ToPAS'18
Torneio de Programação para Alunos do Secundário

Departamento de Informática

http://eventos.fct.unl.pt/topas-lx

Conjunto de Problemas

Faculdade de Ciências e Tecnologia da NOVA

11 de maio de 2018

Este conjunto de problemas deverá conter sete (7) problemas e dezasseis (16) páginas.
Se faltar algum problema, por favor avise a organização.

Edição realizada em colaboração com:

Departamento de Ciência de Computadores, Faculdade de Ciências, Universidade do Porto

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

ToPAS'18
Torneio de Programação para Alunos do Secundário

Departamento de Informática – FCT NOVA
11 de maio de 2018

Conteúdo

Problema A: Arpejos 3
Problema B: Agarra que é balão 5
Problema C: Compras *online* com cupão 7
Problema D: Campeonato de telepatia 9
Problema E: Jogo dos feijões 11
Problema F: Manda p'ra *offshore* 13
Problema G: Os (co)primos mais afastados 15

Arpejos

João sempre quis ser músico. Recentemente comprou um teclado, a pilhas, com esperança de no futuro poder tocar em pianos de cauda, em grandes salas de concerto.

O teclado tem 37 teclas. Há teclas brancas e teclas pretas, tal como nos pianos de verdade.

Junto com o teclado, vinha um manual para principiantes, que ensina a tocar algumas melodias populares, só com um dedo. Na verdade, é muito simples. Cada tecla é numerada a partir de 1, da esquerda para a direita. Assim, cada melodia é uma sequência de números, que representam as teclas que devem ser tocadas, e as respetivas notas, pela ordem da sequência.

João já ouviu falar em *arpejos*. Um arpejo é uma sequência de três notas que surgem de seguida na melodia e tais que a distância da segunda para a primeira é quatro teclas e da terceira para a segunda é três teclas. A primeira tecla está à esquerda da segunda e a segunda está à esquerda da terceira, por exemplo, <18, 22, 25> e <3, 7, 10>.

Para tocar bem, convém reconhecer os arpejos.[1] Por isso, João quer identificá-los todos, antes de começar a treinar.

Tarefa

A sua missão é escrever um programa que, dada uma lista de números representando a sequência de notas de uma melodia, da forma acima explicada, liste na consola todos os arpejos que existem nessa melodia, pela ordem por que surgem. Note que a terceira nota de um arpejo pode ser a primeira do arpejo seguinte, como na sequência <20, 24, 27, 31, 34>, em que existem dois arpejos: <20, 24, 27> e <27, 31, 34>.

Input

A primeira linha tem um número inteiro, n, que representa o comprimento da melodia, medido em número de notas.

Segue-se uma segunda linha com n números inteiros, os quais representam a sequência de notas da melodia. O valor de cada nota é um número inteiro maior ou igual a 1 e menor ou igual a 37.

Restrições

$3 \leq n \leq 200$ Comprimento da melodia

[1] Nota: em teoria musical, o conceito de arpejo é mais geral do que o usado neste problema.

Output

A primeira linha do output conterá um número, m, representando o número de arpejos existentes na melodia.

Seguem-se m linhas, cada uma com três números representando um arpejo pela ordem por que os arpejos surgem na melodia.

Exemplo 1

Input

```
10
5 7 9 13 16 2 3 7 10 31
```

Output

```
2
9 13 16
3 7 10
```

Exemplo 2

Input

```
12
1 5 9 11 20 23 27 37 27 29 32 35
```

Output

```
0
```

Exemplo 3

Input

```
18
10 14 17 21 24 12 12 10 14 17 33 12 12 18 22 25 29 32
```

Output

```
5
10 14 17
17 21 24
10 14 17
18 22 25
25 29 32
```

Agarra que é balão

O que seria das festas de aniversário das crianças sem o colorido dos balões? Balões de cores diferentes, cheios de hélio, aguardam esperançosos sobreviver aos abraços, aos encontrões e a todas as brincadeiras dos convidados. Uns são mais sacrificados, pela sua cor que chama muito a atenção, e outros têm mais sorte, porque passam mais despercebidos.

É fácil contar balões de uma cor dada. Supomos que cada cor é codificada por três inteiros (cada um entre 0 e 255), como no sistema RGB. Esses inteiros indicam a intensidade do vermelho (*Red*), do verde (*Green*) e do azul (*Blue*) na cor. Por exemplo, $(255, 0, 0)$, $(0, 255, 0)$ e $(0, 0, 255)$ representam, respetivamente, a cor vermelha, a cor verde e a cor azul com intensidade máxima.

