na cadeia lida pelo	0 caractere e aparece 3 vez(es) na cadeia lida pelo programa :) 0 caractere f aparece 6 vez(es) na cadeia lida pelo programa :) 0 caractere g aparece 2 vez(es) na cadeia lida pelo programa :) 0 caractere h aparece 7 vez(es) na cadeia lida pelo programa :) 0 caractere i aparece 6 vez(es) na cadeia lida pelo programa :) 0 caractere j aparece 4 vez(es) na cadeia lida pelo programa :) 0 caractere k aparece 2 vez(es) na cadeia lida pelo	

Aviso!

?

	Input	Expected	Got	
		programa :) 0 caractere l aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere m aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere n aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere o aparece 3 vez(es) na cadeia lida pelo programa :) 0 caractere p aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere q aparece 6 vez(es) na cadeia lida pelo programa :) 0 caractere r aparece 1 vez(es) na cadeia lida	programa :) 0 caractere l aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere m aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere n aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere o aparece 3 vez(es) na cadeia lida pelo programa :) 0 caractere p aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere q aparece 6 vez(es) na cadeia lida pelo programa :) 0 caractere r aparece 1 vez(es) na cadeia lida	

Aviso!

?

	Input	Expected	Got	
		pelo programa :) 0 caractere s aparece 3 vez(es) na cadeia lida pelo programa :) 0 caractere t aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere u aparece 5 vez(es) na cadeia lida pelo programa :) 0 caractere v aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere w aparece 3 vez(es) na cadeia lida pelo programa :) 0 caractere x aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere y aparece 2 vez(es) na	pelo programa :) 0 caractere s aparece 3 vez(es) na cadeia lida pelo programa :) 0 caractere t aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere u aparece 5 vez(es) na cadeia lida pelo programa :) 0 caractere v aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere w aparece 3 vez(es) na cadeia lida pelo programa :) 0 caractere x aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere y aparece 2 vez(es) na	

Aviso!

?

	Input	Expected	Got	
		cadeia lida pelo programa :) 0 caractere z aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 0 aparece 2 vez(es) na cadeia lida pelo programa :) 0 caractere 1 aparece 2 vez(es) na cadeia lida pelo programa :) 0 caractere 2 aparece 2 vez(es) na cadeia lida pelo programa :) 0 caractere 3 aparece 3 vez(es) na cadeia lida pelo programa :) 0 caractere 4 aparece 2 vez(es) na cadeia lida pelo programa :) 0 caractere 5 aparece 1	cadeia lida pelo programa :) 0 caractere z aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 0 aparece 2 vez(es) na cadeia lida pelo programa :) 0 caractere 1 aparece 2 vez(es) na cadeia lida pelo programa :) 0 caractere 2 aparece 2 vez(es) na cadeia lida pelo programa :) 0 caractere 3 aparece 3 vez(es) na cadeia lida pelo programa :) 0 caractere 4 aparece 2 vez(es) na cadeia lida pelo programa :) 0 caractere 5 aparece 1	

Aviso!

?

	Input	Expected	Got	
		vez(es) na cadeia lida pelo programa :) 0 caractere 6 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 7 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 8 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 9 aparece 4 vez(es) na cadeia lida pelo programa :)	vez(es) na cadeia lida pelo programa :) 0 caractere 6 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 7 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 8 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 9 aparece 4 vez(es) na cadeia lida pelo programa :)	
✓	dsajhgwjdhf92uy91y3uyfh3109fuyh3091urfhq09uyrtiuh sdfjkadf qoer190u	0 caractere a aparece 2 vez(es) na cadeia lida pelo programa :) 0 caractere b aparece 0 vez(es) na cadeia lida pelo programa :)	0 caractere a aparece 2 vez(es) na cadeia lida pelo programa :) 0 caractere b aparece 0 vez(es) na cadeia lida pelo programa :)	✓

Aviso!

?

	Input	Expected	Got	
		0 caractere c aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere d aparece 4 vez(es) na cadeia lida pelo programa :) 0 caractere e aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere f aparece 6 vez(es) na cadeia lida pelo programa :) 0 caractere g aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere h aparece 6 vez(es) na cadeia lida pelo programa :) 0 caractere i aparece 1 vez(es) na cadeia lida pelo	0 caractere c aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere d aparece 4 vez(es) na cadeia lida pelo programa :) 0 caractere e aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere f aparece 6 vez(es) na cadeia lida pelo programa :) 0 caractere g aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere h aparece 6 vez(es) na cadeia lida pelo programa :) 0 caractere i aparece 1 vez(es) na cadeia lida pelo	

Aviso!

?

	Input	Expected	Got	
		programa :) 0 caractere j aparece 3 vez(es) na cadeia lida pelo programa :) 0 caractere k aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere l aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere m aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere n aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere o aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere p aparece 0 vez(es) na cadeia lida	programa :) 0 caractere j aparece 3 vez(es) na cadeia lida pelo programa :) 0 caractere k aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere l aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere m aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere n aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere o aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere p aparece 0 vez(es) na cadeia lida	

Aviso!

?

	Input	Expected	Got	
		pelo programa :) 0 caractere q aparece 2 vez(es) na cadeia lida pelo programa :) 0 caractere r aparece 3 vez(es) na cadeia lida pelo programa :) 0 caractere s aparece 2 vez(es) na cadeia lida pelo programa :) 0 caractere t aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere u aparece 7 vez(es) na cadeia lida pelo programa :) 0 caractere v aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere w aparece 1 vez(es) na	pelo programa :) 0 caractere q aparece 2 vez(es) na cadeia lida pelo programa :) 0 caractere r aparece 3 vez(es) na cadeia lida pelo programa :) 0 caractere s aparece 2 vez(es) na cadeia lida pelo programa :) 0 caractere t aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere u aparece 7 vez(es) na cadeia lida pelo programa :) 0 caractere v aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere w aparece 1 vez(es) na	

Aviso!

?

	Input	Expected	Got	
		cadeia lida pelo programa :) 0 caractere x aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere y aparece 5 vez(es) na cadeia lida pelo programa :) 0 caractere z aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 0 aparece 4 vez(es) na cadeia lida pelo programa :) 0 caractere 1 aparece 4 vez(es) na cadeia lida pelo programa :) 0 caractere 2 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 3 aparece 3	cadeia lida pelo programa :) 0 caractere x aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere y aparece 5 vez(es) na cadeia lida pelo programa :) 0 caractere z aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 0 aparece 4 vez(es) na cadeia lida pelo programa :) 0 caractere 1 aparece 4 vez(es) na cadeia lida pelo programa :) 0 caractere 2 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 3 aparece 3	

Aviso!

?

	Input	Expected	Got	
		vez(es) na cadeia lida pelo programa :) 0 caractere 4 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 5 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 6 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 7 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 8 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 9 aparece 6 vez(es) na cadeia lida pelo programa :)	vez(es) na cadeia lida pelo programa :) 0 caractere 4 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 5 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 6 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 7 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 8 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 9 aparece 6 vez(es) na cadeia lida pelo programa :)	

Aviso!

?

	Input	Expected	Got	
✓	tozt0dkt1l3ax860q12o	0 caractere a aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere b aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere c aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere d aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere e aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere f aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere g aparece 0 vez(es) na cadeia lida pelo	0 caractere a aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere b aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere c aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere d aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere e aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere f aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere g aparece 0 vez(es) na cadeia lida pelo	✓

Aviso!

?

	Input	Expected	Got	
		programa :) 0 caractere h aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere i aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere j aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere k aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere l aparece 2 vez(es) na cadeia lida pelo programa :) 0 caractere m aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere n aparece 0 vez(es) na cadeia lida	programa :) 0 caractere h aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere i aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere j aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere k aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere l aparece 2 vez(es) na cadeia lida pelo programa :) 0 caractere m aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere n aparece 0 vez(es) na cadeia lida	

Aviso!

?

	Input	Expected	Got	
		pelo programa :) 0 caractere o aparece 2 vez(es) na cadeia lida pelo programa :) 0 caractere p aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere q aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere r aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere s aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere t aparece 3 vez(es) na cadeia lida pelo programa :) 0 caractere u aparece 0 vez(es) na	pelo programa :) 0 caractere o aparece 2 vez(es) na cadeia lida pelo programa :) 0 caractere p aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere q aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere r aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere s aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere t aparece 3 vez(es) na cadeia lida pelo programa :) 0 caractere u aparece 0 vez(es) na	

Aviso!

?

	Input	Expected	Got	
		cadeia lida pelo programa :) 0 caractere v aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere w aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere x aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere y aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere z aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 0 aparece 2 vez(es) na cadeia lida pelo programa :) 0 caractere 1 aparece 1	cadeia lida pelo programa :) 0 caractere v aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere w aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere x aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere y aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere z aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 0 aparece 2 vez(es) na cadeia lida pelo programa :) 0 caractere 1 aparece 1	

Aviso!

?

	Input	Expected	Got	
		vez(es) na cadeia lida pelo programa :) 0 caractere 2 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 3 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 4 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 5 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 6 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 7 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere	vez(es) na cadeia lida pelo programa :) 0 caractere 2 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 3 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 4 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 5 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere 6 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 7 aparece 0 vez(es) na cadeia lida pelo programa :) 0 caractere	

Aviso!

?

	Input	Expected	Got	
		8 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 9 aparece 0 vez(es) na cadeia lida pelo programa :)	8 aparece 1 vez(es) na cadeia lida pelo programa :) 0 caractere 9 aparece 0 vez(es) na cadeia lida pelo programa :)	

Passou em todos os teste! ✓

Para resolver o problema, deve-se ler a cadeia e percorrer um a um verificando quais elementos estão repetidos.

Question author's solution (Python3):

```
1 cadeia = input()
2
3 freq = {}
4 alfabeto = "abcdefghijklmnopqrstuvwxyz0123456789"
5
6 for ch in alfabeto:
7     freq[ch] = 0
8
9 for ch in cadeia:
10    freq[ch] += 1
```

Correto

Notas para este envio: 1,00/1,00.

Aviso!

?

Questão 7

Parcialmente correto

Atingiu 0,29 de 1,00

Ajuda aí!

Dêivis é um menino muito travesso e gosta de brincar com [vetores](#). Um dia Dêivis tinha um [vetor](#) de inteiros e se fez duas perguntas, quantos números y eu tenho dentro desse [vetor](#)? E a outra, qual a primeira posição em que y aparece? Enfim, como você sabe, Dêivis não é muito esperto e estava tendo dificuldades em resolver essa questão. Como você achou esse problema muito fácil, resolveu ajudar Dêivis a descobrir a resposta para os problemas.

