

ToPAS'23

Torneio de Programação para Alunos do Secundário

Departamento de Ciência de Computadores

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

Conjunto de Problemas



Faculdade de Ciências da Universidade do Porto

12 de maio de 2023

Este conjunto de problemas deverá conter oito (8) problemas e dezoito (18) páginas.
Se faltar algum problema, por favor avise a organização.

Edição realizada em colaboração com:



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



Departamento de Informática, NOVA School of Science and
Technology | FCT NOVA, Universidade NOVA de Lisboa

ToPAS'23

Torneio de Programação para Alunos do Secundário

Departamento de Ciência de Computadores – FCUP
12 de maio de 2023

Conteúdo

Problema A: Febre!!!	3
Problema B: Nove e Meia	5
Problema C: Onda de Calor	7
Problema D: E o vencedor é...	9
Problema E: Miradouros	11
Problema F: 4-em-linha	13
Problema G: O Génio da Lâmpada	15
Problema H: Batota?	17

Febre!!!

De forma a controlar a evolução da temperatura corporal de um doente, quer-se identificar diferentes situações, em particular, quando a temperatura é superior ou igual a 37°C .

Tarefa

Escreva um programa que, dadas duas temperaturas, assinala se a pessoa está com febre (ou seja, se a segunda temperatura é superior ou igual a 37°C). Caso a pessoa esteja com febre, o programa também indica se a segunda temperatura é superior, igual ou inferior à primeira temperatura.



Input

O input tem uma única linha com quatro inteiros: I_1 , D_1 , I_2 e D_2 . Os dois primeiros números especificam a parte inteira (I_1) e a parte decimal (D_1) da primeira temperatura. Os dois últimos números especificam a parte inteira (I_2) e a parte decimal (D_2) da segunda temperatura. Por exemplo, se a linha tiver 37 8 36 0, o valor da primeira temperatura é $37,8^{\circ}\text{C}$ e o valor da segunda temperatura é $36,0^{\circ}\text{C}$.

Restrições

$20 \leq I \leq 50$ Parte inteira de uma temperatura

$0 \leq D \leq 9$ Parte decimal de uma temperatura

Output

O output tem uma única linha, cujo formato depende do valor da segunda temperatura:

- Se o valor da segunda temperatura for inferior a 37°C , a linha tem apenas a palavra **NORMAL** (em maiúsculas);
- Se o valor da segunda temperatura for superior ou igual a 37°C , a linha tem duas palavras, separadas por um espaço. A primeira palavra é **FEBRE**. A segunda palavra é:
 - **SUBIU**, caso a segunda temperatura seja maior que a primeira temperatura;
 - **MANTEVE**, caso as duas temperaturas sejam iguais;
 - **BAIXOU**, caso a segunda temperatura seja menor que a primeira temperatura.

Exemplo 1**Input**

37 1 36 6

Output

NORMAL

Exemplo 2**Input**

38 9 38 9

Output

FEBRE MANTEVE

Exemplo 3**Input**

36 1 37 6

Output

FEBRE SUBIU

Exemplo 4**Input**

37 0 37 0

Output

FEBRE MANTEVE

Exemplo 5**Input**

37 9 37 1

Output

FEBRE BAIXOU

Exemplo 6**Input**

36 1 36 6

Output

NORMAL

Nove e Meia

O Rui faz anos hoje e, como se sente muito crescido, perguntou à mãe se, a partir de agora, já poderia ir para a cama depois das nove e meia. A mãe, que sabe como é importante ele deitar-se cedo, respondeu-lhe (com bastante exagero):

— *Não. Só poderás ir para a cama depois das nove e meia quando tiveres, pelo menos, metade da minha idade.*



Por coincidência, a mãe nasceu no mesmo dia do ano que o Rui e, por isso, também faz anos hoje.

Consegues dizer ao Rui daqui a quantos anos ele poderá deitar-se após as nove e meia, se souberes quantos anos ele e a mãe fazem hoje?

