ToPAS'17
Torneio de Programação para Alunos do Secundário

Departamento de Ciência de Computadores

http://topas.dcc.fc.up.pt

Conjunto de Problemas

U. PORTO

Faculdade de Ciências da Universidade do Porto

12 de maio de 2017

Este conjunto de problemas deverá conter sete (7) problemas e dezoito (18) páginas.
Se faltar algum problema, por favor avise a organização.

Edição realizada em colaboração com:

UAlg FCT UNIVERSIDADE DO ALGARVE FACULDADE DE CIÊNCIAS E TECNOLOGIA

Departamento de Engenharia Electrónica e Informática, Faculdade de Ciências e Tecnologia, Universidade do Algarve

FCT FACULDADE DE CIÊNCIAS E TECNOLOGIA UNIVERSIDADE NOVA DE LISBOA

Departamento de Informática, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa

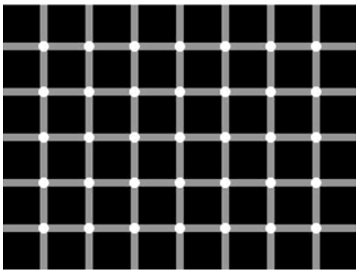
ToPAS'17
Torneio de Programação para Alunos do Secundário

Dep. Ciência de Computadores – FCUP
12 de maio de 2017

Conteúdo

Chão de ilusão

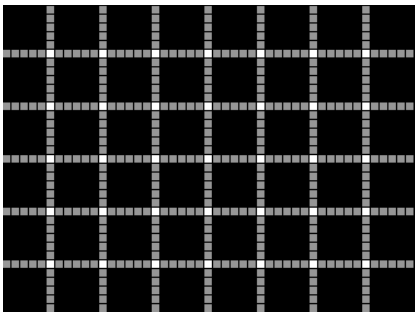
Fiquei fascinado com esta imagem, que descobri na Internet:



Olhando para ela descontraidamente, parece-me que os círculos brancos se tornam pretos instantaneamente, quando os meus olhos passam por eles. Mas se me fixar em qualquer um dos círculos, vejo que é branco e que a cor não muda.

Como vou fazer obras na minha cozinha, decidi reproduzir o padrão daquela imagem fantástica no chão. Para isso vou usar mosaicos quadrados de três cores — preto, cinzento e branco — e de dois tamanhos — grande e pequeno.

Os mosaicos grandes são pretos e preencherão a maior parte do chão, tal como na imagem. Entre os mosaicos pretos haverá fiadas de mosaicos cinzentos, pequenos, para representar as faixas cinzentas na figura. Finalmente, nos locais onde estão os círculos brancos, colocarei um mosaico branco, do mesmo tamanho dos mosaicos cinzentos. Em esquema, ficará assim:



Como sou perfeccionista, quero que os mosaicos preencham completamente o chão da cozinha, com quatro mosaicos pretos exatamente nos cantos, e os outros distribuídos precisamente de acordo com o esquema.

Tarefa

Escreva um programa que, dados o comprimento e a largura da cozinha, a medida do lado dos azulejos pretos e a medida do lado dos azulejos cinzentos (e dos azulejos brancos, que é a mesma), verifique se é possível preencher o chão da cozinha com o padrão que pretendo e, se for possível preencher o chão, calcule o número de mosaicos pretos, o número de mosaicos cinzentos e o número de mosaicos brancos que tenho de comprar.

Input

O programa lê da consola quatro números inteiros positivos, representando, por esta ordem, o comprimento da cozinha, a largura da cozinha, a medida do lado dos azulejos pretos e a medida do lado dos azulejos cinzentos (que é a mesma que a medida do lado dos azulejos brancos). A medida do lado dos mosaicos pretos é maior ou igual à medida dos mosaicos cinzentos, garantidamente.

Output

Se não for possível preencher o chão completamente, o programa escreverá uma linha com três asteriscos; se for, o programa escreverá três números, representando o número de mosaicos pretos, o número de mosaicos cinzentos e o número de mosaicos brancos que eu devo comprar.