Tarefa

Faça um programa que, dada uma cor (r, g, b) e dados n ternos de inteiros que representam cores RGB de balões, indica quantos balões têm a cor (r, g, b).

Input

A primeira linha tem quatro inteiros r, g, b e n, separados por um espaço, que definem a cor (r, g, b) e o número n de balões a analisar. Cada uma das n linhas seguintes tem três inteiros, r', g' e b', que representam a intensidade de vermelho, verde e azul na cor do balão correspondente. Cada balão pode ter uma cor diferente.

Restrições

$0 \leq n \leq 10\,000$	Número total de balões
$0 \leq r, r' \leq 255$	Intensidade de vermelho numa cor
$0 \leq g, g' \leq 255$	Intensidade de verde numa cor
$0 \leq b, b' \leq 255$	Intensidade de azul numa cor

Output

Uma linha com o número de balões da cor (r, g, b) indicada.

Exemplo 1

Input

```
160 100 200 14
0 255 0
90 0 255
70 16 35
255 50 70
0 255 0
255 0 0
160 100 200
255 0 0
0 255 255
0 255 0
0 0 0
100 150 170
255 255 255
116 255 0
```

Output

```
1
```

Exemplo 2

Input

```
255 0 0 6
255 0 0
0 255 0
255 0 0
255 0 0
0 255 0
0 0 255
```

Output

```
3
```

Exemplo 3

Input

```
255 255 255 6
255 0 255
255 255 0
255 0 255
255 0 0
0 255 1
1 0 255
```

Output

```
0
```

Compras *online* com cupão

Nas compras *online* com cupão, o comprador tem um cupão com um dado valor e só pode comprar artigos cujo valor total não exceda o valor do cupão. Nestas transações, o comprador seleciona os artigos que quer comprar. Cada vez que um artigo é selecionado, ele só será colocado no carrinho de compras se o seu valor não exceder o valor restante no cupão. Nesse caso, o seu valor é deduzido ao valor do cupão. Caso contrário, a compra do artigo é rejeitada, mas o comprador poderá continuar a selecionar artigos.

Quando não se deseja comprar mais artigos, dá-se a encomenda por finalizada. Nesse momento, são emitidos dois documentos: **(1)** a nota de encomenda, que tem todos os artigos colocados no carrinho de compras (pela ordem em que foram selecionados) e o valor total desses artigos; **(2)** um novo cupão com o valor restante.

Tarefa

Escreva um programa que, dados o valor do cupão e a sequência dos artigos selecionados (preço e descrição), gera a nota de encomenda e o novo cupão.

Input

A primeira linha contém dois números inteiros: o valor C do cupão (em cêntimos) e o número N de artigos selecionados.

As $2N$ linhas seguintes têm o preço e a descrição de cada artigo selecionado. Para cada artigo, há uma primeira linha com o seu preço P (em cêntimos) e uma segunda linha com uma cadeia de D caracteres correspondente à sua descrição. Os números P e D são inteiros.

Restrições

$1 \leq C \leq 10\,000$	Valor do cupão (inicial)
$1 \leq N \leq 1\,000$	Número de artigos selecionados
$1 \leq P \leq 5\,000$	Preço de um artigo
$1 \leq D \leq 50$	Número de caracteres da descrição de um artigo

Output

As primeiras linhas têm a descrição dos artigos da nota de encomenda (i.e., dos artigos colocados no carrinho de compras), pela ordem de colocação. A última linha contém dois valores inteiros: o valor total a pagar (pelos artigos colocados no carrinho de compras, em cêntimos) e o valor do novo cupão emitido (em cêntimos).

Exemplo 1

Input

```
1225 5
1400
camisa
245
luvas azuis
799
meias amarelas
1000
camisola
150
gorro vermelho
```

Output

```
luvas azuis
meias amarelas
gorro vermelho
1194 31
```

Exemplo 2

Input

```
1225 5
1200
camisa
1245
luvas azuis
1799
meias amarelas
1000
camisola
25
gorro vermelho
```

Output

```
camisa
gorro vermelho
1225 0
```

Exemplo 3

Input

```
120 2
1400
camisa
125
gorro vermelho com riscas brancas
```

Output

```
0 120
```

Campeonato de telepatia

No Campeonato de Telepatia, cada prova envolve um telepata e um leigo, escolhido aleatoriamente entre a assistência. O leigo pensa numa sequência de números inteiros, com um comprimento pré-acordado e escreve-a. O telepata concentra-se nos pensamentos do leigo e escreve uma sequência de igual comprimento que, idealmente, é exatamente igual à do leigo.