Vejam os dois exemplos:

- Se hoje o Rui fizer 5 anos e a mãe fizer 26 anos, daqui a 16 anos o Rui fará 21 anos e a mãe fará 42 anos. Só nessa altura a idade do Rui será, pelo menos, metade da idade da mãe.
- Se hoje o Rui fizer 6 anos e a mãe fizer 25 anos, daqui a 13 anos o Rui fará 19 anos e a mãe fará 38 anos. Só nessa altura a idade do Rui será, pelo menos, metade da idade da mãe.

Tarefa

Escreva um programa que, dadas as idades atuais do Rui e da mãe, calcula o número de anos que têm de passar para a idade do Rui (nesse momento) ser, pelo menos, metade da idade da mãe (nesse momento).

Input

O input tem duas linhas, cada uma com um número inteiro. A primeira linha tem o número (R) de anos que o Rui faz hoje. A segunda linha tem o número (M) de anos que a mãe faz hoje.

Restrições

$3 \leq R \leq 7$ A idade do Rui (hoje)

$20 \leq M \leq 50$ A idade da mãe (hoje)

Output

O output tem uma única linha, com um inteiro que representa o número de anos que têm de passar para a idade do Rui (nesse momento) ser, pelo menos, metade da idade da mãe (nesse momento).

Exemplo 1

Input

5
26

Output

16

Exemplo 2

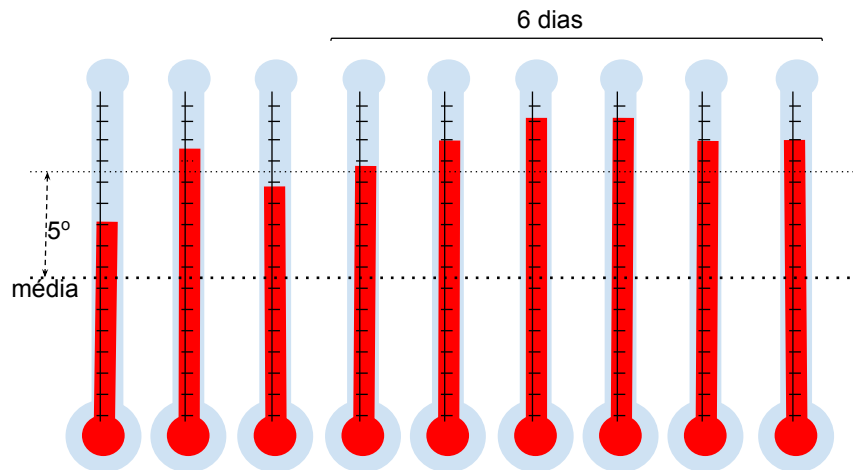
Input

6
25

Output

13

Onda de Calor



Vem aí o tempo quente e receia-se um aumento das ondas de calor devido às alterações climáticas. O IPMA pediu a nossa colaboração na deteção de possíveis ondas de calor, por forma a serem difundidos avisos à população. Uma onda de calor é uma sequência de pelo menos 6 dias consecutivos em que a temperatura máxima é superior em 5° centígrados ao valor médio diário no período de referência.

Tarefa

Escreva um programa que, dadas uma temperatura média de referência e uma sequência de valores máximos de temperatura previstos para os próximos dias, indica se é previsível uma onda de calor.

Input

A linha inicial do input tem um inteiro M , a temperatura média de referência. A linha seguinte tem um inteiro N , o número de dias consecutivos para os quais há previsões. As N linhas seguintes contêm por ordem cronológica crescente os inteiros T_i , os valores previstos da temperatura máxima de cada dia.

Restrições

- $-60 < M < 60$ Temperatura média de referência
- $6 < N < 60$ Número de dias consecutivos para os quais há previsões
- $-60 < T_i < 60$ Valor previsto para a temperatura máxima no dia i

Output

O output é constituído por apenas uma linha com uma única a palavra. Essa palavra é:

- WAVE, no caso de estar prevista alguma onda de calor;
- FLAT, no caso contrário.

Exemplo 1**Input**

19
10
24
23
26
27
24
27
26
29
27
24

Output

FLAT

Exemplo 2**Input**

19
10
24
23
26
27
28
27
26
29
27
24

Output

WAVE

E o vencedor é...