Exemplo 1

Input

```
94 70 10 2
```

Output

```
48 410 35
```

Exemplo 2

Input

```
40 20 6 2
```

Output

```
***
```

Sistema de rega gota-a-gota

O sistema gota-a-gota é um dos sistemas mais eficazes de rega. De fácil instalação, pode ser aplicado em todos os tipos de terrenos agrícolas e adapta-se facilmente às diferentes culturas. Cada cultura é caracterizada pelo *valor mínimo*, pelo *valor máximo* e pelo *intervalo adequado* da quantidade de água fornecida diariamente a cada planta. A quantidade de água é medida em número de gotas. Por exemplo, a cultura de tomate é caracterizada por:

- Valor mínimo: 150 gotas por dia
- Valor máximo: 500 gotas por dia
- Intervalo adequado: $[250, 355]$ gotas por dia

O sistema de rega gota-a-gota regista diariamente uma sequência não vazia de zeros e uns, que indica se não houve ou se houve a rega de uma gota num dado minuto. O comprimento dessas sequências varia entre 1 e 1440 (24 horas vezes 60 minutos).

Por exemplo, a sequência `0111101111101111000000000011111` indica que a quantidade de água regada neste dia foi 18 gotas (porque há 18 uns).

Tarefa

Escreva um programa que, dados os valores que caracterizam a cultura e as sequências de zeros e uns registadas em alguns dias, indica:

1. O número de dias em que a quantidade de água regada foi menor que o valor mínimo;
2. O número de dias em que a quantidade de água regada excedeu o valor máximo;
3. O número de dias em que a quantidade de água regada foi adequada (encontra-se dentro do intervalo adequado para a cultura).

Input

A primeira linha tem quatro números inteiros (m, M, A e B) que caracterizam a cultura: m é o valor mínimo, M é o valor máximo e $[A, B]$ é o intervalo adequado. A segunda linha tem o número D de dias registados. Cada uma das D restantes linhas tem o registo de um dia, que é uma sequência com C elementos, onde cada elemento é `0` ou `1`.

Restrições

$1 \leq m \leq A \leq B \leq M \leq 1\,000$	Valores que caracterizam a cultura
$1 \leq D \leq 50$	Número de dias registados
$1 \leq C \leq 1\,440$	Comprimento da sequência registada num dia

Output

Uma linha com três números inteiros, que correspondem ao número de dias em que a quantidade de água regada: foi menor que o valor mínimo; foi maior que o valor máximo; foi adequada.

Exemplo 1

Input

```
4 9 4 9
2
00010101010
11111111000000001
```

Output

```
0 0 2
```

Exemplo 2

Input

```
1 1 1 1
2
00000000000000000000000000
0
```

Output

```
2 0 0
```

Exemplo 3

Input

```
3 10 4 9
3
0000000001111111111111111111111110000110101
00010101010111111111111100000111111111111111
1110000000111110100000011
```

Output

```
0 3 0
```

Exemplo 4

Input

```
3 10 4 9
5
00000001111000000000000000000000
000000000000000000000000000000000011111111110000000000000000000
0
111110000000000000000000000000000000000000000000000000000111111
1
```

Output

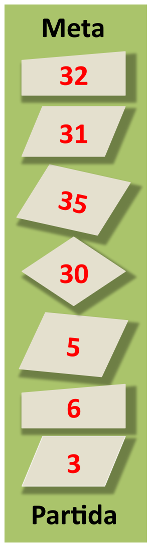
```
2 1 1
```

Salta lajes

Os alunos da Professora Didá gostam muito de brincar ao Salta Lajes no recreio. Antes de cada jogo, a Professora Didá escreve em cada laje um número com giz (como se ilustra na figura). Todos os números são diferentes. Depois, cada criança faz uma sequência de saltos sobre as lajes, começando na partida (antes da primeira laje) e terminando na meta (depois da última laje), obtendo uma pontuação. Ganha quem obtiver a maior pontuação.