Entrada

A primeira linha consiste no inteiro n ($1 \leq n \leq 10^5$), o tamanho do [vetor](#) de Dêivis. A segunda linha contém n inteiros, os elementos do [vetor](#) de Dêivis. A terceira linha contém um inteiro q ($1 \leq q \leq 10^5$), a quantidade de perguntas que Dêivis irá fazer. As próximas q linhas conterão dois inteiros x, y cada, se $x = 1$ então responda quantas ocorrências de y há no [vetor](#) e se $x = 2$ responda qual a posição do primeiro y no [vetor](#). Imprima -1 caso o elemento não esteja no [vetor](#) em ambos os tipos de pergunta. (Imprima a posição considerando o 0 como o primeiro elemento!)

Saída

Sua saída deverá conter q linhas. Para cada linha, imprima as perguntas feitas por Dêivis na [ordem](#) que foram dadas.

Observações

Atenção: implementar uma solução para esse problema utilizando listas levará ao *Time Limit Exceeded* (TLE). Por isso, utilize dicionários na sua solução.

- No primeiro exemplo de teste, o resultado é 0 pois a posição da primeira ocorrência de 1 é 0.

For example:

Aviso!

Result

?

Input	Result
2 1 1 1 2 1	0
10 9 10 5 1 2 7 8 10 1 10 6 1 10 1 8 2 2 1 5 1 10 2 5	3 1 4 1 3 2
4 1 4 1 3 5 1 3 2 4 1 1 2 3 2 1	1 1 2 3 0

Answer: (penalty regime: 0, 0, 10, 20, ... %)

```

1 tamanho = int(input())
2 lista = list(map(int, input().strip().split(' ')))
3 vetor = {
4     i : lista[i] for i in range(tamanho)
5 }
6
7 qtd_perguntas = int(input())
8
9
10 def getPosicao(numero, dicionario):
11     for key in dicionario:
12         if dicionario[key] == numero:

```

Aviso!

?

```

12 |         if dicionario[key] == numero:
13 |             return key
14 |         return -1
15 |

```

	Input	Expected	Got	
✓	2 1 1 1 2 1	0	0	✓
✓	10 9 10 5 1 2 7 8 10 1 10 6 1 10 1 8 2 2 1 5 1 10 2 5	3 1 4 1 3 2	3 1 4 1 3 2	✓
✓	4 1 4 1 3 5 1 3 2 4 1 1 2 3 2 1	1 1 2 3 0	1 1 2 3 0	✓
✓	97 1 52 14 25 65 8 85 57 78 61 81 44 97 75 93 85 64 93 82 35 19 81 28 91 42 84 78 96 27 54 68 5 92 88 73 70 10 29 34 57 59 9 82 48 20 8 31 68 30 80 6 51 29 88 7 37 44 82 46 33 67 6 19 45 38 20 75 51 8 50 83 17 84 13 68 53 12 93 15 44 60 47 9 43 56 50 60 15 71 14 10 91 35 56 39 93 62 60 2 80 37 7	49 1 54 2 2 75 4 96 38	49 1 54 2 2 75 4 96 38	✓

Aviso!

?

	Input	Expected	Got	
	1 75	78	78	
	1 60	1	1	
	2 53	1	1	
	2 65	2	2	
	2 62	3	3	
	2 34	1	1	
	2 15	29	29	
	1 12	37	37	
	1 64	8	8	
	1 51	32	32	
	1 44	2	2	
	1 97	80	80	
	2 54	46	46	
	2 29	2	2	
	2 78	14	14	
	2 92	11	11	
	1 78	59	59	
	2 60	1	1	
	2 31	23	23	
	1 75	1	1	
	2 93	23	23	
	2 44	2	2	
	2 33	2	2	
	1 39	2	2	
	2 91	2	2	
	1 97	80	80	
	2 91	2	2	
	1 29	3	3	
	1 6	36	36	
	1 78	1	1	
	1 88	2	2	
	2 60	94	94	
	1 88	25	25	
	1 8	41	41	
	2 10	2	2	
	1 97	63	63	
	1 10	14	14	
	2 39	50	50	
	2 84	2	2	
	2 9	2	2	
	2 51	7	7	

Aviso!

?

	Input	Expected	Got	
	2 45 2 93 2 6 1 6 1 78 2 57 1 20 1 9 2 8 1 96 2 17 1 62 1 97 2 85 1 91 2 97	2 2 5 1 71 1 1 6 2 12	2 2 5 1 71 1 1 6 2 12	
✓	279 233 112 203 166 200 101 164 264 252 34 48 237 9 76 221 218 214 47 192 10 221 206 232 265 215 239 86 78 177 27 74 134 85 108 204 27 198 110 36 166 258 142 125 99 187 154 31 125 115 226 53 66 160 151 237 146 257 22 168 82 114 42 270 157 133 265 99 24 96 106 100 205 42 8 106 134 10 266 20 188 47 52 250 73 28 196 228 236 259 101 233 224 248 210 136 205 7 90 12 175 130 231 118 90 180 24 212 2 244 179 177 260 278 26 145 270 129 31 210 230 93 152 40 275 171 179 68 10 207 248 65 207 250 238 131 84 209 208 33 197 193 192 94 93 8 239 61 277 252 166 168 39 169 206 213 115 40 265 263 89 248 103 121 80 60 250 194 163 61 151 182 221 244 72 42 29 215 231 185 227 211 177 244 217 234 19 259 127 259 248 128 266 76 206 20 206 12 170 48 274 165 84 164 222 40 265 196 121 34 166 46 212 178 13 226 139 43 47 101 133 38 237 13 269 30 252 125 106 100 175 176 109 247 234 164 212 70 188 57 115 45 95 210 180 17 13 177 101 182 151 155 20 230 277 138 24 258 216 123 8 6 128 69 220 132 52 235 213 123 41 223 154 227 125 246 161 67 246 79 385 2 212 2 31 2 221 1 101 1 78 1 244 1 270 1 33 277 125	106 46 14 4 1 3 2 1 2 4 2 4 2 1 3 4 101 1 2 21 158 1 1 1 2 1	106 46 14 4 1 3 2 1 2 4 2 4 2 1 3 4 101 1 2 21 158 1 1 1 2 1	✓

Aviso!

?

	Input	Expected	Got	
	1 258	2	2	
	1 73	4	4	
	1 151	166	166	
	1 101	2	2	
	2 231	2	2	
	1 165	106	106	
	1 207	161	161	
	2 206	107	107	
	2 263	213	213	
	1 112	4	4	
	1 41	102	102	
	1 264	1	1	
	1 188	1	1	
	1 274	2	2	
	1 34	190	190	
	1 206	21	21	
	2 194	197	197	
	1 61	185	185	
	1 123	-1	-1	
	2 212	31	31	
	2 103	73	73	
	2 2	2	2	
	2 13	3	3	
	1 206	183	183	
	2 118	1	1	
	1 82	1	1	
	1 36	3	3	
	1 76	216	216	
	2 128	1	1	
	2 206	2	2	
	2 170	2	2	
	2 19	31	31	
	2 279	1	1	
	2 134	135	135	
	2 8	1	1	
	1 192	4	4	
	1 10	91	91	
	2 217	260	260	
	1 86	2	2	
	1 169	3	3	
	1 210	1	1	

Aviso!

?

	Input	Expected	Got	
	2 43	3	3	
	1 160	1	1	
	1 128	3	3	
	1 34	1	1	
	2 134	58	58	
	1 165	2	2	
	2 84	190	190	
	1 193	28	28	
	1 177	43	43	
	2 224	2	2	
	2 6	11	11	
	1 12	108	108	
	1 259	3	3	
	1 110	94	94	
	2 166	1	1	
	1 171	97	97	
	1 47	3	3	
	1 130	1	1	
	2 168	97	97	
	1 179	10	10	
	2 128	3	3	
	2 177	3	3	
	2 99	107	107	
	1 226	159	159	
	2 237	1	1	
	2 244	28	28	
	1 40	1	1	
	2 136	162	162	
	1 235	104	104	
	2 90	2	2	
	1 47	1	1	
	1 157	2	2	
	2 90	113	113	
	2 48	70	70	
	1 212	1	1	
	1 20	18	18	
	2 2	59	59	
	2 89	8	8	
	1 197	152	152	
	1 177	30	30	
	1 275	2	2	

Aviso!

?

	Input	Expected	Got	
	2 121	121	121	
	2 180	3	3	
	1 205	87	87	
	1 118	215	215	
	1 239	3	3	
	2 26	1	1	
	2 100	199	199	
	1 165	128	128	
	2 192	3	3	
	2 82	48	48	
	2 252	62	62	
	2 169	2	2	
	2 74	4	4	
	1 133	53	53	
	2 152	2	2	
	1 13	2	2	
	2 236	2	2	
	2 139	3	3	
	1 237	1	1	
	1 204	109	109	
	2 274	73	73	
	2 207	4	4	
	1 20	1	1	
	2 115	224	224	
	2 270	3	3	
	1 76	33	33	
	1 101	17	17	
	2 151	3	3	
	1 205	4	4	
	1 76	5	5	
	1 207	108	108	
	1 24	3	3	
	1 211	4	4	
	2 179	3	3	
	2 8	2	2	
	1 125	2	2	
	1 82	128	128	
	2 30	263	263	
	1 151	275	275	
	1 108	126	126	
	2 47	97	97	

Aviso!

?

	Input	Expected	Got	
	1 164	28	28	
	1 166	2	2	
	2 101	1	1	
	2 244	33	33	
	1 10	104	104	
	1 166	50	50	
	1 210	1	1	
	1 34	162	162	
	1 100	64	64	
	2 207	93	93	
	2 220	1	1	
	2 161	81	81	
	2 68	3	3	
	2 90	1	1	
	2 177	3	3	
	1 133	24	24	
	1 232	1	1	
	2 108	49	49	
	2 180	213	213	
	2 53	1	1	
	1 80	170	170	
	2 121	101	101	
	2 133	1	1	
	2 210	154	154	
	1 129	163	163	
	2 52	6	6	
	2 166	3	3	
	1 217	137	137	
	1 237	2	2	
	2 215	83	83	
	1 29	213	213	
	2 226	1	1	
	2 13	2	2	
	1 264	2	2	
	2 182	2	2	
	2 231	42	42	
	1 9	3	3	
	2 213	4	4	
	2 80	68	68	
	1 164	3	3	
	1 13	1	1	

Aviso!