Um grupo de amigos gosta muito de jogos com cartas. No outro dia descobriram um jogo muito engraçado. O jogo usa apenas as cartas numeradas de 1 a 10 (o ás representa o 1) de um baralho usual de cartas. Logo, há 40 cartas em jogo (10 de cada naipe).

Inicialmente, cada jogador recebe 5 cartas e, ao longo do jogo, vai trocando cartas (uma por uma) com outros jogadores. Quando o jogo termina, conta-se a pontuação de cada jogador, a qual depende das 5 cartas que tiver na mão. Ganha quem tiver a maior pontuação, podendo haver vários vencedores.

A pontuação é a soma de várias parcelas:

- por cada par de cartas com o mesmo número n , conta-se $3 \times n$ pontos;
- por cada trio de cartas com o mesmo número n , conta-se $5 \times n$ pontos;
- por cada quadra de cartas com o mesmo número n , conta-se $10 \times n$ pontos;
- o resto das cartas (que aparecem isoladamente) contam apenas o seu número.



A pontuação é obtida pela soma de todos estes valores, tendo em conta que nenhuma carta pode contar para duas parcelas diferentes.

Vejamus um exemplo com três jogadores. Um dos jogadores termina o jogo com 1 ás, 1 três, 1 quatro e 2 cincos; a sua pontuação é 23 ($23 = 1 + 3 + 4 + 3 \times 5$). Outro jogador acaba com 2 ases, 1 quatro, 1 cinco e 1 sete; a sua pontuação é 19 ($19 = 3 \times 1 + 4 + 5 + 7$). No fim do jogo, o terceiro jogador tem 4 dois e 1 três; a sua pontuação é 23 ($23 = 10 \times 2 + 3$). Este jogo teria 2 vencedores, ambos com 23 pontos.

Os amigos gostam do jogo, mas queriam ter um programa que calculasse o resultado final, para se focarem na parte divertida do jogo. Consegues ajudá-los?

Tarefa

Escreva um programa que, dadas as cartas nas mãos dos jogadores, calcula quantos vencedores houve e qual a pontuação dos vencedores.

Input

A primeira linha do input tem um número inteiro, J , que representa o número de jogadores. Cada uma das J linhas seguintes tem cinco inteiros, n_1, n_2, \dots, n_5 , separados por um espaço, que correspondem aos números das cartas na mão de um jogador, no final do jogo. Não há mais de 4 cartas de cada número no conjunto das mãos dos jogadores.

Restrições $2 \leq J \leq 8$ Número de jogadores $1 \leq n_i \leq 10$ Número de uma carta (para $i = 1, 2, \dots, 5$)**Output**

O output tem uma linha com dois inteiros: o primeiro é o número de vencedores do jogo e o segundo é a pontuação dos vencedores.

Exemplo 1**Input**

```
4
1 6 10 6 1
3 3 9 3 2
5 10 3 5 10
4 7 4 4 4
```

Output

```
1 48
```

Exemplo 2**Input**

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

Output

```
2 23
```

Miradouros

Os principais miradouros de Faro, Lisboa e Porto são:

- **Faro:** Arco da Vila, Monte de Faro;
- **Lisboa:** Graça, Senhora do Monte, São Pedro de Alcântara, Santa Luzia, Santa Catarina, Monte Agudo, Penha de França, São Jorge;
- **Porto:** Sé Catedral, Igreja dos Grilos, Torre dos Clérigos, Serra do Pilar, Vitória, Jardins do Palácio de Cristal.



Todos os miradouros são locais muito apreciados pelos turistas, que têm uma aplicação no telemóvel para registar os miradouros que visitaram. Por exemplo, se forem registadas cinco visitas, aos miradouros Graça, Monte de Faro, Vitória, Vitória e Graça, os miradouros mais visitados são (por ordem alfabética) Graça e Vitória, cada um com duas visitas.

Tarefa

Escreva um programa que, dados os nomes dos miradouros visitados pelos turistas, indica quantas visitas foram registadas (no total), quantas visitas foram efetuadas ao(s) miradouro(s) mais visitado(s) e qual ou quais foram os miradouros mais visitados.