Para efetuar a sequência de saltos, a criança coloca-se na partida. No primeiro salto, a criança só pode colocar o pé direito numa laje; no segundo salto, só pode colocar o pé esquerdo; no terceiro salto, só pode colocar o pé direito; e assim sucessivamente, alternando o pé que assenta no chão, até chegar à meta. Todos os saltos são na direção da meta. Cada salto pode ter *comprimento* 1, 2 ou 3, ou seja, a criança pode saltar para a primeira, a segunda ou a terceira laje imediatamente a seguir ao sítio onde se encontra. A sequência de saltos é pontuada somando os números nas lajes tocadas com o pé direito e subtraindo os números nas lajes tocadas com o pé esquerdo.

O Rui, que às vezes ganha e às vezes perde, usa sempre a seguinte estratégia:



- Quando o salto termina com o pé direito no chão: se o Rui puder saltar para uma laje, salta para a laje (de entre as permitidas) com o maior número; se já não houver lajes, salta para a meta.
- Quando o salto termina com o pé esquerdo no chão: se o Rui puder, salta para a meta; se tiver de saltar para uma laje, salta para a laje (de entre as permitidas) com o menor número.

No exemplo da figura, o Rui saltaria para as lajes com os números:

- **6** (primeiro salto, de comprimento 2, com o pé direito, porque $6 > 3$ e $6 > 5$),
- **5** (segundo salto, de comprimento 1, com o pé esquerdo, porque $5 < 30$ e $5 < 35$) e
- **35** (terceiro salto, de comprimento 2, com o pé direito, porque $35 > 30$ e $35 > 31$)

e depois saltaria para a meta (quarto salto, de comprimento 3, com o pé esquerdo). A pontuação do Rui seria $\mathbf{6} - \mathbf{5} + \mathbf{35} = 36$.

Neste exemplo, o comprimento máximo dos saltos é 3, mas não será sempre assim. Quando o comprimento máximo dos saltos é C (com $C \geq 2$), o comprimento de cada salto pode ser qualquer número inteiro entre 1 e C.

Tarefa

Escreva um programa que, dado o comprimento máximo dos saltos e a sequência de números que a Professora Didá escreve nas lajes, calcula a pontuação que o Rui obtém.

Input

A primeira linha tem dois números inteiros: C (o comprimento máximo dos saltos) e L (o número de lajes). Seguem-se L linhas, cada uma com um número inteiro (n) escrito numa laje, pela ordem em que os números ocorrem da partida para a meta. Os números escritos nas lajes são todos distintos, ou seja, nunca há duas lajes com o mesmo número.

Restrições

$2 \leq C \leq 20$	Comprimento máximo de cada salto
$C \leq L \leq 1\,000$	Número de lajes
$1 \leq n \leq 5\,000$	Um número escrito numa laje

Output

Uma linha com um número, que representa a pontuação que o Rui obtém.

Exemplo 1

Input

```
3 7
3
6
5
30
35
31
32
```

Output

```
36
```

Exemplo 2

Input

```
4 8
1
2
3
4
8
7
6
5
```

Output

```
-1
```

O número perdido

O João está a estudar progressões aritméticas nas aulas de matemática, mas está a ter algumas dificuldades. Para o ajudar, a Maria, que é a irmã mais velha, fez um programa que gera os primeiros elementos de uma progressão aritmética crescente, mas que omite um desses elementos, quando os escreve. O elemento omitido nunca é o primeiro nem o último. O João tem de descobrir qual é o *número perdido* (o número que foi omitido).

Por exemplo, se a sequência apresentada pelo programa for `2 4 6 8 12 14`, o número perdido é 10.

A Maria recordou que qualquer termo de uma progressão aritmética (exceto o primeiro) se obtém do anterior somando uma constante, a que se chama razão da progressão aritmética. Neste exemplo, a razão é 2.