?

	Input	Expected	Got	
	2 208	1	1	
	1 175	79	79	
	2 73	1	1	
	2 13	2	2	
	1 223	1	1	
	1 213	1	1	
	1 93	1	1	
	1 205	163	163	
	2 125	17	17	
	1 151	93	93	
	1 177	2	2	
	2 96	1	1	
	1 10	2	2	
	1 19	5	5	
	1 95	134	134	
	2 188	2	2	
	1 198	1	1	
	1 234	21	21	
	1 264	1	1	
	1 217	170	170	
	1 145	2	2	
	2 80	101	101	
	2 47	34	34	
	2 210	28	28	
	1 100	1	1	
	1 131	1	1	
	1 93	3	3	
	2 101	1	1	
	2 131	2	2	
	1 133	3	3	
	1 112	2	2	
	2 206	85	85	
	1 36	1	1	
	2 182	9	9	
	1 180	231	231	
	2 231	2	2	
	2 204	232	232	
	2 177	1	1	
	1 169	108	108	
	2 216	2	2	
	1 42	1	1	

Aviso!

?

	Input	Expected	Got	
	1 211	180	180	
	1 84	1	1	
	1 244	71	71	
	1 230	2	2	
	2 196	3	3	
	1 160	1	1	
	2 34	128	128	
	2 109	3	3	
	1 123	1	1	
	2 247	135	135	
	1 178	1	1	
	2 244	2	2	
	1 227	151	151	
	1 139	1	1	
	2 211	67	67	
	1 67	2	2	
	2 205	3	3	
	1 31	4	4	
	1 237	3	3	
	1 157	2	2	
	2 207	1	1	
	1 8	45	45	
	1 130	107	107	
	2 84	1	1	
	1 187	2	2	
	1 239	100	100	
	2 39	4	4	
	1 198	3	3	
	2 24	2	2	
	1 31	1	1	
	1 24	41	41	
	1 206	28	28	
	1 8	111	111	
	1 121	1	1	
	1 78	2	2	
	2 154	2	2	
	2 2	1	1	
	1 95	3	3	
	1 123	3	3	
	1 130	241	241	
	1 125	199	199	

Aviso!

?

	Input	Expected	Got	
	1 237	102	102	
	1 231	2	2	
	2 112	120	120	
	2 142	3	3	
	2 177	1	1	
	2 260	109	109	
	1 73	3	3	
	1 133	1	1	
	1 230	53	53	
	1 127	1	1	
	1 244	4	4	
	1 252	167	167	
	2 95	223	223	
	2 274	230	230	
	2 118	1	1	
	1 233	1	1	
	2 93	3	3	
	1 115	263	263	
	1 67	3	3	
	2 179	2	2	
	1 212	13	13	
	1 142	187	187	
	2 151	29	29	
	1 127	2	2	
	1 101	1	1	
	2 163	108	108	
	2 269	128	128	
	2 176	1	1	
	1 103	4	4	
	1 171	2	2	
	1 237	138	138	
	2 220	2	2	
	1 151	3	3	
	1 48	2	2	
	2 76	104	104	
	2 127	98	98	
	2 27	126	126	
	1 277	14	14	
	1 5	4	4	
	Aviso! 244	1	1	
	2 207	1	1	

	Input	Expected	Got	
	1 33	2	2	
	1 206	2	2	
	1 128	109	109	
	2 33	3	3	
	1 213	29	29	
	1 151	1	1	
	1 154	1	1	
	2 180	1	1	
	2 12	147	147	
	2 68	139	139	
	2 221	1	1	
	1 101	130	130	
	1 142	133	133	
	1 96	2	2	
	1 34	2	2	
	1 230	3	3	
	2 179	58	58	
	1 40	4	4	
	2 27	92	92	
	1 235	1	1	
	1 109	122	122	
	1 257	128	128	
	2 277	185	185	
	2 197	3	3	
	1 193	1	1	
	2 65	1	1	
	2 238	1	1	
	1 175	275	275	
	1 52	98	98	
	1 13	3	3	
	2 168	3	3	
	1 265	3	3	
	2 248	50	50	
	1 165	2	2	
	2 40	108	108	
	2 207	2	2	
	2 19	29	29	
	1 210	3	3	
	1 45	1	1	
	1 55	190	190	
	1 193	120	120	

Aviso!

?

	Input	Expected	Got	
	2 161	25	25	
	2 12	1	1	
	1 250	2	2	
	1 237	28	28	
	1 13	60	60	
	2 53	3	3	
	1 34	1	1	
	2 244	59	59	
	1 133	4	4	
	2 27	108	108	
	1 244	1	1	
	1 263	139	139	
	2 128	1	1	
	2 93	107	107	
	2 239	184	184	
	1 29	73	73	
	1 270	258	258	
	2 177	4	4	
	2 114	179	179	
	2 166	3	3	
	1 264	2	2	
	2 82	236	236	
	1 265	179	179	
	2 244	258	258	
	1 269	29	29	
	2 197	48	48	
	1 176	154	154	
	2 2	1	1	
	2 234	77	77	
	2 8	17	17	
	2 123	134	134	
	1 248	1	1	
	2 227	1	1	
	1 237			
	1 231			
	2 70			
	2 227			
	2 123			
	2 27			
	Aviso! 115			
	2 213			

	Input	Expected	Got	
	1 278 2 266 2 47 2 131 1 232 1 146			
✖	44971 24815 30195 19243 10313 39236 36174 27549 11738 18894 10118 25274 16137 5813 15766 29231 42789 11586 39018 36402 29873 21326 20499 41531 15496 13328 17733 28576 25119 28527 25179 32562 42227 16718 42945 31134 23767 9957 20986 22504 22920 28292 40482 5607 307 25042 6549 948 17452 10210 15299 21291 43867 29680 4574 21138 40125 10286 41702 5115 5495 11118 26710 15902 36755 43613 27261 37784 10545 41826 16998 40698 41555 32689 5787 29585 37891 19476 35082 7227 1789 24210 1810 44282 965 13920 116 13345 4691 34843 43412 35034 19997 16254 792 10919 15047 7808 18281 10112 28301 43158 29905 43766 37645 30372 28128 3720 26584 38540 26232 646 23861 19079 5545 29283 36353 32197 5948 34724 12993 3363 31426 17112 13962 40034 23973 6898 28331 28398 16962 34386 17117 33107 5970 17420 12328 28483 2254 1009 53 27057 20411 25685 41454 14826 21511 16877 31826 33150 38367 11498 44246 39660 15064 14609 3493 42651 43789 43158 21643 39877 39481 14204 39058 8406 42842 26358 8034 14384 24622 8945 11389 36237 25909 38577 28018 23131 18384 13699 42376 16918 8226 35132 35200 33853 1301 14347 2724 29482 32940 12104 4198 12779 32322 13999 13733 28789 3756 7284 25315 41307 38317 36575 38038 4418 6538 5527 21627 31136 35748 28107 22637 9881 8819 43048 22382 44003 8669 4783 37599 20466 19684 2249 16710 44123 20667 39931 8142 7288 31566 24192 5298 29846 23862 43251 8304 18521 19273 15847 12927 32176 31419 24412 6816 2386 1816 27105 9007 40442 22260 5831 4348 14174 15975 3147 7838 2949 7638 27743 5374 23786 2515 21014 9992 33785 9845 38354 3727 2679 41730 30598 22486 42987 30043 14417 12968 17276 29280 14609 43138 285 28440 814 38958 42744 33061 29759 6309 19874 36945 34736 14923 14587 16773 5170 24610 32600 24021 16755 32382 32674 14725 23315 16774 20673 32579 14971 31889 39449 9870 29013 16635 40998 35866 24782 27176 9631 17004 23028 20952 19614 19660 6930 14987 5681 34807 34035 30936 5381 42203 16229 37436 6840 19318 19566 19494 18460 29846 30636 41601 13079 25110 22859 38851 65 42091 12943 23941 27517 15521 30827 32520 17318 42152 43043 7804 43594 6150 24318 27921 30999 35434 2151 19541 40452 41584 37050 37293 33642 28961 23441 31586 23934 11601 27091 27791 14583 8695 25246 16332 11811 43472 6920 2305 34252 34510 33267 19769 19136 19313 26445 11020 15232 7106 13695 42348 20463 27673 33556 11383 19529 14152 38770 3729 22262 4234 19138 20668 21795 7464 30599 18354 11734 12573 41451 19106 16913 28359 22223 37384 38513 1446 18381 35031 28807 31321 18155 25150 30939 10253 33054 42011 30472 44626 22505 34276 23021 41398 6608 36965 21567 13284 133 32497 34755 20323 25392 37172 12952 42109 11387 36246 16588 9764 33552 16761 30246 34280 36614 23552 30261 6875 32808 42442 39306 1732 10625 17500 28556 39034 41243 12267 2409 32180 25373 1569 40661 19409 29760 25634 822 829 34682 17136 14703 20271 4422 35640 28343 28697 11333 17914 1261 32341 17271 36117 55 4100 31047 39502 24227 14265 2644 13594 5650 3826 41586 44 5367 8242 18796 15732 7793	17428 1 1 1 6017 12908 2 2 2 2 11844 2 2 25601 1 1 3590 13914 2 1403 18894 2 1 1654 848 1656 3 2 9889 6138 5461 3 9020 1104 27685	17428 1 1 1 6017 12908 2 2 2 2 11844 2 2 25601 1 1 3590 13914 2 1403 18894 2 1 1654 848 1656 3 2 9889 6138 5461 3 9020 1104 27685	✖

Aviso!

?

