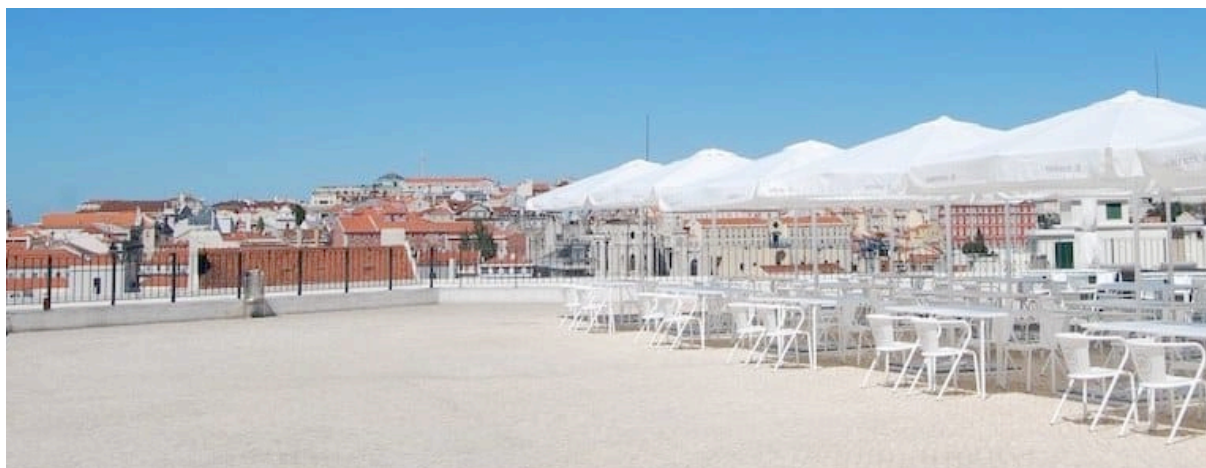


Festa da Programação

23 de abril de 2021

Esplanada



Finalmente, podemos voltar para as esplanadas. Ainda assim, há regras: as mesas têm de estar espaçadas, dispostas retangularmente, não pode haver mais que quatro pessoas em cada mesa e, ao chegar, a gerência obriga-nos a escolher a mesa que nos deixe tão afastados dos restantes convivas quanto possível.

Mais exatamente: quando chegamos à esplanada, é-nos atribuída a mesa tal que a soma das distâncias entre nós e cada uma das pessoas sentadas às outras mesas é máxima.

A distância usada é a distância de Manhattan, calculada pela soma dos valores absolutos das diferenças das coordenadas.

Tarefa

Escrever um programa que, dada a disposição das mesas na esplanada e a atual ocupação das mesas, calcule a mesa que deve ser atribuída ao próximo grupo que chegue.

Submeta no problema A.

Input

O input é obtido da consola. A primeira linha contém dois números que representam o número de “linhas” de mesas, **R**, e o número de “colunas” de mesas, **C**, significando que há **R** \times **C** mesas dispostas “matricialmente”, em linhas e colunas.

Seguem-se linhas em número indeterminado, cada uma com três números, **Y**, **X** e **N**: o par (**Y**, **X**) identifica uma mesa, pelo seu número de linha, **Y**, e pelo seu número de coluna **X**. O número **N** representa o número de pessoas que estão sentadas nessa mesa.

Não há mesas repetidas no input. Apenas as mesas ocupadas são indicadas. Em todos os casos de teste haverá pelo menos uma mesa livre e pelo menos uma mesa ocupada.

Output

O output é uma linha com três números. Os dois primeiros representam as coordenadas, linha e coluna, da mesa selecionada, de acordo com o critério indicado: maximizar a soma das distâncias relativamente a cada uma das pessoas das outras mesas. O terceiro número representa a soma das distâncias de uma pessoa sentada a essa mesa relativamente a cada uma das pessoas sentadas a outra mesa, na esplanada.

Em caso de empate, isto é, se houver mais do que uma mesa que maximize a somas das distâncias, deve ser escolhida aquela cuja linha seja menor e em caso de empate na linha, aquela cuja coluna seja menor.

Exemplo

Input

```
3 4
0 0 3
2 3 5
2 0 4
```

Output

0 3 39

Explicação

A seguinte figura representa a esplanada: há 3 pessoas na mesa (0,0), 4 pessoas na mesa (2,0), e 5 pessoas na mesa (2,3). Note que identificamos as mesas pelo número de linha e pelo número de coluna, e não pelas coordenadas euclidianas, x e y .