A pontuação do telepata depende dos seus "acertos". Existe um *acerto* quando o mesmo número ocorre na mesma posição em ambas as sequências. Acertos *consecutivos* são acertos que ocorrem em posições consecutivas das sequências. Se as posições imediatamente à esquerda e imediatamente à direita da posição de um acerto não existem ou não são de acerto, esse acerto diz-se *isolado*.

A *pontuação* da prova é calculada da seguinte forma: é atribuído 1 ponto por cada acerto isolado e são atribuídos 3 pontos por cada acerto consecutivo.

Por exemplo, se as sequências escritas pelo leigo e pelo telepata forem, respetivamente:

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

o telepata obtém 11 pontos. São atribuídos 9 pontos aos três primeiros acertos (nos números 3, 5 e 6 que se encontram na terceira, quarta e quinta posições) e 2 pontos aos dois acertos isolados no número 4 (na sétima e na décima terceira posições).

Tarefa

Escreva um programa que, dadas duas sequências de números inteiros, uma escrita pelo leigo e a outra pelo telepata, calcula o número total de acertos e a pontuação do telepata.

Input

A primeira linha tem um número inteiro, C, que é o comprimento das sequências. A segunda e a terceira linhas têm as sequências escritas pelo leigo e pelo telepata, respetivamente. Cada uma dessas linhas é composta por C números inteiros, $n_1\, n_2\, \ldots\, n_C$, separados por um espaço.

Restrições

$2 \leq C \leq 30$ Comprimento das sequências

$1 \leq n_i \leq 9$ Um número escrito pelo leigo ou pelo telepata (para $i = 1, 2, \ldots, C$)

Output

A primeira linha tem o número total de acertos (isolados + consecutivos).
A segunda linha tem a pontuação do telepata.

Exemplo 1

Input

```
13
1 2 3 5 6 7 4 2 3 6 3 3 4
2 1 3 5 6 9 4 1 1 1 7 1 4
```

Output

```
5
11
```

Exemplo 2

Input

```
9
1 2 3 4 5 6 7 8 9
9 1 2 3 4 5 6 7 8
```

Output

```
0
0
```

Exemplo 3

Input

```
13
5 5 5 5 5 5 5 5 5 5 5 5 5
5 1 2 3 4 5 6 7 8 9 1 1 5
```

Output

```
3
3
```

Exemplo 4

Input

```
13
4 4 6 9 1 8 8 2 4 3 5 8 7
4 4 6 9 1 8 8 2 4 3 5 8 7
```

Output

```
13
39
```

Jogo dos feijões

O Jogo dos Feijões é jogado por dois jogadores que retiram, alternadamente, um montinho (completo) de feijões de uma das extremidades de uma sequência de montinhos de feijões. No início, há um número par de montinhos de feijões, cada um com uma quantidade diferente de feijões. O jogo acaba quando é retirado o último montinho de feijões e o vencedor é o jogador que conseguiu acumular mais feijões no decorrer do jogo. Se os dois jogadores tiverem retirado o mesmo número de feijões, há empate.

O Alex e a Bela usam estratégias diferentes. O Alex retira sempre o monte com mais feijões. A estratégia da Bela é mais elaborada: começa por retirar o monte com menos feijões; na sua segunda jogada, escolhe o monte com mais feijões; na sua terceira jogada, volta a retirar o monte com menos feijões; e assim sucessivamente, alternando entre o monte menor e o maior. O Alex é sempre o primeiro a jogar, porque é o mais novo.

Por exemplo, se a sequência inicial tiver seis montes, com **6**, **2**, **3**, **1**, **12** e **5** feijões, por esta ordem, as jogadas do Alex e da Bela são as seguintes:

1. O Alex poderia retirar o monte com **6** feijões ou o monte com **5** feijões. Como ele escolhe sempre o monte maior, retira o que tem **6** feijões.
2. A Bela começa sempre a retirar o monte com menos feijões. Sendo $2 < 5$, retira o monte com **2** feijões.
3. Depois, o Alex retira o monte com **5** feijões (porque $5 > 3$).
4. Na jogada seguinte, a Bela escolhe o monte maior, que tem **12** feijões.
5. Quando já só há dois montes, o Alex retira o que é composto por **3** feijões.
6. Por fim, a Bela retira o único monte que resta, que tem apenas **1** feijão.

Este jogo é ganho pela Bela, que acumula 15 feijões (**2**+**12**+**1**), dado que o Alex só arrecada 14 feijões ($\mathbf{6} + \mathbf{5} + \mathbf{3}$).