Input		Got	
30532 3321 2896 34675 19887 13085 806 31141 15899 32443 25001 3440 6179 11780 16631 4090	15359	15359	
9219 14042 20308 31545 35028 16944 17984 33576 32139 37782 28813 31888 25369 44032 9682	1	1	
24880 14562 6696 22545 17610 2943 44282 30199 18067 10368 4322 612 27340 13727 39166 38885	38763	38763	
27797 12629 8370 6432 39070 15958 10814 24108 25035 728 14556 24103 38449 25658 25266 16229	32331	32331	
22911 44410 3834 8251 22919 22361 32539 42712 40205 38284 25513 8358 15507 26619 36364	1480	1480	
11694 40280 11578 7209 1758 23679 30701 29421 38698 12713 1209 30621 8314 30340 41939 41256	2705	2705	
12880 35873 17949 29535 35452 32440 21188 35382 28696 22447 12502 30138 23771 37797 11017	27510	27510	
20113 27505 12766 14538 39504 24561 12892 19863 25549 6915 16945 11199 25861 30740 6004	37339	37339	
4838 12286 11847 21800 10827 20734 16398 35565 41197 37130 16138 32477 34159 9176 39385	44073	44073	
14677 8436 33219 1121 17308 20635 26844 9619 42469 35603 27915 32798 21094 43924 36127	1	1	
36951 4366 4223 15306 19706 29164 5071 31088 6570 7694 7951 31708 37090 18264 34613 25552	1	1	
1290 39800 2995 37175 17343 42069 15125 17731 28487 ...snip... 2	2	2	
2 18787	1	1	
1 21923	16441	16441	
2 10303	37300	37300	
1 41145	7633	7633	
2 997	2	2	
2 21802	1	1	
1 3080	36354	36354	
1 19464	24927	24927	
2 10922	4	4	
2 6816	1	1	
1 205	1	1	
2 27951	20582	20582	
1 44113	2	2	
1 36948	2	2	
1 8596	22869	22869	
1 25919	1	1	
1 36838	4970	4970	
2 30962	20270	20270	
1 41502	2	2	
1 31933	4	4	
1 883	41419	41419	
2 2198	6	6	
2 21209	2	2	
1 20489	31565	31565	
1 16521	2	2	
1 2434	9698	9698	
2 586	1824	1824	
30645	1	1	
879	13765	13765	

Aviso!

?

Input		Got	
2 29937	7754	7754	
2 38346	3	3	
1 29438	13914	13914	
1 19312	2	2	
1 43285	2	2	
2 41161	18222	18222	
1 31661	3	3	
1 16401	11443	11443	
2 31509	1	1	
1 30016	2	2	
1 41731	4	4	
1 21752	8626	8626	
2 25733	3	3	
2 5830	12972	12972	
1 24990	40323	40323	
1 10346	1	1	
1 32313	1	1	
2 25150	3812	3812	
1 29830	11037	11037	
2 17812	7142	7142	
1 41850	1	1	
2 25647	1	1	
2 32733	859	859	
1 35831	18100	18100	
1 16522	2	2	
2 23885	1	1	
2 25000	1	1	
1 26582	26583	26583	
2 31097	5661	5661	
1 6896	2	2	
1 977	1	1	
2 20221	25011	25011	
1 18477	37102	37102	
2 41229	30845	30845	
1 9121	2	2	
1 17434	1	1	
1 35958	28246	28246	
1 16874	2	2	
1 31751	2511	2511	
2 42252	2	2	
1 3569	3	3	

Aviso!

?

	Input		Got	
	2 10265	2	2	
	2 5272	2352	2352	
	2 25332	34977	34977	
	1 38762	1	1	
	2 36600	2003	2003	
	1 8741	390	390	
	2 42575	3	3	
	2 8163	1	1	
	2 4234	4848	4848	
	1 25048	2	2	
	1 9091	8672	8672	
	2 26964	2	2	
	1 2460	34317	34317	
	1 29627	30425	30425	
	1 28930	1	1	
	2 28268	12388	12388	
	1 23238	6691	6691	
	2 18787	25081	25081	
	1 36112	1	1	
	1 21045	3	3	
	1 7872	2	2	
	1 42676	1889	1889	
	1 7650	1	1	
	2 22528	13309	13309	
	1 38910	30679	30679	
	1 20686	23772	23772	
	2 15732	2	2	
	2 31654	7510	7510	
	1 28941	11213	11213	
	2 29255	19298	19298	
	2 2339	2	2	
	2 42856	2	2	
	1 35181	2	2	
	2 7004	2	2	
	2 21036	10164	10164	
	1 16024	26228	26228	
	1 4348	1	1	
	2 39611	2	2	
	1 28332	5	5	
	2 38077	2	2	
	2 35904	6627	6627	

Aviso!

?

	Input		Got	
	1 26305	10388	10388	
	1 44257	3	3	
	2 43433	3	3	
	1 44625	742	742	
	2 4557	8585	8585	
	1 26045	26210	26210	
	1 28187	2980	2980	
	2 28057	6	6	
	1 39855	1	1	
	2 22115	14678	14678	
	2 24911	35844	35844	
	2 24393	34583	34583	
	2 4630	2	2	
	2 4938	1	1	
	2 14789	1	1	
	2 2077	3	3	
	1 37422	3	3	
	1 13227	2	2	
	1 15931	4	4	
	2 39452	18831	18831	
	2 38032	1	1	
	2 16473	17153	17153	
	2 18614	42986	42986	
	2 31208	2	2	
	1 4850	25102	25102	
	1 35073	2	2	
	1 32170	1	1	
	2 7907	12650	12650	
	1 35045	17300	17300	
	1 28121	8262	8262	
	1 30992	1	1	
	2 3869	3	3	
	2 6709	2	2	
	2 39891	34407	34407	
	1 16316	16680	16680	
	1 14752	1	1	
	2 21825	16643	16643	
	2 4762	1875	1875	
	2 39828	2	2	
	1 38223	1	1	
	2 35251	2	2	

Aviso!

?

Input		Got	
1 31826	3	3	
1 34097	21486	21486	
2 44886	2	2	
2 21160	2292	2292	
1 3714	11720	11720	
1 44743	1	1	
2 31470	20625	20625	
2 17002	2	2	
2 22147	2	2	
2 21429	1	1	
2 32292	6120	6120	
1 44949	3	3	
2 22295	10288	10288	
2 9103	614	614	
1 25265	21569	21569	
1 31917	3	3	
2 6093	15854	15854	
1 1355	2	2	
1 10775	13134	13134	
2 15114	1249	1249	
1 26130	3	3	
2 11003	2	2	
2 21209	1	1	
1 5676	2	2	
2 7157	2	2	
2 26725	11986	11986	
1 15884	2	2	
2 14974	2	2	
2 10367	33677	33677	
1 17613	2324	2324	
2 8323	4619	4619	
1 15896	4	4	
2 21042	2	2	
2 41824	1	1	
1 14743	9103	9103	
1 39406	2	2	
2 28624	19870	19870	
2 20329	750	750	
2 25749	11778	11778	
2 26742	13465	13465	
2 15462	1	1	

Aviso!

?

	Input		Got	
	2 29019	15525	15525	
	2 6452	9003	9003	
	1 9272	19465	19465	
	1 19270	1	1	
	2 32550	40010	40010	
	1 29773	1	1	
	1 35149	1	1	
	1 29553	7416	7416	
	1 43495	39656	39656	
	1 35043	28241	28241	
	1 15540	2	2	
	1 5961	5702	5702	
	2 6509	6449	6449	
	2 4250	3391	3391	
	1 33454	22730	22730	
	1 27830	12039	12039	
	1 37240	1	1	
	2 3775	12455	12455	
	2 1681	2	2	
	1 39203	4407	4407	
	2 40731	4	4	
	1 27777	3	3	
	2 41532	10759	10759	
	1 39738	10851	10851	
	1 35314	2	2	
	2 38238	12523	12523	
	1 30927	8945	8945	
	2 42876	3	3	
	2 34291	25730	25730	
	2 1080	8519	8519	
	1 28558	5	5	
	1 21113	2	2	
	2 6280	2	2	
	2 7765	3	3	
	1 25825	1	1	
	2 41321	6390	6390	
	1 26088	6	6	
	1 35365	14383	14383	
	2 31682	11726	11726	
	1 15193	11753	11753	
	1 30194	13996	13996	

Aviso!

?

	Input		Got	
	2 6650	26881	26881	
	2 6714	20727	20727	
	2 44765	562	562	
	2 13882	10749	10749	
	1 42303	3	3	
	1 39819	2	2	
	2 36502	1	1	
	1 43142	41586	41586	
	1 20123	3	3	
	1 6383	24501	24501	
	1 13080	9784	9784	
	2 9162	8166	8166	
	2 20952	25836	25836	
	1 37848	5218	5218	
	1 31287	289	289	
	1 40808	4706	4706	
	1 40715	17242	17242	
	1 23423	2	2	
	1 18736	2425	2425	
	1 32962	37339	37339	
	1 31010	1	1	
	1 23950	2	2	
	2 42547	1691	1691	
	1 22848	2	2	
	2 38626	42113	42113	
	1 40995	2	2	
	1 11703	2	2	
	1 14298	1	1	
	2 21670	2	2	
	1 38178	1573	1573	
	1 22705	28796	28796	
	1 3669	1	1	
	1 39529	5	5	
	1 25762	2	2	
	1 21661	14466	14466	
	2 16023	3	3	
	2 44406	2	2	
	2 15462	18228	18228	
	2 33087	4	4	
	1 1825	1	1	
	2 23869	4	4	

Aviso!

?

Input		Got	
1 21214	1	1	
2 42088	7438	7438	
1 23991	1	1	
1 17568	30702	30702	
2 3784	2	2	
1 15284	1	1	
2 30458	5	5	
1 23246	1	1	
1 6321	8867	8867	
2 30840	8262	8262	
1 38664	3	3	
2 2361	4	4	
1 15670	1	1	
1 9143	3	3	
2 1167	1	1	
1 38182	2	2	
2 19133	5	5	
2 14749	31765	31765	
2 24419	2	2	
1 43496	27680	27680	
1 26421	4	4	
2 17324	4613	4613	
1 4023	3	3	
1 31692	13309	13309	
2 40801	3	3	
1 36744	1	1	
1 6910	1	1	
1 33578	2609	2609	
1 834	1	1	
1 16530	4	4	
2 12533	878	878	
2 2721	1	1	
2 43321	12710	12710	
1 8114	2	2	
1 14912	31226	31226	
1 24895	7590	7590	
2 3184	7532	7532	
2 19897	2200	2200	
2 44584	1	1	
2 3955	34488	34488	
2 299	35824	35824	

Aviso!

?

Input		Got	
2 26646	2	2	
1 5976	14588	14588	
2 41944	26548	26548	
2 7981	35750	35750	
2 35633	3	3	
2 22816	1	1	
2 14001	2473	2473	
2 44516	4	4	
2 9666	19543	19543	
2 37145	2463	2463	
1 7003	2	2	
2 23050	1	1	
2 27912	1	1	
2 11051	2	2	
2 25720	19277	19277	
2 28961	3	3	
1 20547	25294	25294	
1 6466	2	2	
1 21043	229	229	
1 5001	8197	8197	
2 35439	10406	10406	
2 36495	30666	30666	
1 16927	3626	3626	
1 9414	36169	36169	
1 14754	1	1	
2 9866	1	1	
2 3163	18761	18761	
2 31101	26989	26989	
2 11878	32723	32723	
2 15181	4683	4683	
1 2120	8608	8608	
2 16499	4415	4415	
1 247	3	3	
2 7601	37514	37514	
1 37014	1	1	
2 27501	12149	12149	
2 38049	2	2	
2 14034	8090	8090	
2 19309	43134	43134	
2 7479	13472	13472	
2 25951	11869	11869	

Aviso!