	0	1	2	3
0	3			
1				
2	4			5

Se ao chegar uma pessoa, lhe fosse atribuída a mesa (0, 1), a distância total em relação às outras pessoas seria $3*1+4*(1+2)+5*(2+2)=35$. A primeira parcela representa a distância em relação às pessoas na mesa do canto superior esquerdo, a segunda parcela representa a distância em relação às pessoas na mesa do canto inferior esquerdo e a terceira parcela representa a distância em relação às pessoas na mesa do canto inferior direito.

Se a nova pessoa que chegou ficasse na mesa (0, 2), a distância total seria 37; se ficasse na mesa (0, 3), a distância total seria 39, etc. De facto, de entre as mesas vagas, a que maximiza a soma das distâncias é a mesa (0, 3).

Vacina



Estamos na segunda fase de vacinação. Durante a primeira fase, foram vacinados os cidadãos com 80 ou mais anos e ainda alguns mais novos com patologias graves. Agora, na segunda fase, vacinam-se todos os cidadãos com 65 ou mais anos (e menos de 80 anos), os cidadãos com 50 ou mais anos (e menos de 65 anos) com patologias graves, e a ainda os cidadãos com 80 ou mais anos que, por algum motivo, não tenham sido vacinados na primeira fase.

Em cada posto de vacinação, as vacinas são dadas por ordem de idade, dentro de cada *grupo*. Tal como descrito no parágrafo anterior, há três grupos: o dos cidadãos com entre 50 e 64 anos e com patologias, o dos cidadãos com entre 65 e 79 anos e o dos cidadãos com mais de 80 anos. O grupo mais numeroso, nesta fase, será o segundo grupo: o dos cidadãos com entre 65 e 79 anos.

Assim, na segunda fase vacinam-se primeiro os cidadãos do segundo grupo, por ordem decrescente de idade *ajustada*. A idade ajustada é a idade real adicionada de 5 unidades, por cada patologia grave, no máximo de duas. Vacinam-se a seguir os cidadãos do primeiro grupo, por ordem decrescente de idade (real). Vacinam-se finalmente os cidadãos do terceiro grupo, também por ordem decrescente de idade (real). Os cidadãos que não pertençam a nenhum destes grupos não serão vacinados na segunda fase. Em caso de empate, usa-se a ordem alfabética.

Tarefa

Escrever um programa que, dados os nomes, as idades e as patologias graves de um conjunto de pessoas a vacinar, produza a lista ordenada que deve ser usada na segunda fase da vacinação.

Submeta no problema B.

Input

O input é um ficheiro CSV que será lido da consola.

Cada linha do ficheiro CSV tem dois ou mais campos, separados por vírgulas. O primeiro campo é o nome da pessoa, formado por uma ou mais palavras, separadas cada uma da seguinte por um ou mais espaços; o segundo campo é um número que representa a idade da pessoa; seguem-se zero ou mais campos, separados por vírgulas, cada um identificando uma patologia grave.

Este ficheiro contém os registos de todos os cidadãos inscritos num certo posto de vacinação que ainda não foram vacinados. Portanto, pode conter também cidadãos que só serão vacinados em fases posteriores.

Output

O output conterá tantas linhas quantas as pessoas a vacinar, uma linha para cada pessoa, ordenadas pela ordem de vacinação prescrita.

Em cada linha surgirá o nome da pessoa, a idade e o número de patologias, cada valor separado do seguinte por um espaço.

Exemplo

Input

```
carlos pires,55
maria oliveira,75
francisca gomes,68,hipertensao
rute carvalho,39,obesidade
isabel rocha,73
jose antonio silva,78,diabetes
abel manuel fonseca,62
```

```
rui melo,45,obesidade,insuficiencia hepatica
pedro castro,67,insuficiencia renal,obesidade,hipertensao
nicolau sousa,72
joao jose lopes,81,diabetes
rita cunha,58,obesidade,diabetes,hipertensao
```

Output

```
jose antonio silva 78 1
pedro castro 67 3
maria oliveira 75 0
francisca gomes 68 1
isabel rocha 73 0
nicolau sousa 72 0
rita cunha 58 3
joao jose lopes 81 1
```

