

5ª. Hackathon

FATEC Garça - FullTime

(14 de maio de 2022)

Caderno de Problemas

Informações Gerais

- Este caderno contém 10 problemas – A a J, com páginas numeradas de 01 a 14. Verifique se o caderno está completo.
- A competição possui duração de 4 horas (início as 13:00h término as 17:00h);
- Durante a competição é permitida aos competidores a consulta a material próprio impresso em papel (livros, apostilas, anotações);
- NÃO é permitido acesso a conteúdo da Internet e nem conteúdos disponíveis em qualquer meio magnético;
- É vedada a comunicação entre os competidores durante a competição, bem como a troca de material de consulta entre eles;
- Cada competidor terá acesso a 1 computador dotado do ambiente de submissão de programas BOCA, dos compiladores, link-editores e IDEs requeridos pelas linguagens de programação permitidas;
- Não é permitido o uso de notebooks ou outro tipo de computador ou assistente pessoal;
- Os problemas têm o mesmo valor na correção;
- A correção é automatizada, portanto siga atentamente as exigências do problema quanto ao formato da entrada e saída de seu programa;
- Não implemente nenhum recurso gráfico nas suas soluções (janelas, menus, etc.), todos os dados devem ser lidos da entrada padrão e escritos na saída padrão. Nem utilize qualquer rotina para limpar a tela ou posicionar o cursor;
- Preste muita atenção no nome dos arquivos fonte indicados nos problemas;
- As soluções serão testadas com outras entradas além das apresentadas como exemplo nos problemas;

Problema A

Tempo de Jogo

Nome do arquivo fonte: `jogo.{c|cpp|java|py}`

Leia a hora inicial e a hora final de um jogo. A seguir calcule a duração do jogo, sabendo que o mesmo pode começar em um dia e terminar em outro, tendo uma duração mínima de 1 hora e máxima de 24 horas.

Entrada

A entrada contém dois valores inteiros representando a hora de início e a hora de fim do jogo.

Saída

Apresente a duração do jogo conforme exemplo abaixo.

Exemplos

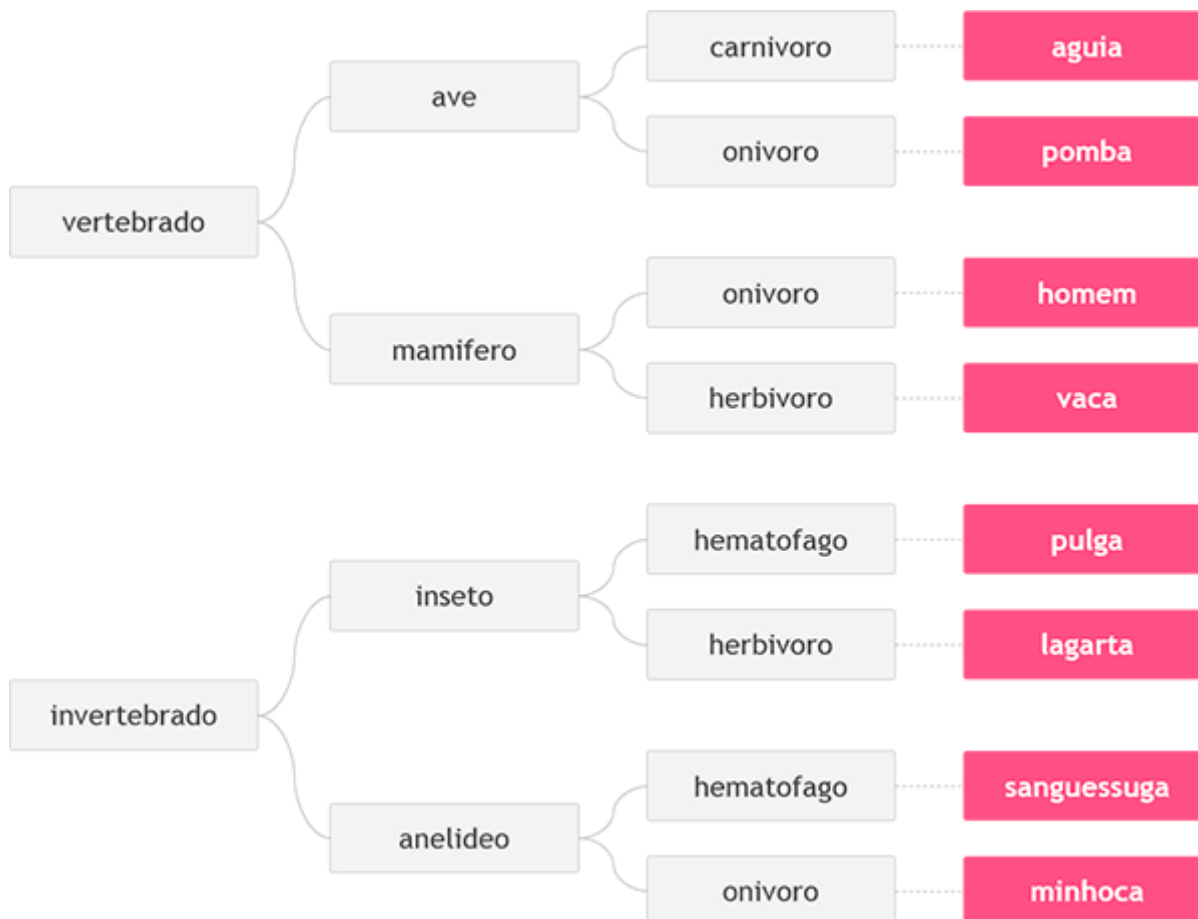
Entrada 16 2	Saída O JOGO DUROU 10 HORA(S)
Entrada 0 0	Saída O JOGO DUROU 24 HORA(S)
Entrada 2 16	Saída O JOGO DUROU 14 HORA(S)