?

Input		Got	
2 17319	1	1	
1 14178	9520	9520	
1 12383	38478	38478	
1 20287	2	2	
1 3263	31470	31470	
1 4670	9726	9726	
2 36089	13117	13117	
2 3124	22287	22287	
1 16460	2	2	
1 42064	20275	20275	
1 5605	3	3	
1 1632	2072	2072	
1 38344	1479	1479	
1 24105	3	3	
1 23342	3	3	
2 39665	1	1	
1 20365	32095	32095	
2 5300	2	2	
1 20135	12998	12998	
2 21634	8583	8583	
2 18548	2	2	
2 44433	26766	26766	
1 13319	2	2	
2 5637	3	3	
1 31199	2	2	
1 1445	2	2	
1 34294	3	3	
1 35800	38583	38583	
1 42108	24428	24428	
1 7545	33389	33389	
1 14724	1	1	
2 43985	2	2	
1 23558	2	2	
1 38237	8061	8061	
1 22864	31371	31371	
1 21428	1	1	
1 8379	24417	24417	
1 38638	4	4	
1 35873	7865	7865	
1 37759	1	1	
1 40041	3	3	

Aviso!

?

	Input		Got	
	1 40998	3181	3181	
	2 868	2	2	
	2 37470	3	3	
	2 40728	4230	4230	
	2 29301	1	1	
	1 17661	26579	26579	
	2 35326	1	1	
	1 40369	1	1	
	2 29408	36071	36071	
	1 10430	24505	24505	
	1 44766	5	5	
	1 35151	15615	15615	
	2 8181	3	3	
	1 4550	1	1	
	2 35499	10150	10150	
	2 35094	4781	4781	
	2 43966	28557	28557	
	1 12297	16856	16856	
	1 4688	1146	1146	
	2 17803	2401	2401	
	1 4516	1	1	
	1 36948	39166	39166	
	2 10172	1	1	
	2 43570	1	1	
	1 11598	1	1	
	1 9996	4	4	
	1 24420	15063	15063	
	1 25045	1	1	
	2 1977	37778	37778	
	2 43655	1	1	
	2 40083	3	3	
	1 42868	39841	39841	
	2 12713	5780	5780	
	2 44898	3	3	
	2 25950	1	1	
	1 16895	3	3	
	2 17090	2	2	
	1 33510	27948	27948	
	1 2195	19083	19083	
	1 11121	2	2	
	1 8861	2	2	

Aviso!

?

	Input		Got	
	1 34059	26374	26374	
	2 7004	3	3	
	1 11013	28283	28283	
	1 38118	5428	5428	
	2 20769	3	3	
	1 44766	1	1	
	1 7900	1	1	
	2 13361	1	1	
	1 44697	6355	6355	
	2 7145	22760	22760	
	2 14987	3	3	
	2 10805	3	3	
	1 18727	16191	16191	
	1 23214	3	3	
	1 9689	7633	7633	
	1 23806	2	2	
	1 30338	18327	18327	
	1 26454	2	2	
	1 7121	29631	29631	
	1 25991	2	2	
	1 3732	2	2	
	1 7032	848	848	
	2 19014	1390	1390	
	1 37486	11452	11452	
	2 8073	1	1	
	1 8857	3	3	
	2 11789	16927	16927	
	2 28232	2	2	
	2 26257	2	2	
	1 21222	3	3	
	1 37861	10125	10125	
	2 7064	3903	3903	
	2 24016	4033	4033	
	2 40350	1239	1239	
	2 14053	31703	31703	
	1 13383	4183	4183	
	1 3776	2	2	
	2 2510	25489	25489	
	1 43780	3	3	
	1 44669	1	1	
	1 8571	2	2	

Aviso!

?

Input		Got	
1 17115	1	1	
2 37576	4120	4120	
2 31367	1	1	
1 13657	400	400	
1 2358	3	3	
1 13088	18858	18858	
1 15791	4029	4029	
1 39459	2	2	
1 33228	2	2	
2 11752	11466	11466	
2 44948	20464	20464	
1 10090	2	2	
2 728	1	1	
1 17170	1602	1602	
1 21352	2	2	
1 12078	1	1	
2 38925	16098	16098	
2 31405	29493	29493	
1 19303	20396	20396	
2 19207	24359	24359	
2 89	13893	13893	
2 22120	30487	30487	
1 35367	1	1	
2 29343	7199	7199	
1 5681	6668	6668	
2 14739	20248	20248	
1 40233	1	1	
1 13533	2	2	
2 42084	1	1	
2 5578	11063	11063	
1 40483	20644	20644	
1 2068	2	2	
2 42230	3	3	
1 40526	14810	14810	
1 9411	12804	12804	
1 16553	2	2	
	2	2	
	5	5	
	2	2	
	2	2	
	7111	7111	

Aviso!

?

Input		Got	
		26046	26046
		3	3
		1976	1976
		1	1
		1	1
		24984	24984
		2	2
		44213	44213
		1	1
		25362	25362
		1	1
		2335	2335
		16009	16009
		4428	4428
		14038	14038
		1	1
		6207	6207
		1	1
		849	849
		2	2
		2	2
		9411	9411
		1	1
		4	4
		33960	33960
		4	4
		4992	4992
		1	1
		9443	9443
		1	1
		1	1
		3	3
		28338	28338
		3	3
		806	806
		1	1
		35241	35241
		14543	14543
		5420	5420
		6956	6956
		10539	10539

Aviso!

?

Input		Got	
		8435	8435
		1	1
		1	1
		3872	3872
		2	2
		2	2
		8339	8339
		3	3
		2	2
		14436	14436
		2	2
		4069	4069
		4	4
		13153	13153
		2	2
		1	1
		1	1
		17758	17758
		27394	27394
		3	3
		2	2
		3	3
		26413	26413
		24273	24273
		3	3
		4004	4004
		25168	25168
		4	4
		2	2
		2	2
		2	2
		19664	19664
		4	4
		29750	29750
		34475	34475
		3099	3099
		1193	1193
		3	3
		11078	11078
		1	1
		2	2

Aviso!

?

Input		Got
	1	1
	16627	16627
	3	3
	2	2
	5113	5113
	3	3
	7823	7823
	4	4
	8090	8090
	1131	1131
	1	1
	1	1
	3	3
	1	1
	2	2
	2	2
	2	2
	15766	15766
	41257	41257
	24876	24876
	2	2
	1	1
	9721	9721
	1	1
	3	3
	11369	11369
	1	1
	2	2
	2	2
	1	1
	2	2
	20014	20014
	1	1
	20226	20226
	1	1
	18784	18784
	1	1
	1	1
	3	3
	2	2
	4699	4699

Aviso!

?

Input		Got	
		42066	42066
		4	4
		32335	32335
		1	1
		3	3
		1	1
		2320	2320
		1	1
		2	2
		7907	7907
		1	1
		2	2
		10064	10064
		3	3
		1	1
		1	1
		3	3
		1	1
		20144	20144
		17287	17287
		3	3
		2	2
		6596	6596
		2	2
		4	4
		1	1
		2	2
		2	2
		1	1
		4	4
		4	4
		30591	30591
		2	2
		2	2
		2	2
		7172	7172
		1	1
		25873	25873
		12543	12543
		6737	6737
		14857	14857

Aviso!

?

Input		Got	
		1	1
		1	1
		8991	8991
		42362	42362
		1829	1829
		35195	35195
		7976	7976
		23743	23743
		1	1
		1	1
		3	3
		5	5
		7110	7110
		2232	2232
		2	2
		1	1
		1168	1168
		2	2
		2	2
		7404	7404
		3623	3623
		16439	16439
		10986	10986
		1	1
		4587	4587
		5227	5227
		1	1
		1	1
		1	1
		2	2
		1	1
		1	1
		1	1
		2	2
		1	1
		4712	4712
		1	1
		1	1
		39137	39137
		1	1
		9778	9778

Aviso!

?

Input		Got	
		28128	28128
		1	1
		5976	5976
		35067	35067
		19746	19746
		9630	9630
		3046	3046
		2	2
		3	3
		1	1
		9435	9435
		2	2
		16220	16220
		1	1
		4359	4359
		1	1
		1	1
		2	2
		3	3
		1	1
		28584	28584
		1	1
		2	2
		2	2
		614	614
		3	3
		1	1
		1	1
		4	4
		24039	24039
		28545	28545
		13621	13621
		3	3
		2	2
		3	3
		5492	5492
		3	3
		2	2
		3	3
		1	1
		3	3

Aviso!

?

Input		Got	
		2976	2976
		11860	11860
		1	1
		26524	26524
		17254	17254
		3	3
		9124	9124
		16439	16439
		3	3
		1	1
		26944	26944
		40718	40718
		3	3
		6144	6144
		20876	20876
		2519	2519
		26135	26135
		2	2
		9138	9138
		1955	1955
		3	3
		19625	19625
		4783	4783
		15236	15236
		3	3
		3228	3228
		422	422
		4	4
		1	1
		8702	8702
		1	1
		36419	36419
		1	1
		11951	11951
		3338	3338
		43034	43034
		2	2
		26471	26471
		5	5
		9791	9791
		4	4

Aviso!

?

Input		Got	
		12934	12934
		12129	12129
		9430	9430
		24054	24054
		1	1
		2	2
		27404	27404
		2	2
		3	3
		2	2
		11126	11126
		396	396
		13618	13618
		4738	4738
		24749	24749
		1	1
		1210	1210
		3553	3553
		32159	32159
		11937	11937
		1	1
		11138	11138
		26986	26986
		1	1
		8307	8307
		2705	2705
		20204	20204
		3	3
		1	1
		11657	11657
		15571	15571
		3	3
		16345	16345
		3	3
		2	2
		2167	2167
		4989	4989
		3	3
		7551	7551
		2	2
		2	2

Aviso!

?

Input		Got	
		4	4
		41253	41253
		7934	7934
		4952	4952
		1	1
		4	4
		17526	17526
		13609	13609
		3	3
		2	2
		34436	34436
		3	3
		8659	8659
		20661	20661
		18383	18383
		1	1
		1910	1910
		1	1
		14994	14994
		1	1
		6193	6193
		3	3
		20216	20216
		1	1
		1	1
		1228	1228
		6791	6791
		41497	41497
		1	1
		1	1
		1	1
		4	4
		16237	16237
		3	3
		2	2
		2	2
		1	1
		1	1
		9638	9638
		2728	2728
		7424	7424

Aviso!