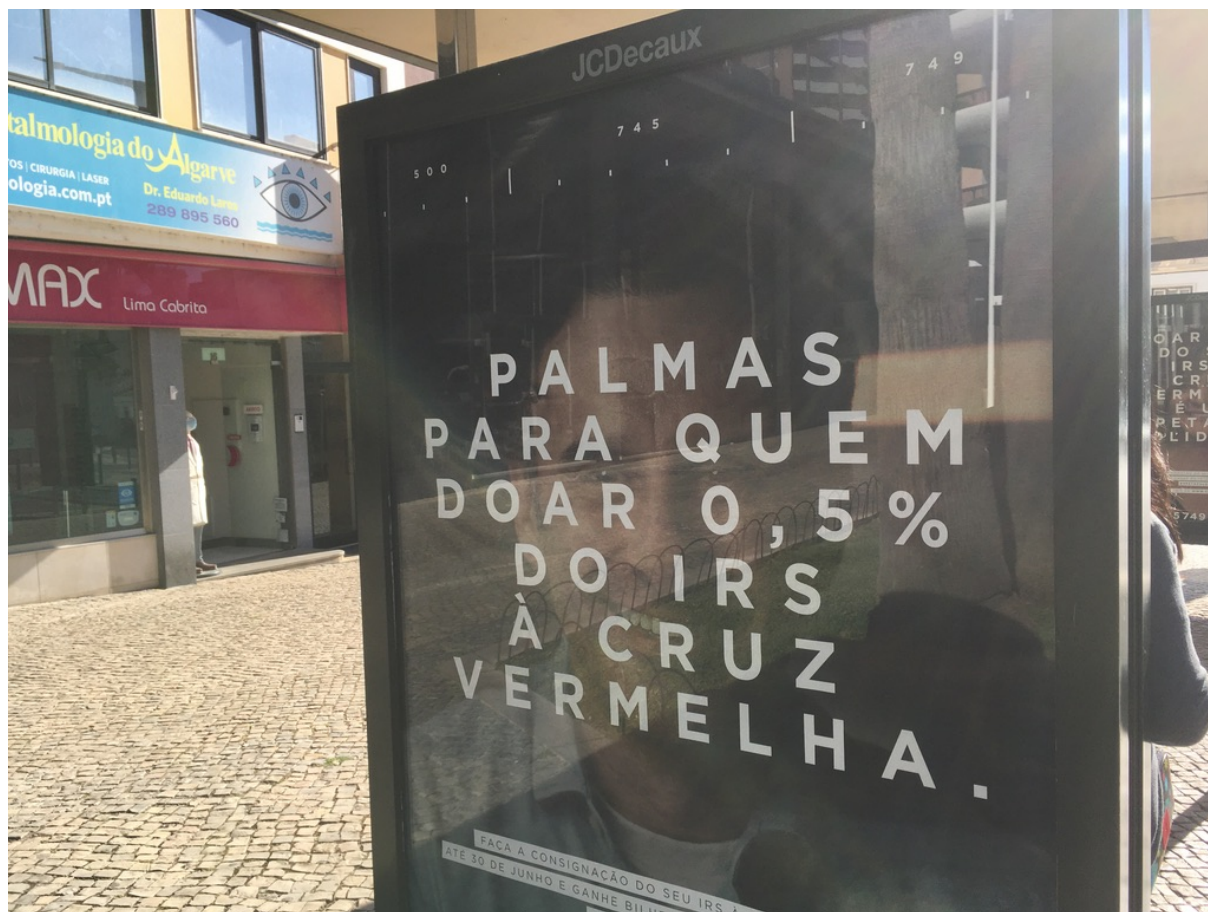
Restrições

As palavras no ficheiro de input usam apenas letras minúsculas sem sinais diacríticos. O ficheiro não tem mais de 1000 linhas. Nenhuma pessoa tem mais de oito patologias.

Requisitos técnicos

A ordenação deve ser realizada por meio da função `qsort`.

Consignação



Estamos na época da entrega do IRS. Nas paragens dos autocarros abundam cartazes de instituições diversas, por exemplo, a Cruz Vermelha e a Amnistia Internacional, a apelar para que lhes doemos 0.5% do nosso imposto, mediante o mecanismo de *consignação*.

Com efeito, ao fazer a declaração do IRS, cada contribuinte pode selecionar uma instituição do seu agrado, consignando-lhe os tais 0.5%. O Estado, automaticamente, retira 0.5% do imposto calculado e entrega-o à instituição selecionada.

A lista de instituições é numerosa (mais de quatro mil) e nem sempre é fácil saber, primeiro, se a instituição que queremos está na lista, depois, estando, qual é o seu “número”, já que é preciso inscrever esse número na declaração.

Felizmente, temos o ficheiro com a listagem das instituições. Faz-nos falta é um programa para consultar interativamente esse ficheiro.

Tarefa

Escrever um programa que, dado um ficheiro com a listagem das instituições autorizadas a receber consignação de IRS, permita a um utilizador consultar essa listagem interativamente.

