

ToPAS'07

Torneio de Programação para Alunos do Secundário

Departamento de Ciência de Computadores

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

Conjunto de Problemas



Faculdade de Ciências da Universidade do Porto

12 de Maio de 2007

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

ToPAS'07

Torneio de Programação para Alunos do Secundário

Dep. Ciência de Computadores – FCUP
12 de Maio de 2007

Contents

Problema A: Minuto Verde	3
Problema B: Febre de Sábado à Tarde	5
Problema C: Bart e a Matemática	7
Problema D: Jogo da Vida	9
Problema E: Não lhes dê Troco	11
Problema F: Pânico na AutoEstrada	13
Problema G: Corte e Costura	15

Problema A

Minuto Verde



Problema

Mais de metade do lixo que produzimos todos os dias pode ser reciclado. Por isso, os cidadãos devem participar activamente na correcta separação dos resíduos para posterior reciclagem.

Papelão – contentor azul (produtos 1 a 10), papel e cartão, jornais, revistas e papel de escrita;
Embalão – contentor amarelo (produtos 24 a 40), embalagens de plástico e metal;
Vidrão – contentor verde (produtos 11 a 23), embalagens de vidro;

Tarefa

Dada uma sequência de produtos para reciclar, identificados por números de 1 a 40, determinar quantos seriam colocados em cada contentor.

Input

Uma sequência de inteiros (com valores de 1 a 40) e que termina por -1. O número de valores de cada um dos tipos é inferior a $2^{30} = 1\text{GByte}$.

Output

Em cada linha escreve uma sequência **cor: numero**, em que **cor** é a cor do contentor e **numero** é o número de produtos colocados nesse contentor. Começa pelo contentor de cor azul, depois amarelo, e finalmente verde.

Exemplo 1

Input

3
13
13
-1

Ouput

azul: 1
amarelo: 0
verde: 2

Exemplo 2

Input

8
3
10
3
1
25
30
3
2
15
23
40
1
3
6
8
14
-1

Ouput

azul: 11
amarelo: 3
verde: 3

Problema B

Febre de Sábado à Tarde



Problema

A vasculhar no sótão encontramos uma verdadeira relíquia: um dos primeiros termómetros digitais do século passado que, além da a temperatura actual, apresenta também as temperaturas máxima e mínima das últimas 24 horas. Infelizmente o termómetro está desregulado e não parece haver uma maneira de o calibrar.

No entanto, como sabemos as temperaturas máxima e mínima registadas no dia anterior, e podemos facilmente saber quais foram as temperaturas máxima e mínima efectivas consultando um jornal, podemos aplicar uma função linear para corrigir os valores do termómetro. Ou seja, para corrigir os valores podemos usar uma função cujo gráfico é uma recta.

A função de correcção transformará a temperatura mínima do termómetro na temperatura mínima real e a temperatura máxima do termómetro na temperatura máxima real. Podemos assumir que as temperaturas mínima e máxima registadas pelo termómetro nunca são de igual valor.

Tarefa

A vossa tarefa é fazer um programa que começa por ler as temperaturas mínima e máxima registadas pelo termómetro, depois as temperaturas mínima e máxima efectivas, seguida por uma sequência de temperaturas terminada por um valor igual ou superior a 100.0. Para cada uma destas temperaturas (excepto as quatro iniciais usadas para calibrar e a que termina a sequência) o programa escreve o valor da temperatura aplicando a correcção linear. Os valores de todas as temperaturas são em virgula flutuante, e os valores devem ser arredondados a uma casa decimal.

Exemplo

Input

```
0.0 30.0
10.0 20.0
0.0
```

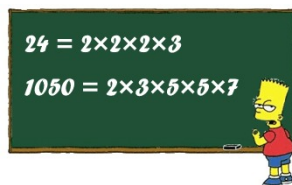
30.0
15.0
10.0
100.0

Ouput

10.0
20.0
15.0
13.3

Problema C

Bart e a Matemática



Problema

O nosso bem conhecido Bart Simpson está com um problema. Adoraria comprar um novo skate, mas a sua mãe, Marge, apenas o fará se tiver boas notas. Mas ele tem grandes problemas a Matemática ...

Bart sabe que na próxima aula será chamado ao quadro para factorizar um número. Lembra que um número primo positivo é um número inteiro positivo maior que 1 que apenas é divisível por 1 e por ele próprio. Factorizar um número é decompô-lo num produto dos seus factores primos. Por exemplo, 24 pode ser escrito como $2 \times 2 \times 2 \times 3$ (2 e 3 são primos) e 1050 pode ser escrito como $2 \times 3 \times 5 \times 5 \times 7$ (2, 3, 5 e 7 são primos).

Bart poderia pedir ajuda à sua irmã Lisa, mas é demasiado orgulhoso para isso. Será que o podes ajudar e mostrar como factorizar um número?

Tarefa

Escreve um programa que dado um número o factoriza, decompondo-o num produto de factores primos.