Problema B

Animal

Nome do arquivo fonte: `animal.{c|cpp|java|py}`

Neste problema, você deverá ler 3 palavras que definem o tipo de animal possível segundo o esquema abaixo, da esquerda para a direita. Em seguida conclua qual dos animais seguintes foi escolhido, através das três palavras fornecidas.



Entrada

A entrada contém 3 palavras, uma em cada linha, necessárias para identificar o animal segundo a figura acima, com todas as letras minúsculas.

Saída

Imprima o nome do animal correspondente à entrada fornecida.

Exemplos

Entrada	Saída
vertebrado mamifero onivoro	homem

Entrada vertebrado ave carnivoro	Saída aguia
--	-----------------------

Entrada invertebrado anelideo onivoro	Saída minhoca
---	-------------------------

Problema C

Sequência Lógica

Nome do arquivo fonte: `sequencia.{c|cpp|java|py}`

Escreva um programa que leia um valor inteiro N . $N * 2$ linhas de saída serão apresentadas na execução do programa, seguindo a lógica do exemplo abaixo. Para valores com mais de 6 dígitos, todos os dígitos devem ser apresentados.

Entrada

O arquivo de entrada contém um número inteiro positivo N ($1 < N < 1000$).

Saída

Imprima a saída conforme o exemplo fornecido.

Exemplos

Entrada	Saída
5	1 1 1 1 2 2 2 4 8 2 5 9 3 9 27 3 10 28 4 16 64 4 17 65 5 25 125 5 26 126

Problema D

A Corrida de Lesmas

Nome do arquivo fonte: `lesmas.{c|cpp|java|py}`

A corrida de lesmas é um esporte que cresceu muito nos últimos anos, fazendo com que várias pessoas dediquem suas vidas tentando capturar lesmas velozes, e treina-las para faturar milhões em corridas pelo mundo. Porém a tarefa de capturar lesmas velozes não é uma tarefa muito fácil, pois praticamente todas as lesmas são muito lentas. Cada lesma é classificada em um nível dependendo de sua velocidade:

Nível 1: Se a velocidade é menor que 10 cm/h .

Nível 2: Se a velocidade é maior ou igual a 10 cm/h e menor que 20 cm/h .

Nível 3: Se a velocidade é maior ou igual a 20 cm/h .

Sua tarefa é identificar qual nível de velocidade da lesma mais veloz de um grupo de lesmas.

Entrada

A entrada consiste em duas linhas: A primeira linha contém um inteiro L ($1 \leq L \leq 500$) representando o número de lesmas do grupo, e a segunda linha contém L inteiros V_i ($1 \leq V_i \leq 50$) representando as velocidades de cada lesma do grupo.