Tarefa

Escreva um programa que, dada a sequência inicial de montes de feijões, calcula o número de feijões acumulados por cada jogador, indicando se há empate ou quem é o vencedor.

Input

A primeira linha tem um número inteiro par, m, que é o número de montes de feijões quando o jogo começa. A segunda linha é composta por m números inteiros, $n_1\, n_2\, \ldots\, n_m$, que indicam quantos feijões existem em cada monte da sequência. Nunca há dois montes com a mesma quantidade de feijões.

Restrições

$2 \leq m \leq 100$ Número de montes de feijões

$1 \leq n_i \leq 200$ Número de feijões no monte i (para $i = 1, 2, \ldots, m$)

Output

Uma única linha, cuja forma depende dos números de feijões acumulados pelo Alex e pela Bela no fim do jogo, denotados por A e B, respetivamente. A linha tem a forma:

- `Alex ganha com` A `contra` B, se $A > B$;
- `Bela ganha com` B `contra` A, se $B > A$;
- `Alex e Bela empatam a` A, se $A = B$.

Exemplo 1

Input

```
6
6 2 3 1 12 5
```

Output

```
Bela ganha com 15 contra 14
```

Exemplo 2

Input

```
2
2 5
```

Output

```
Alex ganha com 5 contra 2
```

Exemplo 3

Input

```
8
6 5 16 17 7 18 10 1
```

Output

```
Alex e Bela empatam a 40
```

Exemplo 4

Input

```
10
1 3 6 8 10 12 4 9 5 2
```

Output

```
Alex ganha com 31 contra 29
```

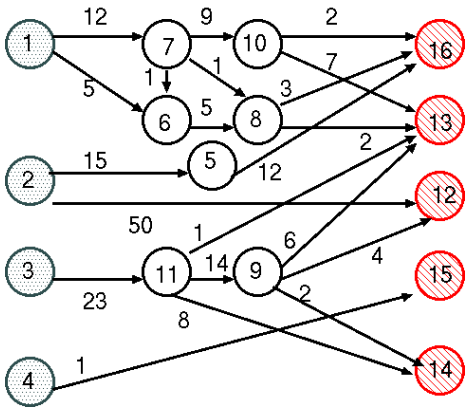
Manda p'ra *offshore*

Além do nosso querido futebol, os temas de abertura dos noticiários são muitas vezes intricados processos, que alguns jornalistas e/ou o Ministério Público tentam deslindar. Alguns envolvem redes de transferências de quantias avultadas para *offshores* em vários destinos, supostamente a *mando de A, ou grupo de A's, para B ou grupo de B's*, com suspeição de favorecimentos, evasões fiscais, ou outras fugas paradisíacas.

A complexidade pode ser tal, que talvez um programa nos pudesse ajudar a verificar o paradeiro e encaminhamento dos montantes...

Tarefa

Escreva um programa que analise uma rede sem circuitos fechados (isto é, sem ciclos) e determine o montante total que chega a um subconjunto de destinos com proveniência num subconjunto de origens. A rede pode ter várias origens e vários destinos. **Todos os montantes que chegam a cada nó intermédio provêm da mesma origem**, como no exemplo. No entanto, aos destinos, podem chegar montantes provenientes de várias origens. Podendo existir cobrança de comissões sobre as transações, o montante que sai de um nó intermédio pode ser inferior ao montante que tal nó recebe.



Input

A primeira linha tem três inteiros N, O e D, que representam o número total de nós da rede, o número de origens e o número de destinos, respetivamente. Segue-se uma tabela com $N-D$ linhas, cada uma com $N-O$ inteiros não negativos. A primeira linha da tabela tem os valores que saem do nó 1. A segunda os que saem do nó 2, e assim sucessivamente. O valor na linha i e coluna j da tabela é o montante enviado do nó i para o nó $O+j$, com $1 \leq i \leq N-D$ e $1 \leq j \leq N-O$. As origens são os nós 1 a O. Os destinos são os nós $N-D+1$ a N. Os restantes nós são os nós intermédios e são identificados por inteiros de $O+1$ a $N-D$.

Finalmente, tem uma linha com dois inteiros s e t que indicam o número de origens e o número de destinos a analisar. Segue-se uma linha com s inteiros que definem tais origens e uma linha com t inteiros que definem esses destinos.