Input

Na primeira linha tem um número inteiro **C**, que indica o número de casos a tratar ($1 \leq C \leq 20$). Seguem-se exactamente **C** linhas, cada uma contendo um único número inteiro **N**, que representa um número que queremos factorizar ($2 \leq N < 5000$).

Output

É constituído por **C** linhas, cada uma contendo a factorização de um número (pela ordem em que vêm no input, no formato `NUMERO = FACTOR1*FACTOR2*...*FACTORn`. Vê o exemplo de output para perceberes melhor. Os factores primos devem vir sempre por ordem crescente de grandeza

(isto é, deve-se escrever $24 = 2*2*2*3$ e nunca $24 = 3*2*2*2$). Nota também que existe apenas um e só um espaço antes e depois do símbolo de igualdade.

Exemplo

Input

3

5

24

1050

Ouput

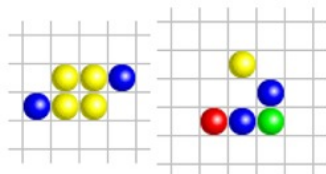
5 = 5

24 = 2*2*2*3

1050 = 2*3*5*5*7

Problema D

Jogo da Vida



Problema

O Jogo da Vida não é um jogo "convencional". Não existem jogadores, nem se pode ganhar ou perder. E uma vez colocadas as peças, regras aparentemente simples determinam tudo o que vai acontecer a seguir, com comportamentos muito complexos, cheios de surpresas! Este jogo é jogado num tabuleiro quadriculado. Cada quadrícula (ou célula) pode estar *morta* ou *viva*. Assinalamos uma célula viva com o caracter '#' e uma célula morta com o caracter '.'. Neste sentido, um tabuleiro inicial válido para o jogo da vida seria por exemplo:

```
.....  
.....  
...#...  
...###...  
...#.#...  
...#...  
.....  
.....
```

Cada célula tem 8 células vizinhas (excepto as células no limite do tabuleiro), que são as quadrículas adjacentes em todas as direcções (incluindo as diagonais). Dada uma configuração inicial, o tabuleiro vai evoluindo com a aplicação de regras. Assim, para cada célula, contamos o número de vizinhos com células vivas e o que acontece a seguir depende desse número:

- Uma célula morta com exactamente 3 vizinhos vivos *nasce*, ficando viva no passo seguinte;
- Uma célula viva com 2 ou 3 vizinhos vivos *sobrevive*, ficando viva no passo seguinte;
- Em todos os outros casos, uma célula *morre* (se estava viva), ou mantém-se morta.

Nota que esta regra é aplicada ao mesmo tempo a todas as células, isto é, para determinarmos o que acontece num determinado passo temos de verificar o que acontece a cada célula antes de realmente fazermos as mudanças que as regras determinam. Por exemplo, para o tabuleiro dado acima, a aplicação das regras originava os seguintes passos (e depois continuaria):

```

.....
.....
...#...
...###...
...#.#...
...#.#...
...#...
.....
.....
->
.....
...###...
...#.#...
...#.#...
...#...
.....
.....
->
.....
...#...
...#.#...
...###...
...#.#...
...#...
.....
.....

```

Tarefa

Escreve um programa que, dada uma configuração inicial do tabuleiro do jogo da vida, obtenha o estado desse mesmo tabuleiro ao fim de um determinado número de passos.

Input

A primeira linha tem três números inteiros **L**, **C** e **P** (separados por espaços), definindo o número de linhas do tabuleiro, o de colunas e o de passos a usar, respectivamente ($1 \leq \mathbf{L}, \mathbf{C} \leq 100$ e que $1 \leq \mathbf{P} \leq 250$). Seguem-se exactamente **L** linhas, cada uma contendo exactamente **C** caracteres, indicando o estado inicial do tabuleiro ('#' para célula viva e '.' para morta).

Output

O estado final do tabuleiro.

Exemplo

Input

```

8 9 3
.....
.....
...#...
...###...
...#.#...
...#.#...
...#...
.....
.....

```

Ouput

```

.....
...#...
..###...
..###...
..###...
...#...
.....
.....

```

Problema E

Não lhes dê Troco



Problema

Uma *slot-machine* retém frequentemente o troco ou parte dele, deixando os utilizadores descontentes. Tal acontece porque a máquina só aceita moedas de certos valores (de 2 euros, 1 euro, e de 50, 20, 10 e 5 cêntimos) e, por vezes, após algumas transacções, certas moedas ficam temporariamente esgotadas. Acontece também porque, para perfazer uma quantia, a máquina começa sempre por usar as moedas de valor mais alto que tiver. Este modo de operar não é muito correcto. Por exemplo, se num certo instante, a máquina dispuser apenas de moedas de 50 e 20 cêntimos, não conseguiria devolver 60 cêntimos (reteria 10 cêntimos). Utilizadores desprevenidos, que insiram quantias muito superiores às necessárias, podem ficar perplexos.