Saída

Para cada caso de teste, imprima uma única linha indicando o nível de velocidade da lesma mais veloz do grupo.

Exemplos

Entrada 10 10 10 10 10 15 18 20 15 11 10	Saída 3
---	-------------------

Entrada 10 1 5 2 9 5 5 8 4 4 3	Saída 1
---	-------------------

Entrada 10 19 9 1 4 5 8 6 11 9 7	Saída 2
---	-------------------

Problema E

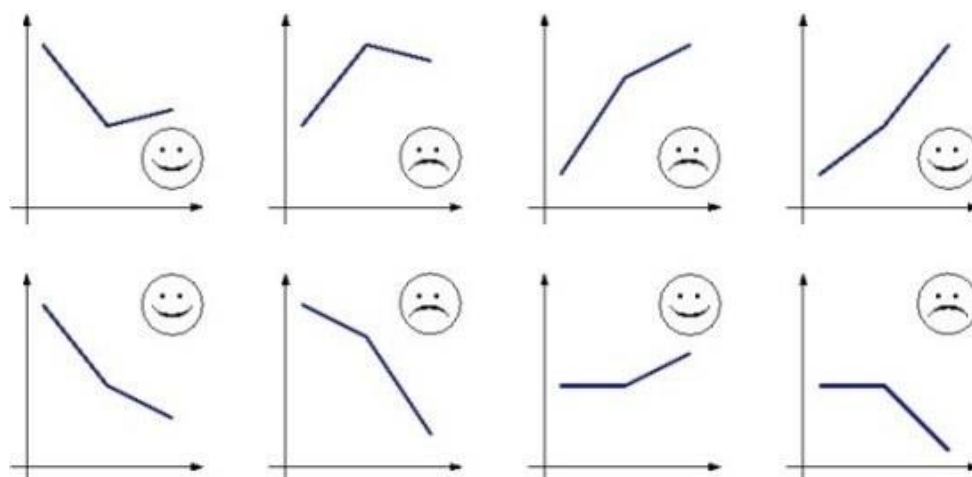
Bem-vindos e Bem-vindas ao Inverno!

Nome do arquivo fonte: `inverno.{c|cpp|java|py}`

Bem-vindos e bem-vindas à Escola de Inverno da Maratona de Programação 2015 de Erechim! Esperamos sinceramente que vocês aprendam muito nestes dias para que tenham muito sucesso nas competições de Programação ainda por vir, mas sobretudo esperamos que vocês curtam a Escola, pois quando nos divertimos e temos prazer em estudar e programar, o treino deixa de ser um fardo e se torna um hobby. Então, divirtam-se!

O inverno é uma estação maravilhosa, não é mesmo? Todos nós amamos vestir um poncho, participar de uma roda de chimarrão, assar pinhões no fogão a lenha... Mas nem todos gostam do inverno, especialmente em lugares onde o inverno costuma ser muito cruel. Em Westeros, por exemplo, o humor das pessoas é definido de acordo com as tendências climáticas. Com base nas temperaturas dos três últimos dias, as pessoas podem ficar tristes ou felizes, ficando mais propensas a fazer guerra ou fazer amor, respectivamente. E, sejamos sinceros, é justamente por causa das cenas de amor e de guerra que amamos Game of Thrones!

- Se a temperatura desceu do 1º para o 2º dia, mas subiu ou permaneceu constante do 2º para o 3º, as pessoas ficam felizes (primeira figura).
- Se a temperatura subiu do 1º para o 2º dia, mas desceu ou permaneceu constante do 2º para o 3º, as pessoas ficam tristes (segunda figura).
- Se a temperatura subiu do 1º para o 2º dia e do 2º para o 3º, mas subiu do 2º para o 3º menos do que subira do 1º para o 2º, as pessoas ficam tristes (terceira figura).
- Se a temperatura subiu do 1º para o 2º dia e do 2º para o 3º, mas subiu do 2º para o 3º no mínimo o tanto que subira do 1º para o 2º, as pessoas ficam felizes (quarta figura).
- Se a temperatura desceu do 1º para o 2º dia e do 2º para o 3º, mas desceu do 2º para o 3º menos do que descera do 1º para o 2º, as pessoas ficam felizes (quinta figura).
- Se a temperatura desceu do 1º para o 2º dia e do 2º para o 3º, mas desceu do 2º para o 3º no mínimo o tanto que descera do 1º para o 2º, as pessoas ficam tristes (sexta figura).
- Se a temperatura permaneceu constante do 1º para o 2º dia, as pessoas ficam felizes se subiu do 2º para o 3º dia ou tristes caso contrário (respectivamente, sétima e oitava figuras).