Input

O input tem $N + 1$ linhas. Cada uma das N primeiras linhas tem o nome de um miradouro. Como o registo das visitas é gerado por uma aplicação, o nome do miradouro é sempre um dos nomes apresentados na Tabela 1, escrito em maiúsculas e sem acentos nem cedilhas. A última linha do input tem a palavra FIM.

Restrições

$1 \leq N \leq 1\,000$ Número de visitas registadas

Output

A primeira linha tem dois valores inteiros que indicam, respetivamente, o número total de visitas registadas e o número de visitas efetuadas ao(s) miradouro(s) mais visitado(s). Cada uma das restantes linhas tem o nome de um dos miradouros mais visitados e o nome da respetiva cidade, separados por um espaço. Estas linhas ocorrem por ordem alfabética de nome do miradouro. Os nomes dos miradouros devem ser escritos em maiúsculas e sem acentos nem cedilhas. Os nomes das cidades devem ser escritos como habitualmente: a primeira letra é maiúscula e as restantes em minúsculas.

Nome do Miradouro	Cidade
ARCO DA VILA	Faro
GRACA	Lisboa
IGREJA DOS GRILOS	Porto
JARDINS DO PALACIO DE CRISTAL	Porto
MONTE AGUDO	Lisboa
MONTE DE FARO	Faro
PENHA DE FRANCA	Lisboa
SANTA CATARINA	Lisboa
SANTA LUZIA	Lisboa
SAO JORGE	Lisboa
SAO PEDRO DE ALCANTARA	Lisboa
SE CATEDRAL	Porto
SENHORA DO MONTE	Lisboa
SERRA DO PILAR	Porto
TORRE DOS CLERIGOS	Porto
VITORIA	Porto

Tabela 1: Miradouros

Exemplo 1

Input

GRACA
 MONTE DE FARO
 VITORIA
 VITORIA
 GRACA
 FIM

Output

5 2
 GRACA Lisboa
 VITORIA Porto

Exemplo 2

Input

GRACA
 ARCO DA VILA
 PENHA DE FRANCA
 SANTA LUZIA
 VITORIA
 ARCO DA VILA
 PENHA DE FRANCA
 VITORIA
 PENHA DE FRANCA
 PENHA DE FRANCA
 SENHORA DO MONTE
 FIM

Output

11 4
 PENHA DE FRANCA Lisboa

Exemplo 3

Input

GRACA
 ARCO DA VILA
 SANTA LUZIA
 MONTE DE FARO
 VITORIA
 SE CATEDRAL
 ARCO DA VILA
 VITORIA
 GRACA
 FIM

Output

9 2
 ARCO DA VILA Faro
 GRACA Lisboa
 VITORIA Porto

4-em-linha

4-em-linha (em inglês, “Connect 4”) é um jogo para dois jogadores. Cada um escolhe uma cor e, em seguida, revezam-se, colocando fichas dessa cor numa grade de 7 colunas e 6 linhas, mantida na vertical.

O objetivo do jogo é colocar 4 ou mais fichas *em linha*, ou seja, fazer uma linha de 4 ou mais fichas da cor do jogador na vertical, horizontal ou diagonal. Na imagem à direita, o jogador amarelo ganhou o jogo.

O jogo termina assim que algum jogador o vence (há 4 ou mais fichas desse jogador em linha) ou quando a grade está completa (tem 42 fichas).



Tarefa

Escreva um programa que lê uma configuração de um jogo (um estado da grade) e indica se algum jogador venceu e qual.

Input

Para facilitar, identificamos as fichas dos dois jogadores com as letras maiúsculas X e O. O input consiste na descrição da grade: tem 6 linhas, cada uma com 7 caracteres. Cada caráter é X, O ou . (ponto), conforme a respetiva posição da grade tem uma ficha ou está vazia. Pode assumir que a grade corresponde a uma situação de jogo válida. Portanto, se há vencedor, a última ficha colocada é desse jogador e tem de ser a ficha mais alta de uma coluna.