Tarefa

Escreva um programa que, dada uma sequência apresentada pelo programa da Maria, descubra qual é o número perdido.

Input

A primeira linha tem o número n de elementos da sequência apresentada pelo programa da Maria. A segunda linha tem n números inteiros, que correspondem aos elementos da sequência.

Restrições

$3 \leq n \leq 150$ Número de elementos da sequência

Output

Uma linha com o número perdido.

Exemplo 1

Input

```
7
1 2 3 4 5 7 8
```

Output

```
6
```

Exemplo 2

Input

```
6
2 4 6 8 12 14
```

Output

```
10
```

Exemplo 3

Input

```
4
3 9 12 15
```

Output

```
6
```

Exemplo 4

Input

```
3
-15 -10  0
```

Output

```
-5
```

Exemplo 5

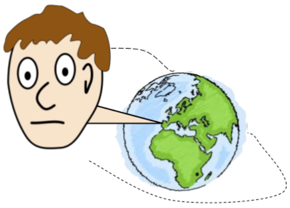
Input

```
5
-5 3 7 11 15
```

Output

```
-1
```

De ficar com a cabeça em órbita!



Em 1937, o matemático alemão Lothar Collatz definiu uma função sobre os número naturais, aparentemente muito simples:

$$C(x) = \begin{cases} 3x+1 & \text{se } x \text{ é ímpar} \\ x/2 & \text{se } x \text{ é par} \end{cases}$$

Portanto, os valores pares são divididos por 2, os ímpares são multiplicados por 3 e é-lhes somado 1. Como o resultado da função é também um natural, a função pode ser aplicada sucessivamente. Por exemplo:

$$C(3) = 10 \quad C(10) = 5 \quad C(5) = 16 \quad C(16) = 8 \quad C(8) = 4 \quad C(4) = 2 \quad C(2) = 1$$

Depois de chegar a 1 a sequência entra em ciclo: $C(1) = 4; C(4) = 2; C(2) = 1$. A sequência resultante da sucessiva aplicação da função C ao número x até entrar em ciclo chama-se a *órbita* de x. Mas, será que isto acontece sempre? A sequência iniciada em qualquer número natural acaba por chegar a 1 e fica nesse ciclo? Ou regressa ao próprio número, entrando também em ciclo mas sem atingir 1? Ou será que há sequências em que os números crescem indefinidamente?

A conjetura de Collatz é que a órbita de qualquer número inclui o 1. No entanto esta conjetura nunca foi provada. Foram geradas muitas órbitas para tentar descobrir padrões, por vezes com auxílio de computadores, mas sem grande sucesso. Claro que um computador pode testar muitos valores e a conjetura já foi comprovada para inteiros até 5×2^{60}. No entanto, outras conjeturas foram refutadas apenas com valores excecionalmente altos, como as conjeturas de Pólya ou de Mertens. Por isso, só se poderá ter a certeza com uma demonstração.

Paul Erdős, um dos matemáticos mais profícuos do século XX, afirmou que a matemática ainda não está preparada para estes problemas! A opinião generalizada entre os matemáticos hoje em dia é que resolver esta conjetura será um grande avanço para a matemática, porque as técnicas necessárias para a sua demonstração levarão certamente a novas e importantes descobertas. Mas também há quem pergunte: será este um dos teoremas indemonstráveis previstos pelos teoremas da incompletude de Gödel?

Tarefa

Escreva um programa que, dado o inteiro n, produza a sua órbita, isto é, a sequência de valores obtida por sucessivas aplicações da função C até atingir 1.

Input

Uma linha com um único valor inteiro positivo n.

Restrições

$1 \leq n \leq 100\,000$ Número a analisar

Output

Uma sequência de linhas com um único inteiro por linha. A primeira linha tem o inteiro lido. Cada uma das linhas seguintes tem o resultado da aplicação da função C ao inteiro da linha anterior. A sequência termina com o número 1.