Entrada

A entrada consiste apenas de três inteiros, **A**, **B** e **C** ($-100 \leq \mathbf{A}, \mathbf{B}, \mathbf{C} \leq 100$), os quais representam respectivamente as temperaturas registradas no 1º, no 2º e no 3º dias.

Saída

Imprima uma linha contendo uma carinha feliz ou triste, representando como fica o humor do povo de Westeros de acordo com as tendências climáticas.

Exemplos

Entrada 20 10 12	Saída :)
----------------------------	--------------------

Entrada 10 20 18	Saída :(
----------------------------	--------------------

Entrada 4 16 20	Saída :(
---------------------------	--------------------

Entrada 4 10 20	Saída :)
---------------------------	--------------------

Entrada 20 10 6	Saída :)
---------------------------	--------------------

Entrada 20 16 4	Saída :(
---------------------------	--------------------

Entrada 10 10 14	Saída :)
----------------------------	--------------------

Entrada 10 10 2	Saída :(
---------------------------	--------------------

Problema F

Corvo Contador

Nome do arquivo fonte: `corvo.{c|cpp|java|py}`

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

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

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

Entrada

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

Um grito é representado pela string **caw caw**. Uma piscada é representada por três caracteres ***** ou **-**, representando, respectivamente, um olho aberto ou um olho fechado, da esquerda para a direita. Lembre-se que o corvo tem **3** olhos.

Os números sorteados na loteria não excedem 1000.

Saída

A saída são três linhas, cada linha com um número da loteria.

Exemplos

Entrada	Saída
--*	1
caw caw	4
*--	0
caw caw	
caw caw	

Entrada	Saída
--*	3
--*	8
--*	5
caw caw	
*--	
*--	
caw caw	
--*	
*--	
caw caw	

Problema G

Numeração Romana para Números de Página

Nome do arquivo fonte: romano.{c|cpp|java|py}

A ECI (*Editio Chronica Incredibilis* ou Editora de Crônicas Incríveis) é muito tradicional quando se trata de numerar as páginas de seus livros. Ela sempre usa a numeração romana para isso. E seus livros nunca ultrapassam as 999 páginas pois, quando necessário, dividem o livro em volumes.

Você deve escrever um programa que, dado um número arábico, mostra seu equivalente na numeração romana.

Lembre que I representa 1, V é 5, X é 10, L é 50, C é 100, D é 500 e M representa 1000.

Entrada

A entrada é um número inteiro positivo N ($0 < N < 1000$).

Saída

A saída é o número N escrito na numeração romana em uma única linha. Use sempre letras maiúsculas.

Exemplos

Entrada 666	Saída DCLXVI
Entrada 83	Saída LXXXIII
Entrada 999	Saída CMXCIX

Problema H

De Dentro para Fora

Nome do arquivo fonte: `impressora.{c|cpp|java|py}`

A sua impressora foi infectada por um vírus e está imprimindo de forma incorreta. Depois de olhar para várias páginas impressas por um tempo, você percebe que ele está imprimindo cada linha de dentro para fora. Em outras palavras, a metade esquerda de cada linha está sendo impressa a partir do meio da página até a margem esquerda. Do mesmo modo, a metade direita de cada linha está sendo impressa à partir da margem direita e prosseguindo em direção ao centro da página.

Por exemplo a linha:

THIS LINE IS GIBBERISH

está sendo impressa como:

I ENIL SIHTHSIREBBIG S

Da mesma forma, a linha " MANGOS " está sendo impressa incorretamente como "NAM SOG". Sua tarefa é desembaralhar (decifrar) a string a partir da forma como ela foi impressa para a sua forma original. Você pode assumir que cada linha conterá um número par de caracteres.