?

Input		Got
	1	1
	16029	16029
	20463	20463
	37869	37869
	18305	18305
	23659	23659
	6181	6181
	2	2
	3	3
	1	1
	1	1
	4	4
	36787	36787
	4	4
	1	1
	1	1
	3	3
	3012	3012
	2	2
	1	1
	3	3
	1	1
	20727	20727
	10548	10548
	2	2
	3	3
	1	1
	3458	3458
	24610	24610
	1	1
	2	2
	1	1
	2	2
	2	2
	8588	8588
	1	1
	39818	39818
	6204	6204
	1	1
	3	3
	39758	39758

Aviso!

?

Input		Got	
		4020	4020
		1	1
		5479	5479
		17228	17228
		1	1
		13984	13984
		1	1
		2	2
		27663	27663
		8049	8049
		2115	2115
		1	1
		3	3
		6652	6652
		42528	42528
		6881	6881
		2	2
		300	300
		5228	5228
		11631	11631
		215	215
		3	3
		2	2
		4	4
		18190	18190
		1	1
		1	1
		1	1
		1	1
		7591	7591
		12838	12838
		11211	11211
		1	1
		2	2
		3977	3977
		1	1
		2031	2031
		6711	6711
		1	1
		11111	11111
		2	2

Aviso!

?

Input		Got	
		4	4
		2	2
		3	3
		15804	15804
		2	2
		13930	13930
		900	900
		2	2
		4200	4200
		1	1
		627	627
		15473	15473
		1	1
		2	2
		1	1
		1584	1584
		1	1
		4912	4912
		1	1
		543	543
		32223	32223
		5	5
		1	1
		4772	4772
		2	2
		1	1
		4	4
		1	1
		8893	8893
		26601	26601
		23371	23371
		17246	17246
		2	2
		3044	3044
		2	2
		17130	17130
		32921	32921
		2	2
		44175	44175
		9881	9881
		22470	22470

Aviso!

?

Input		Got
	294	294
	2569	2569
	3391	3391
	10022	10022
	2	2
	35783	35783
	1	1
	10119	10119
	25250	25250
	...snip...	...snip...
	13120	93
	2	4
	2	1435
	13937	2
	2	1
	30566	2
	1	3
	10283	1
	33998	38092
	3	15651
	9906	3
	1	34481
	1812	555
	1	23623
	30608	4887
	3	2
	13064	4423
	37437	29873
	4	6816
	5920	19167
	7306	4
	1715	1
	2	23522
	2	1
	2	2
	18019	2769
	8702	1
	1	16212
	18518	7980
	1	3
	2	2

Aviso!

?

Input		Got
	4	27596
	2	2
	3	1971
	2	4
	11401	3
	21723	1
	4	1
	1	1
	2	3
	6327	20272
	4841	2
	40118	4854
	2	1
	11417	29102
	4	42983
	1594	8384
	2	1
	1	2
	1	3
	1	18808
	41644	7974
	33672	35447
	2	3
	1	11848
	8980	29766
	11143	1
	3	3
	2	1
	2	3
	1239	1
	22342	16390
	30026	1
	3	44058
	4	1115
	2	2
	5606	2
	4807	2
	20213	3
	15691	3
	2	15292
	1	2

Aviso!

?

Input		Got
	1	3
	1	39175
	2	14871
	1	4
	4	2
	20283	38682
	9528	3469
	10292	3041
	29326	36601
	3	4
	1	1
	40915	2
	11569	1333
	2	1
	1703	30404
	22788	2
	11431	39058
	1	6891
	3	1
	7971	28839
	3	4
	2	3
	30007	3
	2	2
	1	2
	4	9003
	3	26231
	5387	2
	32371	2
	5109	1
	2	3168
	1	34858
	4734	4
	1	1
	11242	33412
	30495	3
	20460	1
	2	7245
	32711	3
	35813	2
	2	40115

Aviso!

?

Input		Got
	1	15963
	22075	17758
	8752	3053
	1723	4
	3	1
	1	3071
	13134	6339
	2	8249
	1	2
	1	1630
	16589	9225
	33619	1484
	2	17964
	2	1267
	5	1
	6487	2
	1	4
	2	24238
	3495	1
	1	19290
	4	1
	21888	9966
	4	3
	5	1075
	1585	2
	2	1057
	1	5139
	1	1
	13221	2
	557	1
	3	22530
	3	1
	1	9446
	2	2
	14520	4
	3	1
	1	13471
	4	714
	12211	11625
	2454	2
	23474	955

Aviso!

?

Input		Got	
	2	1	
	10559	6148	
	7727	11890	
	1	3	
	4	1	
	9656	1	
	4	2	
	2	41365	
	1	17820	
	2	2	
	40510	262	
	1	10886	
	7373	7895	
	1	5	
	19878	2	
	12774	1	
	3867	34321	
	2	2	
	4	41849	
	10424	2	
	9904	6449	
	3	2	
	1	1	
	3	26360	
	41082	23524	
	4	3	
	2	2	
	30910	35693	
	2063	22288	
	8645	1394	
	26990	20778	
	1	30172	
	9949	1021	
	16760	4	
	1	3	
	28443	486	
	1	2	
	3	1	
	22494	2	
	18618	2	
	22014	3215	

Aviso!

?

Input		Got	
	6622	1	
	15275	2	
	3	9097	
	3	2	
	1	2	
	20183	77	
	10215	28158	
	19885	21611	
	1	29401	
	2565	1	
	11411	1	
	1	2	
	4	3	
	2	1	
	1	1	
	166	3	
	12611	2	
	3	523	
	2	24775	
	3	13343	
	40674	1	
	11120	404	
	2	14854	
	2	3787	
	1268	1182	
	31978	1	
	10426	40381	
	38387	37219	
	9907	5	
	13461	3	
	2	15902	
	2	19662	
	9196	1	
	4	114	
	2196	2	
	14006	1	
	1	2	
	2	4	
	7227	26232	
	2	2	
	3	1	

Aviso!

?

Input		Got	
		14789	33038
		11200	1
		17938	9343
		2	5980
		2	3
		4700	2
		1	32234
		2	2
		15836	4
		3660	652
		21167	11717
		1	2
		1	3
		22026	2
		35596	1
		4	1
		3	4914
		2	12890
		3	1
		6254	3
		2	1
		1	39869
		1	1
		1	10336
		2	11779
		23495	2508
		3	15912
		1	10528
		2570	2
		1	3
		385	2
		2	1
		1	44617
		2463	2
		1395	28078
		2	21451
		3856	2559
		3	24864
		12408	4891
		13265	2
		2	14576

Aviso!

?

Input		Got	
	12083	2	
	1	2	
	2	2	
	25836	1	
	17547	23745	
	36117	1	
	2	1	
	3585	2	
	3	2	
	15838	2	
	4	15441	
	1	2	
	1	2	
	6411	2	
	309	31824	
	1	3	
	1	3	
	7817	4177	
	23753	2	
	1	17906	
	1	3	
	2	3	
	25676	1	
	2	16106	
	17408	11437	
	2	2	
	3747	28292	
	1	1	
	32030	21541	
	2	1	
	2	1	
	212	2812	
	19220	1	
	1	12565	
	2	18964	
	17798	8529	
	2	1	
	27844	24175	
	32282	2738	
	3	39623	
	2	3	

Aviso!

?

Input		Got	
		9660	25440
		1869	30666
		40753	17857
		1966	16852
		3	3
		2	31559
		8518	2101
		43450	5535
		2	2953
		6635	2
		2	1
		5850	1
		2645	2
		1	4
		1	4
		9156	1
		4862	7070
		43053	2
		2	4
		1	2
		10390	3
		18794	2
		2	4609
		12146	1
		1	2
		14087	34888
		7847	2
		2	3
		2	1
		1	4
		26570	1
		3524	2022
		2	18777
		19164	3
		4089	2
		4	5678
		13416	8678
		1	5254
		4742	1
		10838	1
		11760	1

Aviso!

?

Input		Got	
	11214	4	
	3	23175	
	9145	1	
	8518	3	
	3	3	
	9874	370	
	27141	1	
	2603	18737	
	15391	1	
	1	24308	
	20249	2578	
	5787	11512	
	2	34867	
	2	12928	
	3	2	
	3	15884	
	1	3	
	12013	18424	
	4908	29023	
	19786	1309	
	18019	23528	
	33255	3	
	1184	3	
	5282	3	
	2	1	
	10900	2	
	7348	7520	
	4767	1	
	238	6336	
	24412	11864	
	7270	1	
	2	2	
	1	1	
	36823	2136	
	3	11887	
	2	23885	
	11115	8414	
	2578	1	
	3	6926	
	4	1	
	21896	3	

Aviso!

?

Input		Got	
	1	1	
	2	2	
	2	3	
	35000	3	
	18567	2	
	18737	6636	
	3853	15889	
	6918	1244	
	6015	2	
	1	16701	
	3	11369	
	5	1	
	3	3	
	3	1	
	26165	2	
	2	12056	
	14343	18955	
	1	2	
	1859	1	
	1	13397	
	3	1	
	1906	2	
	349	2	
	2	37827	
	3	10007	
	1	9105	
	3	3	
	5582	3	
	8898	12808	
	35967	42526	
	18343	1	
	2	30765	
	264	2	
	3	11261	
	2	2	
	4	1	
	16038	2	
	2	32377	
	4	6760	
	2	19428	
	2	2	

Aviso!

?

Input		Got
	1	2
	12402	25131
	2	2
	9551	2
	1	14216
	2	15453
	2	2
	14270	1168
	1	1389
	3	1
	1	2
	15558	1
	3	3475
	7189	5972
	21611	3
	35191	11323
	3	14976
	7889	1
	1098	2
	4	3
	11863	28413
	32822	2504
	1	23296
	2	12556
	24609	1
	1	3
	3	2
	449	2
	1	29711
	1	3221
	1	1804
	39363	1
	7680	1
	2	20859
	1	3
	23040	2
	1112	4
	2	1
	1	23308
	1	36331
	1	17664

Aviso!

?

Input		Got	
	6509	2	
	17558	2	
	3	2482	
	36551	30145	
	7	827	
	2	3	
	7459	5388	
	16629	8702	
	1	31511	
	19527	4	
	17523	25079	
	11356	3	
	4	4	
	36500	5496	
	2	2	
	15613	33644	
	3	1	
	4	14191	
	10585	4	
	1	11399	
	6389	1	
	4	3	
	33647	2719	
	34949	3	
	41709	10593	
	3	2	
	3	13663	
	1	19417	
	11390	13422	
	2	1	
	17181	1	
	14172	8166	
	33869	1	
	36510	5	
	1	28562	
	34797	2	
	2	1	
	1220	1	
	3600	27817	
	1	13759	
	3656	4	

Aviso!