Output

O output tem uma linha com GANHOU X, GANHOU O, JOGANDO ou EMPATE, conforme algum jogador ganhou, o jogo ainda não terminou ou o jogo terminou (a grade está totalmente preenchida) e ninguém ganhou.

Exemplo 1

Input

```
.....
.....
....O..
X...O..
X...O..
X..XOOX
```

Output

GANHOU O

Exemplo 2

Input

```
.....
.....
O..X.O.
X..OXX.
XXOXOX.
XOXOOOX
```

Output

GANHOU X

Exemplo 3**Input**

.....
.....
.....
X.XX...
OXOO...
XOXO...

Output

JOGANDO

Exemplo 4**Input**

XOXOXXO
OOXXOOO
XXOXOXX
OXXXOOX
OXOOXXO
OOOXXOX

Output

EMPATE

O Génio da Lâmpada

Génio da Lâmpada: 541?

Xico Marado: X-X!

Falhaste! Mas o 4 faz parte do número.

Génio da Lâmpada: 374?

Xico Marado: -7-!

Falhaste outra vez! O 7 faz parte do número e está na posição correta. O 3 e o 4 também fazem parte; só que as posições não estão bem...

Última tentativa.

Como chegamos aqui? É uma longa longa história; mas, como estão com pressa, vamos simplificar. Temos um Génio da Lâmpada e temos o Xico. Sabem qual foi o terceiro desejo do Xico? Imaginem! Virar, ele próprio, Génio da Lâmpada!!!

— *Não, não e não! Não posso aceitar!* E o Xico, que gosta de desafios contrapôs:

— *Para eu mudar de ideias tens três tentativas para acertar.*

— *Acertar em quê?*, questionou o Génio. E o Xico ripostou:

— *Acertar no número em que estou a pensar.*

Será que o Génio é capaz?



shutterstock.com - 2279933451

Tarefa

Escreva um programa que, dada uma sequência constituída pelas tentativas de acerto do Génio da Lâmpada no número pensado pelo Xico e pelo *feedback* do Xico a essas tentativas, indique, **tendo em consideração o *feedback***, se se pode determinar o número pensado pelo Xico ou se não há certezas sobre qual esse número seja. O número é uma sequência de D algarismos, que pode começar por 0 (zero), assim como cada tentativa.

Input

A primeira linha do input contém dois inteiros D e T , que representam, respetivamente, o número de dígitos do número N pensado pelo Xico e o número de tentativas dadas ao Génio para adivinhar esse número. Seguem-se T pares de linhas em que a primeira de cada par contém um inteiro com D algarismos, que representa a tentativa do Génio, e a segunda contém o *feedback* do Xico.

O *feedback* é uma sequência com D caracteres. Se a posição P da sequência contiver:

- um dígito, esse dígito é igual ao que está na posição P do número N (logo, um acerto);
- o carácter **X**, então o dígito na posição P da tentativa não ocorre em nenhuma posição do número N ;

- o carácter - (sinal menos), então o dígito na posição P da tentativa ocorre no número N , não se encontrando, no entanto, na posição P de N .

Assim, se a tentativa fosse 04402 e o número pensado pelo Xico fosse 04314, o *feedback* seria 04--X. O dígito 2 não ocorre em nenhuma posição. Para cada um dos dígitos 0 e 4, assinalou-se um acerto e uma posição incorreta. Mas, note que 4 ocorre duas vezes no número e 0 ocorre uma vez apenas.

Restrições

$1 \leq D \leq 5$ Número de dígitos de N
 $1 \leq T \leq 10$ Número de tentativas de acerto

Output

O output tem uma única linha com:

- RESPOSTA N , em que N é o número pensado pelo Xico, se for possível determiná-lo;
- AINDA NAO SEI, se não for possível determinar o número pensado pelo Xico.

Exemplo 1	Exemplo 2	Exemplo 3	Exemplo 4
Input	Input	Input	Input
3 2	3 2	5 4	5 5
541	541	84293	04402
X-X	X-X	X4XX-	04--X
374	498	34761	89567
-7-	4XX	-4XX-	XXXXX
		54134	04301
Output	Output	X4--4	043--
RESPOSTA 473	AINDA NAO SEI	14304	04310
		-43-4	0431-
		Output	04313
		RESPOSTA 04314	0431-
			Output
			RESPOSTA 04314

Batota?