Entrada

A entrada contém vários casos de teste. A primeira linha de entrada contém um inteiro **N** que indica a quantidade de casos de teste. Seguem **N** linhas, cada uma com uma frase com no mínimo 2 e no máximo 100 caracteres de letras maiúsculas e espaços que deverá ser desembaralhada (decifrada) à partir da forma impressa para a sua forma original, conforme especificação acima.

Saída

Para cada linha de entrada deverá ser impressa uma linha de saída com a frase decifrada, conforme a especificação acima.

Exemplos

Entrada	Saída
5	
I ENIL SIHTHSIREBBIG S	THIS LINE IS GIBBERISH
LEVELKAYAK	LEVELKAYAK
H YPPAHSYADILO	HAPPY HOLIDAYS
ABCDEFGHIJKLMNPOQRSTUVWXYZ	MLKJIHG FEDCBAZYXWVUTSRQPON
VOD OWT SNEH HCNERF EGDITRAP A DNA SE	FRENCH HENS TWO DOVES AND A PARTRIDGE

Problema I

Conjectura de Collatz

Nome do arquivo fonte: `collatz.{c|cpp|java|py}`

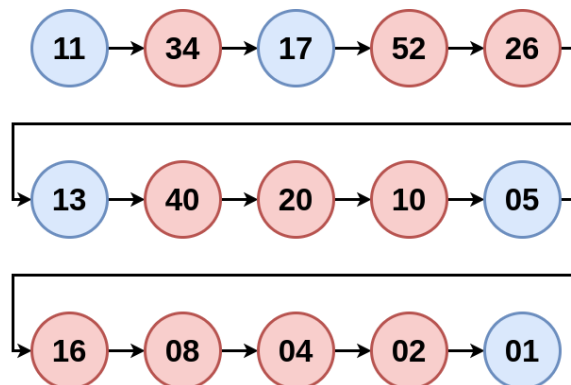
A Conjectura de Collatz, conhecida também por Problema $3x+1$, foi proposta pelo matemático alemão Lothar Collatz em 1937 e permanece em aberto desde então. Seu enunciado é bastante simples e requer apenas um número inteiro positivo x como entrada para o problema. Dado um número inteiro positivo x , caso x seja par, divida-o por dois. Caso contrário, multiplique-o por três e some um. O número resultante de uma dessas operações será o novo valor de x . Repita esse processo até que o valor de x seja igual a 1.

De maneira mais formal, temos que:

$$f(x) = \begin{cases} \frac{x}{2}, & \text{se } x \text{ for par} \\ 3x + 1, & \text{caso contrário} \end{cases}$$

Por exemplo, suponha que x seja inicialmente igual a 11. O número 11 é ímpar, dessa forma o novo valor de x será $(3 \cdot 11) + 1 = 34$. Repetindo o processo temos que o número 34 é par, logo o novo valor de x será $34/2 = 17$. Repetindo esse processo chegaremos eventualmente no número 1, quando o processo acaba.

A imagem a seguir mostra a sequência gerada a partir do número 11 até chegar no número 1. Os números com fundo na cor azul correspondem aos ímpares gerados, enquanto os com fundo na cor vermelha representam os números pares.



Note que os valores na sequência podem aumentar ou diminuir dependendo da paridade de x . A princípio, podemos ter a impressão de que esse processo pode nunca terminar, com o número 1 nunca sendo alcançado. Entretanto, o processo descrito acima foi testado em uma imensa quantidade de números inteiros positivos e até hoje nenhum contraexemplo foi encontrado. Contudo, uma prova matemática de que esse processo funciona para qualquer número inteiro positivo também não foi apresentada ainda.

O objetivo dessa atividade prática consiste em criarmos um programa que gere a sequência de números descrita pela Conjectura de Collatz a partir de um número inteiro positivo x .

Entrada

Como entrada, seu programa receberá um número inteiro positivo N. Em seguida, seu programa receberá N números inteiros positivos. Para cada um dos N números inteiros positivos, seu programa deverá gerar a sequência de números seguindo a descrição apresentada pela Conjectura de Collatz. Além disso, o programa deverá ser capaz de computar a quantidade de números pares e ímpares na sequência, bem como o maior número presente na mesma.