?

Input		Got	
		28133	8942
		1	2030
		1	2
		3	2685
		1	29454
		6727	28903
		12519	4
		2425	12697
		7090	1
		17454	1
		1	13781
		3	3
		3	8504
		1	1390
		2	2
		2	1
		2	2
		29873	2
		2	7139
		8866	1
		1	2
		1	2
		4	1
		2047	2
		41903	36494
		2	23885
		33594	2
		39142	5502
		13938	1
		43037	11889
		36342	470
		1	3
		2	5523
		234	3
		3474	1
		671	15727
		4	33411
		9884	1
		2	1
		5	11394
		39431	3

Aviso!

?

Input		Got	
	21819	1	
	1	2	
	43736	5198	
	1	4	
	6408	3	
	32402	3	
	3	1	
	1	10429	
	35328	42729	
	243	3470	
	1	2	
	28997	1	
	1	43693	
	1	3	
	2	1	
	4	1	
	2	2	
	11518	16810	
	3	15996	
	1	3	
	1	39215	
	28607	1	
	24947	44560	
	3	1	
	1	4	
	3	29865	
	12793	26910	
	20982	130	
	2	6	
	917	1	
	8205	22525	
	2	2	
	2	25938	
	1	4	
	7386	30853	
	2	25217	
	3	7068	
	5420	2	
	1	1	
	3	2	
	2	37165	

Aviso!

?

Input		Got	
		27786	2
		4657	14775
		1	30847
		1	1
		1	15576
		427	1
		1	1163
		719	4
		2	9624
		9598	35460
		19059	3
		3	3
		1	2
		12586	2
		29407	3
		4	15748
		27315	6
		1	4
		4	10868
		22892	3
		1	22582
		26523	21758
		4	40296
		3	1
		2	12252
		1	11784
		2	2
		14878	2807
		1	8334
		14284	2
		32568	1
		6268	3
		1	1
		21301	2
		3	1
		4032	6494
		7279	1
		405	1
		1	2
		2	27759
		4025	1

Aviso!

?

Input		Got
	2	28873
	1	5945
	4	1
	16021	3
	3	6379
	21819	2040
	1	1
	3	1
	1	10940
	2	2
	1	10876
	12204	22557
	2	2
	2	2
	511	2
	5158	2
	4	1
	27472	11453
	11171	1
	37288	4
	2	27706
	17816	7482
	21479	3
	2	36960
	2	3
	14862	1
	1	3
	8600	3
	7974	11985
	3	1
	4	202
	7457	2
	3	3
	5415	33908
	1	4
	3	20727
	34640	30593
	3	25152
	2662	2
	25015	1
	14760	1

Aviso!

?

Input		Got	
	6369	26329	
	1705	2	
	2460	25297	
	17402	42972	
	1	31709	
	3	4	
	1	14255	
	42583	2414	
	1102	16635	
	14445	17694	
	14859	1	
	20160	20985	
	1	1	
	2	13531	
	4	18965	
	39141	16775	
	1	2702	
	1	22639	
	3	1	
	4608	35808	
	3723	14769	
	29543	9192	
	1	31448	
	3	3	
	24863	25030	
	5715	34878	
	10262	2	
	1310	1	
	8093	12047	
	3	1	
	1	2	
	12241	2	
	2248	2	
	2	2	
	2	23454	
	8952	2	
	19065	31646	
	6526	2	
	16549	3	
	10848	1	
	1	1	

Aviso!

?

Input		Got	
	1814	7020	
	41072	3	
	1	1	
	2	11793	
	18936	13367	
	3	3424	
	3	2	
	12684	20499	
	5	2	
	1210	2	
	24947	1	
	1	4428	
	26214	28328	
	5571	17445	
	1	2	
	20291	2	
	41601	9067	
	2	1	
	19990	8201	
	1	15884	
	7332	2115	
	6391	20743	
	1	23361	
	3	19830	
	11710	38325	
	4488	7211	
	2275	1	
	2	2	
	33274	35231	
	36156	13826	
	27977	28267	
	2	2	
	2	1560	
	36823	2	
	2	1	
	3	19756	
	2	29601	
	2	2	
	3	1	
	2	8497	
	2	2528	

Aviso!

?

Input		Got	
		16256	1
		21882	1299
		3	1
		3	9274
		1	2
		27211	1
		32231	14054
		1	3
		33852	2
		1	1
		2834	1
		4	378
		2	2
		15590	28036
		1	2
		25093	18114
		16484	19376
		6819	36122
		2	23308
		1	40594
		37930	14848
		24612	4178
		1	3
		41296	20731
		1	23323
		1	21392
		7922	3868
		2	1
		2	3
		32491	1391
		9791	37101
		16081	2
		17609	4
		1	3
		3	1
		34292	41323
		2	2
		2	31442
		2	25645
		2	25558
		27017	4241

Aviso!

?

Input		Got	
	319	1	
	1	23341	
	3	4	
	2	3	
	1	23558	
	1	6329	
	1	32287	
	2	11483	
	2	14648	
	1	27696	
	15838	31526	
	2	5715	
	7476	3	
	2	2	
	2	31828	
	2	29084	
	14553	16841	
	3	14950	
	1	1	
	2	4198	
	4	5875	
	2	3	
	1	9124	
	5463	3	
	7405	6934	
	33274	1	
	2363	2798	
	7	26831	
	2	1	
	2	1	
	7664	7211	
	1	1023	
	3	2	
	8430	2	
	2	8756	
	25645	1	
	1	1	
	1	3991	
	4767	638	
	1	43607	
	13371	41745	

Aviso!

?

Input		Got	
	2	1	
	2	38280	
	12445	3	
	2	25122	
	1309	2	
	17730	33660	
	5943	1	
	2	3	
	2	3	
	2865	11078	
	1	1	
	3	20571	
	7387	43015	
	1	4	
	2	3892	
	2	39156	
	4	15833	
	2	2	
	8511	3004	
	1953	3	
	4007	22000	
	2	2473	
	2	2	
	2	5	
	10977	41724	
	26150	11250	
	40712	3	
	2	2	
	22876	2	
	6470	1	
	1	4	
	22131	20007	
	2307	20053	
	10981	26	
	18589	2	
	22077	3	
	2031	2	
	31073	4	
	8303	1	
	3	2	
	1475	13811	

Aviso!

?

Input		Got	
		1903	1366
		9762	6267
		16264	10908
		369	2
		4	1
		1	2891
		2	3
		5	3
		5301	1
		1841	18693
		2	4
		1	23437
		1	19271
		42883	2
		11040	10143
		23233	1
		31076	1
		9298	2
		1	3
		25376	2
		1	5973
		19810	5843
		1	1
		6509	12405
		41410	3
		16214	2
		2145	2
		13878	7116
		8138	3060
		7359	28788
		4	2
		4	1
		1	20336
		3	13364
		2	2
		39621	9972
		26812	3
		1	22921
		3	18962
		3	20521
		2	32503

Aviso!

?

Input		Got
	1	43989
	1	1
	2	16904
	4895	1
	2	18896
	15604	9876
	1	23244
	23215	33890
	3811	279
	10283	12352
	4	29061
	1668	1
	1	40684
	1	1
	3	3
	3	36446
	2	1
	2	6770
	3	5481
	5455	1
	4	24472
	3	8
	1	1
	2	8374
	3	625
	1	6657
	3	2
	21134	9875
	3	16144
	4	3
	2910	6776
	25583	23815
	22836	23092
	29332	3
	2	9667
	22938	583
	1	19695
	15607	7293
	1	1
	2	2
	1	1

Aviso!

?

Input		Got
	32459	1
	2	3
	1327	32556
	6536	16887
	3492	2
	2	1
	2	24472
	22075	19818
	1	308
	1	38504
	9698	3334
	1630	41713
	3	2
	3	28199
	2	3
	2	30364
	22118	4
	15274	2
	4360	2
	1	67
	600	13164
	3211	1
	30409	912
	1	22320
	26815	2
	4	10303
	1	21589
	1090	1
	1	24947
	2	1
	17816	24597
	2	1
	2	1581
	11951	2
	2	2677
	3	17809
	19462	41750
	1	23974
	15275	3
	323	2
	13692	20859

Aviso!

?

Input		Got
	4	2
	1	2
	2	2
	1	13820
	2	2523
	3	4988
	2	2
	1	8914
	1	39385
	1	2
	12063	2940
	3	4135
	21773	32955
	3	4909
	8751	1
	34919	1
	31918	6109
	1	1
	3	3
	13175	9593
	28315	15930
	33315	19006
	19130	1
	2	8413
	2	1
	28036	23991
	2	3635
	3158	3
	2	4173
	2	27646
	8766	2
	18492	5396
	2	2776
	1	13397
	1	2
	1	12809
	4	
	2	***Time
	7681	limit
	22184	exceeded***
	2	

Aviso!

?

Input	Got
	569 1 3 1 18786 11784 1 5206 36174 2106 2 3982 2 32652 2 1 6367 13997 2 2 5997 2 4 2

Testing was aborted due to error.

[Show differences](#)

Para resolver o problema, deve-se usar o dicionário e usar a chave como o valor do inteiro e o valor da chave é uma lista em que o primeiro elemento representa o primeiro índice da ocorrência desse elemento no [vetor](#) e o segundo elemento representa quantas ocorrências desse número possuem no [vetor](#). Após isso, percorre-se o [vetor](#) preenchendo esse dicionário. Perceba que uma solução percorrendo o [vetor](#) para cada pergunta levaria a TLE.

Question author's solution (Python3):

Aviso! `= int(input())`

```
3 | vetor = [int(x) for x in input().split()]
4 |
5 | marca = {}
6 |
7 | for idx in range(len(vetor)):
8 |     if not vetor[idx] in marca:
9 |         marca[vetor[idx]] = [idx, 1]
10 |    else:
11 |        marca[vetor[idx]][1] += 1
12 |
13 | q = int(input())
14 |
15 | for i in range(q):
```

Parcialmente correto

Notas para este envio: 0,29/1,00.

Aviso!

?

Questão 8

Correto

Atingiu 1,00 de 1,00

Palíndromo Mais Longo

Dada uma [string](#) s com apenas letras minúsculas, imprima o comprimento do palíndromo mais longo que pode ser construído com essas letras.

Entrada

A única linha de entrada contém uma [string](#) $1 \leq |s| \leq 10^5$.