Exemplo 1

Input

```
3
```

Output

```
3
10
5
16
8
4
2
1
```

Exemplo 2

Input

```
7
```

Output

```
7
22
11
34
17
52
26
13
40
20
10
5
16
8
4
2
1
```

Toques na bola

O Afonso anda a treinar para o campeonato europeu de Toques na Bola. Para analisar o seu desempenho, o seu treinador tem registado os resultados dos treinos. Um *resultado* é o número de toques que o Afonso consegue dar na bola sem que esta caia no chão. O objetivo é verificar se o Afonso tem obtido resultados elevados consistentemente.

Tarefa

Escreva um programa que, dada a sequência de resultados de um treino, apresente o Top 3: os três melhores resultados e quantas vezes cada um deles foi obtido ao longo do treino. No caso de não existirem três resultados diferentes para formar o Top 3, deverá apresentar todos os resultados diferentes e quantas vezes cada um deles foi obtido durante o treino. Portanto, se não houver Top 3, o programa apresenta o Top 2 (como no Exemplo 4) ou o Top 1 (como no Exemplo 5).

Para além disso, o programa deve indicar o resultado que o Afonso conseguiu obter consecutivamente o maior número de vezes e qual foi esse número de vezes. Caso existam vários resultados obtidos consecutivamente o maior número de vezes, deve ser apresentado o maior desses resultados.

Input

Uma linha com $T + 1$ números inteiros. Os primeiros T números representam a sequência de resultados de um treino. Cada resultado (r) é um número positivo. O último número é -1.

Restrições

$1 \le T \le 150$	Número de resultados do treino
$1 \le r \le 250$	Um resultado do treino

Output

O output tem no máximo 4 linhas. As primeiras linhas correspondem ao Top 3, ou ao Top 2 se só existirem dois resultados diferentes, ou ao Top 1 se os resultados forem todos iguais. Cada linha tem um resultado e quantas vezes esse resultado foi obtido durante o treino. As linhas aparecem ordenadas decrescentemente por resultado.

A última linha tem dois números: o resultado obtido consecutivamente o maior número de vezes e esse número de vezes. Caso existam vários resultados obtidos consecutivamente o maior número de vezes, deve ser apresentado o maior desses resultados.