Saída

Como saída, para cada um dos N números inteiros positivos, seu programa deverá imprimir essas informações no seguinte padrão:

Valor inicial: <X>
 Numeros Pares: <PARES>
 Numeros Impares: <IMPARES>
 Maior Numero: <MAX>

Onde <X>, <PARES>, <IMPARES> e <MAX> representam respectivamente o valor inicial dado como entrada, a quantidade de números pares na sequência, a quantidade de números ímpares na sequência e o maior número da sequência.

Exemplos

Entrada	Saída
1 11	Valor inicial: 11 Numeros Pares: 10 Numeros Impares: 5 Maior Numero: 52

Entrada	Saída
3 33 67 56	Valor inicial: 33 Numeros Pares: 18 Numeros Impares: 9 Maior Numero: 100 Valor inicial: 67 Numeros Pares: 19 Numeros Impares: 9 Maior Numero: 304 Valor inicial: 56 Numeros Pares: 14 Numeros Impares: 6 Maior Numero: 56

Entrada	Saída
2 9 15	Valor inicial: 9 Numeros Pares: 13 Numeros Impares: 7 Maior Numero: 52 Valor inicial: 15 Numeros Pares: 12 Numeros Impares: 6 Maior Numero: 160

Problema J

Eleição 2022

Nome do arquivo fonte: `eleicao.{c|cpp|java|py}`

O Brasil adota o sistema eleitoral majoritário de dois turnos e o critério de maioria absoluta nas suas eleições. Para ser eleito no primeiro turno, o candidato precisa obter mais da metade dos votos válidos (excluídos os votos em branco e os votos nulos). Caso contrário, os dois candidatos mais votados no primeiro turno irão compor o segundo turno da eleição.

Você foi contratado pelo Tribunal Superior Eleitoral (TSE) para fazer um programa que indique se houve um vencedor no primeiro turno da eleição de 2022 ou se será necessário a realização de um segundo turno. No caso de haver um vencedor, você deve indicar quem foi o vencedor. Já no caso em que um segundo turno é necessário, você deve indicar os dois candidatos que participarão do segundo turno, seguindo a ordem de número de votos (o mais votado será impresso primeiro).

Entrada

O TSE fornecerá para o seu programa a seguinte entrada:

- Uma linha com a quantidade n de candidatos da eleição;
- Uma lista com o nome dos n candidatos, sendo um nome por linha;
- Uma lista com $n+2$ valores inteiros, um por linha, sendo que a i -ésima linha representa a quantidade de votos recebidos pelo i -ésimo candidato, seguindo a ordem de leitura da lista de nomes. Os últimos dois valores representam a quantidade de votos brancos e nulos, respectivamente.

Você pode assumir que todos os candidatos receberam uma quantidade distinta de votos.

Saída

A impressão deve seguir os modelos abaixo.

Quando existe vencedor no primeiro turno:

```
<nome_candidato_vencedor> foi o vencedor da eleição
```

Quando há a necessidade de segundo turno:

```
Haverá segundo turno entre:
```

```
<nome_primeiro_colocado>
```

```
<nome_segundo_colocado>
```

Além disso, o seu programa deve imprimir o total de votos e o total de votos válidos, respectivamente, seguindo o modelo abaixo.

```
Total de votos: <total de votos>
```

```
Votos válidos: <número de votos válidos>
```

Exemplos

<p>Entrada</p> <p>4 Camila João Natália Maria 110000 500000 50000 90000 80000 100000</p>	<p>Saída</p> <p>João foi o vencedor da eleição Total de votos: 930000 Votos válidos: 750000</p>
---	--

<p>Entrada</p> <p>5 Francisco Carla Rodrigo Roberto Mariana 5962728 17748006 38141123 25700444 1577537 1926300 8143011</p>	<p>Saída</p> <p>Haverá segundo turno entre: Rodrigo Roberto Total de votos: 99199149 Votos válidos: 89129838</p>
---	---

<p>Entrada</p> <p>6 Natália Emanuel João Francisco Rodrigo Roberto 13477809 1875820 6727180 824926 60804689 1226973 3448698 5764964</p>	<p>Saída</p> <p>Rodrigo foi o vencedor da eleição Total de votos: 94151059 Votos válidos: 84937397</p>
--	---