Saída

Imprima o comprimento do palíndromo mais longo que pode ser construído com essas letras.

Notas

- No primeiro exemplo de teste, a partir da [string](#) de entrada "abccccdd", o palíndromo mais longo que pode ser obtido possui comprimento 7. As possibilidades são "dccaccd", "dccbccd", "ccdadcc", "cdcacdc", "ccdbdcc", "cdcbcdc".

For example:

Input	Result
abccccdd	7
a	1
bb	2

Answer: (penalty regime: 0, 0, 10, 20, ... %)

```
1 entrada = input()
```

Aviso!

```
alfabeto = {letra : entrada.count(letra) for letra in 'abcdefghijklmnopqrstuvwxyz'}
```

```

4 soma_pares = 0
5 existeImpar = False
6
7 for letra in alfabeto:
8     if alfabeto[letra] % 2 != 0:
9         soma_pares += alfabeto[letra] - 1
10        existeImpar = True
11    else:
12        soma_pares += alfabeto[letra]
13
14 if existeImpar:
15     print(soma_pares + 1)

```

	Input	Expected	Got	
✓	abccccdd	7	7	✓
✓	a	1	1	✓
✓	bb	2	2	✓
✓	dwjqiodqhwjiurhw	15	15	✓
✓	heuiwuhtndqnwut	11	11	✓
✓	qwrhcnqwgrcqiwygrqiuygtqwiuytgqcnuygexiquyg	39	39	✓

Passou em todos os teste! ✓

Para resolver esse problema, deve-se contabilizar a frequência de cada letra em um dicionário (ou histograma) e verificar se a frequência de cada letra é par ou ímpar. No caso da frequência de uma letra ser par, ela pode compor uma formação palíndroma da entrada. Caso contrário, deve-se descontar uma unidade da frequência dessa letra, pois deve-se considerar na string

Aviso!

?

palíndrome a frequência par dessa letra. Por exemplo, se a frequência de uma letra é 3, pode-se utilizar 2 ocorrências dela para constituir a string palíndrome, enquanto que a ocorrência restante pode ser utilizada no meio do palíndrome, caso seja possível.

Question author's solution (Python3):

```
1 s = input()
2
3 cnt = {}
4
5 for c in s:
6     if c not in cnt:
7         cnt[c] = 0
8         cnt[c] += 1
9
10 length = 0
11 odd = 0
12
13 for v in cnt.values():
14     length += (v // 2) * 2
15
```

Correto

Notas para este envio: 1,00/1,00.

Aviso!

?

Questão 9

Correto

Atingiu 1,00 de 1,00

Telefone novo

Os alunos de CIC e FT gostam de comprar celulares modernos. Não tendo muito dinheiro, eles raramente compram novos dispositivos, e tentam comprar sempre aquele com o melhor custo-benefício, apenas **tentam**. Eles confiam na opinião de seus amigos e escolhem um dispositivo do tipo que o maior número de seus amigos possui. Se houver vários desses dispositivos, os alunos escolhem o mais barato deles. O aluno Lius não é exceção. Ele questionou seus quatro amigos e agora quer escolher um novo dispositivo de acordo com a regra acima. Observe que os amigos podem comprar o mesmo dispositivo em lugares diferentes com preços diferentes, e Lius pode comprar este dispositivo pelo menor preço.

Entrada

A entrada contém descrições de dispositivos de quatro amigos. Cada descrição tem três linhas. Você recebe o nome de um amigo na primeira linha, o nome do dispositivo desse amigo na segunda linha e o preço do dispositivo na terceira linha, e assim por diante para os quatro amigos. Todas as palavras são strings, e os preços são inteiros menores que 40000 reais.

Saída

Na única linha, o [algoritmo](#) deverá imprimir o nome do dispositivo em posse do maior número de amigos. Se houver vários desses dispositivos, imprima o nome do mais barato deles.

Notas

- No primeiro exemplo de teste, o modelo "nexus4" é maioria (2 ocorrências) entre os quatros modelos de seus amigos.
- No segundo exemplo de teste, três, dos quatro amigos, possuem o modelo "iphone". Assim, Lius prefere o modelo "iphone".
- No terceiro exemplo de teste, cada amigo possui um modelo de celular diferente, e assim o critério de desempate é adotado: o modelo "pedaço de papelão pintado com tinta guache" possui o menor valor.

For example:

Input	Result
-------	--------

Aviso!

?

Input	Result
augusto nexus4 13000 jorge kurtka 20000 dan nexus4 12000 mike htc 10000	nexus4
jorginho iphone 20000 nelsinho iphone 23000 joãozinho motorola v3 2000 carolzinha iphone 9000	iphone
jorginho galaxy s10 4000 nelsinho pedaço de papelão pintado com tinta guache 499 joãozinho motorola v3 500 carolzinha iphone 9000	pedaço de papelão pintado com tinta guache

Aviso!

?

Answer: (penalty regime: 0, 0, 10, 20, ... %)

```

1 QTD_AMIGOS = 4
2 celulares = dict() #keys = modelo, values = [preco, qtd_no_grupo]
3 PRECO, QTD_NO_GRUPO = 0, 1
4
5 for x in range(QTD_AMIGOS):
6     input()
7     modelo = input()
8     preco = int(input())
9     if modelo in celulares.keys():
10         celulares[modelo][QTD_NO_GRUPO] += 1
11         if preco < celulares[modelo][PRECO]:
12             celulares[modelo][PRECO] = preco
13     else:
14         celulares[modelo] = [preco, 1]
15

```

	Input	Expected	Got	
✓	augusto nexus4 13000 jorge kurtka 20000 dan nexus4 12000 mike htc 10000	nexus4	nexus4	✓

Aviso!

?

	Input	Expected	Got	
✓	jorginho iphone 20000 nelsinho iphone 23000 joãozinho motorola v3 2000 carolzinha iphone 9000	iphone	iphone	✓
✓	jorginho galaxy s10 4000 nelsinho pedaço de papelão pintado com tinta guache 499 joãozinho motorola v3 500 carolzinha iphone 9000	pedaço de papelão pintado com tinta guache	pedaço de papelão pintado com tinta guache	✓
✓	jorginho galaxy s10 4000 nelsinho pedaço de papelão pintado 501 joãozinho motorola v3 500 carolzinha iphone 9000	motorola v3	motorola v3	✓

Aviso!

?

	Input	Expected	Got	
✓	jorginho nexus4 1200 nelsinho iphone 23000 joãozinho nexus4 2000 carolzinha iphone 9000	nexus4	nexus4	✓
✓	jorginho iphone 20000 nelsinho iphone 23000 joãozinho motorola v3 2000 carolzinha motorola v3 9000	motorola v3	motorola v3	✓

Passou em todos os teste! ✓

Para resolver o problema, deve-se ler os modelos e os preços dos celulares da entrada, e armazenar em um dicionário, as [informações](#) referentes ao modelo, a quantidade de celulares desse modelo existentes entre seus quatro amigos e o menor valor de cada modelo (tratando-se assim, possíveis repetições).

Question author's solution (Python3):

```

1 dic = {}
2
3 for _ in range(4):
4     name = input()
5     price = input()

```

```
5 |     phone = input()
6 |     price = int(input())
7 |     if phone in dic:
8 |         dic[phone].append(price)
9 |     else:
10 |         dic[phone] = [price]
11 | name = ""
12 | price = 999999999
13 | c = -9999
14 | for phone in dic.keys():
15 |     cump = min(dic[phone])
```

Correto

Notas para este envio: 1,00/1,00.

Aviso!

?

Questão 10

Correto

Atingiu 1,00 de 1,00

Contando Consoantes e Vogais

Crie um [programa](#) que lê uma [string](#) *frase* e conta o número de consoantes e vogais encontradas na [string](#), armazenando esses valores em um dicionário de nome *numeros* com chaves de nomes *consoantes* e *vogais*. Após isso, imprima o dicionário *numeros*.

Desconsidere letras maiúsculas e acentuadas. As letras alfabéticas que aparecem nas strings de entrada são todas minúsculas.

Entrada

A entrada consiste em uma [string](#).

Saída

A saída consiste no dicionário *numeros*.

Observações

- No primeiro exemplo de teste, foi digitado "oi", retornando {'consoantes': 0, 'vogais': 2}.
- No segundo exemplo de teste, foi digitado "uma pequena frase" retornando {'consoantes': 7, 'vogais': 8}.
- No terceiro exemplo de teste, foi digitado "dinossauro", retornando {'consoantes': 5, 'vogais': 5}.

For example:

Input	Result
oi	{'consoantes': 0, 'vogais': 2}
uma pequena frase	{'consoantes': 7, 'vogais': 8}
Aviso! ro	{'consoantes': 5, 'vogais': 5}

Answer: (penalty regime: 0, 0, 10, 20, ... %)

```

1 frase = input()
2
3 def contaLetras(palavra, categoria):
4
5     qtd_vogais = 0
6     palavraLower = palavra.lower()
7
8     for l in palavraLower:
9         if l in categoria.lower():
10             qtd_vogais += 1
11
12     return qtd_vogais
13
14 numeros = {
15     'consoantes': contaLetras(frase, 'bcdfghiklmnopqrstuvxz')

```

	Input	Expected	Got	
✓	oi	{'consoantes': 0, 'vogais': 2}	{'consoantes': 0, 'vogais': 2}	✓
✓	uma pequena frase	{'consoantes': 7, 'vogais': 8}	{'consoantes': 7, 'vogais': 8}	✓
✓	dinossauro	{'consoantes': 5, 'vogais': 5}	{'consoantes': 5, 'vogais': 5}	✓
✓	abcdefghijklmnopqrstuvwxyz	{'consoantes': 18, 'vogais': 5}	{'consoantes': 18, 'vogais': 5}	✓
✓	comprei vinte dvds na quitanda ao lado!	{'consoantes': 18, 'vogais': 14}	{'consoantes': 18, 'vogais': 14}	✓
	o fim, vamos testar mais uma vez	{'consoantes': 16, 'vogais': 11}	{'consoantes': 16, 'vogais': 11}	✓

Aviso!

?

Passou em todos os teste! ✓

Question author's solution (Python3):

```
1 numeros = {"consoantes": 0, "vogais": 0}
2 palavra = input()
3 vog = "aeiou"
4 cons = "bcdfghjklmnpqrstvwxyz"
5
6 for caract in palavra:
7     if caract in vog:
8         numeros["vogais"] += 1
9     elif caract in cons:
10        numeros["consoantes"] += 1
```

Correto

Notas para este envio: 1,00/1,00.



Aviso!