Cada instituição é descrita por um conjunto de quatro campos: o número de identificação de pessoa coletiva, ou NIPC, o nome, a área de atuação e o distrito a que pertence. O primeiro é um campo numérico e os outros três são campos de texto. A área de atuação e o distrito podem faltar, caso em que serão representados por uma cadeia vazia.

Em cada consulta, o utilizador escreve uma linha na consola, com uma ou mais palavras, separadas por espaços.

O programa responde listando as instituições tais que cada uma das palavras escritas pelo utilizador seja um prefixo de alguma das palavras presentes nos campos de texto dessa instituição.

Submeta no problema C.

Input

O ficheiro das instituições é um ficheiro CSV com quatro campos em cada linha: NIPC, nome da instituição, área de atuação e distrito. O ficheiro deve ser acedido através do seu nome, que será dado na posição `argv[2]` da linha de comando.

Pode consultar o ficheiro na tutoria, mas note que os testes usam outros ficheiros, além desse.

O input é dado pelo utilizador, linha a linha, na consola. Em cada linha o utilizador escreverá uma ou mais palavras, separadas por espaços.

Output

Para cada linha de input o programa responde escrevendo na consola uma linha com o número de instituições selecionadas e, se não forem mais de 16, listando nas linhas seguintes todas as instituições selecionadas.

Uma instituição será selecionada se, para cada palavra escrita pelo utilizador, existir no nome da instituição, na área de atuação ou no distrito, uma palavra que comece pelas letras dessa palavra escrita pelo utilizador.

Ao listar as instituições selecionadas, o programa escreverá em cada linha o NIPC, o nome, a área de atuação e o distrito, separados por vírgulas. As linhas devem vir ordenadas por distrito, depois por área de atuação, depois por nome e finalmente por número.

Requisitos técnicos

A ordenação deve ser realizada por meio da função `qsort`.

Exemplo

Ficheiro CSV

```
504337823,associacao portuguesa apoio a mulher com cancro  
mama,,lisboa  
501637648,assoc solidariedade pilotos aviacao civil  
aspac,,lisboa  
502993731,confraria de nossa senhora da paz,religiao,braga  
501229884,centro paroquial de solidariedade social de  
anca,apoio social,  
500876088,santa casa da misericordia do  
entroncamento,saude,santarem  
501646825,centro paroquial do estoril,religiao,  
502716207,fundacao aurora da ressurreicao coelho borges,apoio  
social,guarda  
508741149,xxs – associacao portuguesa de apoio ao bebe  
prematuro,,lisboa  
503189073,centro de solidariedade social da ademia,apoio  
social,coimbra  
592000834,alianca evangelica portuguesa,religiao,lisboa  
503661821,associacao de defesa do patrimonio historico e  
arqueologico de aljezur,cultura e desporto,faro  
501310789,assoc filarmonica de faro,,faro
```

Sessão interativa

Na sessão interativa transcrita a seguir terá sido usado o ficheiro CSV indicado acima.

```
apoio  
5
```

501229884,centro paroquial de solidariedade social de
 anca,apoio social,
 503189073,centro de solidariedade social da ademia,apoio
 social,coimbra
 502716207,fundacao aurora da ressurreicao coelho borges,apoio
 social,guarda
 504337823,associacao portuguesa apoio a mulher com cancro
 mama,,lisboa
 508741149,xxs – associacao portuguesa de apoio ao bebe
 prematuro,,lisboa
 ass far
 2
 501310789,assoc filarmonica de faro,,faro
 503661821,associacao de defesa do patrimonio historico e
 arqueologico de aljezur,cultura e desporto,faro
 tavira
 0
 portu
 3
 504337823,associacao portuguesa apoio a mulher com cancro
 mama,,lisboa
 508741149,xxs – associacao portuguesa de apoio ao bebe
 prematuro,,lisboa
 592000834,alianca evangelica portuguesa,religiao,lisboa
 c d r
 3
 501646825,centro paroquial do estoril,religiao,
 502993731,confraria de nossa senhora da paz,religiao,braga
 502716207,fundacao aurora da ressurreicao coelho borges,apoio
 social,guarda

O utilizador entrou as seguintes linhas: **apoio**, **ass far**, **tavira**, **portu** e **c d r**.

Neste exemplo, nunca poderia haver mais de 16 instituições selecionadas. Recorde que nesse caso (isto é, no caso de haver mais de 16 instituições selecionadas) as instituições não seriam listadas.

Variante mais simples

No problema D do Mooshak, cada linha do input do utilizador contém uma só palavra, em todos os testes.