Exemplo 1

Input

```
12 10 4 9 10 9 14 13 11 9 11 9 9 9 9 12 12 -1
```

Output

```
14 1
13 1
12 3
9 4
```

Exemplo 2

Input

```
13 10 4 9 10 9 14 13 7 9 7 10 10 10 10 7 10 -1
```

Output

```
14 1
13 2
10 7
10 4
```

Exemplo 3

Input

```
5 9 8 7 5 8 7 5 7 5 6 5 7 5 4 4 5 -1
```

Output

```
9 1
8 2
7 4
4 2
```

Exemplo 4

Input

```
8 8 8 9 9 9 -1
```

Output

```
9 3
8 3
9 3
```

Exemplo 5

Input

```
8 -1
```

Output

```
8 1
8 1
```

Programa casamenteiro

Em 2012, o prémio Nobel da Economia foi atribuído a Lloyd Shapley e Alvin Roth pelo seu trabalho sobre atribuições estáveis e desenho de mercados, com aplicações a problemas de colocação de pessoas em postos de trabalho, estudantes em escolas, entre outras. Cremos que o prémio teria sido partilhado por David Gale, se fosse vivo à data. Esses problemas com preferências mútuas estão relacionados com o *Problema dos casamentos estáveis*, no qual se pretende formar n casais **estáveis**, dados n homens e n mulheres e as suas listas de preferências, estritamente ordenadas. Para serem estáveis, não pode haver um par (h, m) tal que m prefere h ao seu marido e h prefere m à sua mulher. Em 1962, Gale e Shapley publicaram um algoritmo para resolver o problema. Segundo uma versão do algoritmo, inicialmente todos os homens e todas as mulheres estão livres. A seguir procede-se à formação de casais. Enquanto há algum homem livre, um qualquer homem livre h propõe-se à primeira mulher m a que ainda não se propôs (seguindo as suas preferências). Se m estiver livre (e tiver algum "fraquinho" por h) ou não estiver livre mas preferir h ao seu par h', então m fica com h, rejeita h' e h' volta a estar livre (podemos procurar logo um par para h'). Se não, m rejeita h e h continua livre em busca de par. Provou-se que o resultado final é a melhor solução estável possível para os homens e a pior para as mulheres. Se se aplicar o algoritmo trocando os papéis dos homens e das mulheres, os casais que se formam correspondem à melhor solução estável para as mulheres e à pior para os homens. O algoritmo pode ser aplicado mesmo que haja pares que não são admissíveis, isto é, pessoas que excluem alguns elementos das suas preferências porque "preferem estar sós a mal acompanhadas". Nesses casos, foi provado que, se uma pessoa puder ficar com par, ficará sempre com par, independentemente do algoritmo usado (embora o par possa depender do algoritmo aplicado).

Tarefa

Escreva um programa que resolva uma instância aplicando o método descrito e indique se a solução que os homens preferem é igual à solução que as mulheres preferem. Adicionalmente, o programa descreverá o estado final de um homem h dado. Assumiremos que o número de homens e de mulheres é igual a n, mas que podem existir pares não admissíveis.

Input

A primeira linha tem o valor de n (os homens e as mulheres são numerados de 1 a n). Seguem-se $2n$ linhas com a indicação das preferências dos homens (do homem 1 até ao n): a primeira linha do par i tem o número de mulheres indicadas por i e a segunda tem a lista de preferências de i, por ordem descrescente de preferência. As $2n$ linhas seguintes têm a descrição das preferências das mulheres, de forma idêntica. Cada pessoa indicou pelo menos uma outra do sexo oposto. A última linha do input tem um inteiro que identifica o homem h para o qual se pretende saber o estado final.

Restrições

$2 \leq n \leq 100$ Número de pessoas de cada grupo

Output

Se a melhor solução estável para as mulheres coincidir com a melhor solução estável para os homens, terá a frase "`Nao podia ser melhor!`" na primeira linha. Caso contrário, terá "`Podia ser melhor!`" (sem aspas). A seguir tem uma linha que descreve o estado do homem h e que terá uma das frases seguintes, com `h`, `x`, e `y` substituídos pelos valores corretos.

```
O homem h fica sozinho.
O homem h fica com x em ambos os casos.
O homem h ficaria com x ou y mas prefere x.
```

Exemplo 1

Input

```
2
2
2 1
2
1 2
2
1 2
2
2 1
1
```

Output

```
Podia ser melhor!
O homem 1 ficaria com 2 ou 1 mas prefere 2.
```

Exemplo 2

Input

```
4
2
1 2
3
1 3 4
3
1 3 4
1
3
3
1 3 2
1
```

```
3
3
4 2 1
4
2 3 4 1
3
```

Output

```
Nao podia ser melhor!
O homem 3 fica sozinho.
```

Exemplo 3

Input

```
5
5
3 4 5 1 2
4
2 5 1 4
5
4 1 2 5 3
5
1 5 3 2 4
5
4 2 3 1 5
5
3 2 5 1 4
4
2 1 3 4
5
4 2 5 1 3
4
2 4 3 1
5
1 2 4 5 3
5
```

Output

```
Podia ser melhor!
O homem 5 ficaria com 3 ou 1 mas prefere 3.
```

Exemplo 4

Input

```
5
5
3 4 5 1 2
4
2 5 1 4
5
4 1 2 5 3
5
1 5 3 2 4
5
4 2 3 1 5
5
3 2 5 1 4
4
2 1 3 4
5
4 2 5 1 3
4
2 4 3 1
5
1 2 4 5 3
1
```

Output

```
Podia ser melhor!
O homem 1 fica com 5 em ambos os casos.
```