Restrições

$1 \leq N \leq 1000$ Número total de nós da rede
$1 \leq O < N$ Número total de origens
$1 \leq D < N$ Número total de destinos
$1 \leq s \leq O$ Número de origens a analisar
$1 \leq t \leq D$ Número de destinos a analisar

As quantias transferidas são valores inteiros positivos e não superiores a 100.

Output

Um inteiro que indica o montante total recebido nos destinos indicados com proveniência nas origens indicadas.

Exemplo 1

Input

```
16 4 5
0 5 12 0 0 0 0 0 0 0 0 0 0 0 0 0
15 0 0 0 0 0 0 50 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 23 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 12
0 0 0 5 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 1 0 9 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 2 0 0 3 0 0 0
0 0 0 0 0 0 0 4 6 2 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 7 0 0 2 0 0 0
0 0 0 0 14 0 0 0 1 8 0 0 0 0 0 0
2 3
1 4
16 12 15
```

Output

```
6
```

Exemplo 2

Input

```
16 4 5
0 5 12 0 0 0 0 0 0 0 0 0 0 0 0 0
15 0 0 0 0 0 0 50 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 23 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 12
0 0 0 5 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 1 0 9 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 2 0 0 3 0 0 0
0 0 0 0 0 0 0 4 6 2 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 7 0 0 2 0 0 0
0 0 0 0 14 0 0 0 1 8 0 0 0 0 0 0
4 1
1 2 3 4
13
```

Output

```
16
```

Exemplo 3

Input

```
8 3 4
0 0 10 0 0 0 0 0
0 0 0 0 0 0 0 0
5 0 0 7 0 0 0 0
0 0 2 0 2 0 0 0
1 1
2
5
```

Output

```
0
```

Os (co)primos mais afastados

Professor Condor: *Tozé, em Matemática, dizemos que dois números inteiros positivos são coprimos, ou primos entre si, quando o seu maior divisor comum é o número* 1.

Tozé Porquê: *Porquê, professor Condor?*

Professor Condor: *Porquê?! Porque é assim. Por exemplo, os números* 21 *e* 18 *não são primos entre si, pois o seu máximo divisor comum é* 3. *Já os números* 21 *e* 10 *são primos entre si, porque o seu máximo divisor comum é* 1. *Percebeste Tozé?*

Tozé Porquê: *Acho que sim...*

Professor Condor: *Muito bem. Qual é, então, o par de números primos entre si da sequência* 6 9 24 3 19 10 21 14 27*, com as posições mais afastadas?*

Tozé Porquê: *Humm... É o* 6 *e o* 19.

Professor Condor: *Não Tozé! É o* 9 *e o* 14*...*

Tozé Porquê: *Porquê, professor?*

Professor Condor: *Porque entre o* 6 *e o* 19 *só há três números (*9, 24 *e* 3*), enquanto que entre o* 9 *e o* 14 *há cinco números (*24, 3, 19, 10 *e* 21*).*

O Tozé Porquê, apesar de esperto e curioso, é também muito precipitado. Por isso, não acertou na solução para o problema colocado pelo Professor Condor. Podem dar-lhe uma ajuda?

Tarefa

Escreva um programa que, dada uma sequência de números inteiros positivos, todos distintos, determine o par de números primos entre si que se encontram em posições o mais afastadas possível. No caso de haver vários pares com afastamento máximo, deverá ser considerado o par que, na sequência, se encontre mais à esquerda.

Input

A primeira linha tem um número inteiro, N, que representa o tamanho da sequência. A segunda linha é composta por N números inteiros positivos, $S_1\, S_2\, \ldots\, S_N$, separados por um espaço. Os elementos da sequência são todos distintos.

Restrições

$2 \leq N \leq 100$ Tamanho da sequência de números

$2 \leq S_i \leq 100\,000$ Um número da sequência (para $i = 1, 2, \ldots, N$)

Output

Uma linha com o par de números primos entre si da sequência com as posições mais afastadas (caso exista um só par à máxima distância). Caso existam vários pares de números primos entre si com afastamento máximo, terá de ser escrito o par que ocorre primeiro na sequência. Os números do par deverão ser impressos pela ordem por que aparecem na sequência. Caso a sequência não possua nenhum par de números primos entre si, o programa escreverá uma linha com a mensagem "`NAO HA PRIMOS ENTRE SI`" (sem acentos e sem aspas).

Exemplo 1

Input

```
9
6 9 24 3 19 10 21 14 27
```

Output

```
9 14
```

Exemplo 2

Input

```
6
2 4 6 8 10 12
```

Output

```
NAO HA PRIMOS ENTRE SI
```

Exemplo 3

Input

```
3
10 3 5
```

Output

```
10 3
```