Por vezes necessitamos de simular acontecimentos aleatórios no computador, por exemplo, para baralhar cartas. Em 1946, John von Neumann propôs o *middle-square method* para gerar sequências de números pseudo-aleatórios no intervalo $[0,1[$.



Parte-se de uma *semente*, que é um número inteiro positivo com $2n$ dígitos. Calcula-se o quadrado desse número; se tiver menos de $4n$ dígitos, acrescentam-se zeros à esquerda. Os $2n$ dígitos centrais definem o pseudo-aleatório resultante e a próxima semente.

Por exemplo, partindo de 1111, calculamos o seu quadrado, que é 1234321, e acrescentamos um zero para ter um número com $4n$ dígitos. Fica **01234321** e o pseudo-aleatório é **0,2343**. Depois, partindo de 2343, calculamos o seu quadrado, que é 5489649, e temos de acrescentar um zero. Fica **05489649** e obtém-se **0,4896**. Os pseudo-aleatórios seguintes seriam **0,9708** e **0,2452**, como se ilustra abaixo.

Como o maior inteiro de $2n$ dígitos é $10^{2n} - 1$, o seu quadrado é inferior a 10^{4n} e, por isso, $4n$ dígitos bastam para o representar, sendo, por vezes, todos necessários. Todas as sequências acabam por ter repetições, algumas muito rapidamente, como as que começam com 1001, 0020 ou 9800.

$$\begin{array}{r} \\ \\ \\ \\ + \\ \hline 0 \ 1 \ 2 \ 3 \ 4 \ 3 \ 2 \ 1 \end{array}$$

$$\begin{array}{r} \\ \\ \\ \\ + \\ \hline 0 \ 5 \ 4 \ 8 \ 9 \ 6 \ 4 \ 9 \end{array}$$

$$\begin{array}{r} \\ \\ \\ \\ + \\ \hline 2 \ 3 \ 9 \ 7 \ 0 \ 8 \ 1 \ 6 \end{array}$$

$$\begin{array}{r} \\ \\ \\ \\ + \\ \hline 9 \ 4 \ 2 \ 4 \ 5 \ 2 \ 6 \ 4 \end{array}$$

semente	quadrado	pseudo-aleatório
1001	01002001	0,0020
0020	00000400	0,0004
0004	00000016	0,0000
0000	00000000	0,0000

Tarefa

Escreva um programa que, dados m , n e uma semente x com $2n$ dígitos, imprima a sequência dos m primeiros pseudo-aleatórios com $2n$ dígitos gerados por este método, partindo de x .

Input

A primeira linha tem dois inteiros positivos m e n , separados por um espaço. O primeiro é o número de pseudo-aleatórios a gerar. Segue-se uma linha com $2n$ inteiros (cada um entre 0 e

9), separados por um espaço, que são os dígitos da semente inicial x , lidos da esquerda para a direita.

Restrições

$1 \leq m \leq 20$ Número de pseudo-aleatórios pedidos

$2 \leq 2n \leq 100$ Número de dígitos na semente

Output

O output tem m linhas, uma por cada pseudo-aleatório gerado. Apresenta-os pela ordem em que foram calculados, sem espaços entre os dígitos e sempre com o prefixo “0,” (sem aspas).

Exemplo 1**Input**

6 2
1 1 1 1

Output

0,2343
0,4896
0,9708
0,2452
0,0123
0,0151

Exemplo 2**Input**

4 2
0 0 2 0

Output

0,0004
0,0000
0,0000
0,0000

Exemplo 3**Input**

5 1
5 2

Output

0,70
0,90
0,10
0,10
0,10

Exemplo 4**Input**

7 2
9 8 0 0

Output

0,0400
0,1600
0,5600
0,3600
0,9600
0,1600
0,5600

Exemplo 5**Input**

3 18
9 8 7 6 8 7 6

Output

0,99999999999997537520000000000000000000
0,000000060638077504000000000000000000
0,976443381110870016000000000000000000