Tarefa

Escrever um programa que simule o comportamento da máquina quando é efectuada uma sequência de transacções, a partir dum certo estado dado. O programa deverá determinar a quantia que a máquina reteve e a fracção das transacções com retenção de troco.

Input

Na primeira linha tem o número de moedas de cada tipo disponíveis inicialmente, segundo a ordem dada acima. Segue-se a descrição duma sequência de transacções a processar, terminada por **0 0**. Haverá pelo menos uma transacção. A primeira linha de cada transacção é constituída por um par (E, C) de inteiros não negativos que representam o montante a pagar (E euros e C cêntimos), seguindo-se a sequência de moedas introduzidas pelo cliente (a qual termina por **0**).

Output

Escreverá a quantia total retida e na linha seguinte a fracção de transacções com retenção, terminando por mudança de linha. Essa fracção terá a forma N/T , em que N pode ser zero (T designa o número total de transacções).

Exemplo

Input

```
0 0 4 0 0 0
0 25
2
20
0
3 10
1
20
1
50
20
20
0
0 40
1
0
1 25
20
50
50
5
0
0 45
20
10
5
5
5
0
0 40
2
0
0 0
```

Ouput

```
0 35
2/6
```

Problema F

Pânico na AutoEstrada



Problema

Uma empresa de gestão de autoestradas instalou um sistema de sensores para contar o número de veículos ligeiros e pesados que atravessam cada via das portagens. O sistema é constituído por dois sensores autónomos, um activado por veículos ligeiros e outro por pesados. Por princípio, um veículo só deveria activar um dos tipos de sensores, ligeiro ou pesado. Devido a um erro de colocação, os resultados de cada sensor não são fidedignos se veículos do mesmo tipo passarem com um intervalo inferior a 10 segundos. Portanto, sempre que o sensor de pesados detectar um qualquer veículo, os seus resultados não são fidedignos até terem decorrido 10 segundos sem nada registar. O mesmo acontece com o sensor de ligeiros.

Felizmente existem registos dos sensores com o momento em que foram activados por cada tipo de veículo e podemos assumir que se ocorrem no mesmo momento pertencem ao mesmo veículo. Se um mesmo veículo for assinalado simultaneamente como ligeiro e pesado e tiver passado um pesado nos últimos 10 segundos então podemos concluir que é um ligeiro. Na mesma situação, se tiver passado um ligeiro nos últimos 10 segundos então podemos concluir que é um pesado. No entanto, se não houver sobreposição de registos no mesmo momento e se os dois registos de veículo do mesmo tipo diferem por menos de 10 segundos, então o segundo registo é indefinido.

Tarefa

Escrever um programa que lê os registos dos sensores e dá como resultado a contagem de ligeiros, pesados e veículos de tipo indefinido, por esta ordem. A primeira linha do *input* indica o total de linhas de registo. Cada um das restantes linhas de input tem no início o número de segundos desde que os sensores foram ligados, seguida de um caracter indicando o tipo de veículo detectado, "L" (Ligeiro) ou "P" (Pesado). Podemos assumir que as linhas de registo estão ordenadas por ordem crescente do número de segundos.

Exemplo

Input

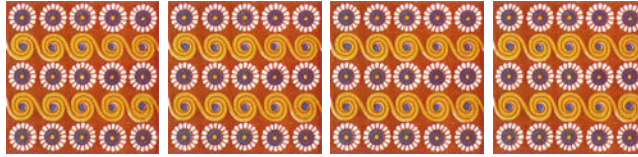
13
100 P
105 P
105 L
200 P
205 L
205 P
300 P
305 L
307 L
307 P
400 P
405 P
500 L

Ouput

4
4
2

Problema G

Corte e Costura



Problema

Pretende-se cortar um pedaço de fita análogo ao da figura seguinte transversalmente de modo que todos os pedaços tenham exactamente o mesmo padrão. O desenho da fita é constituído por *, ? e T. A barra de T's existe sempre pois forma o remate do tecido, mas só ocorre num dos lados, como no exemplo. De resto, não há mais T's no desenho. Os cortes não podem partir os motivos * nem ? ao meio.

```
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
? * ? * * * ? * ? * * * * * ? * ? * * * * ? * ? * * * * ? * ? * * * * ? * ? * * * * ? * ? * * * * ? * ? * * * * ? * ? * * * * ? * ? * * * *
? * ? * * * * * ? * ? * * * * * ? * ? * * * * * ? * ? * * * * * ? * ? * * * * * ? * ? * * * * * ? * ? * * * * * ? * ? * * * * * ? * ? * * * * *
? * ? * * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * *
*****
? * * ? * ? * * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * *
? * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * *
? * ? * * * ? * ? * * * * ? * ? * * * * ? * ? * * * * ? * ? * * * * ? * ? * * * * ? * ? * * * * ? * ? * * * * ? * ? * * * * ? * ? * * * *
* * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * * ? * * *
```

Qual é o **maior número** de pedaços iguais que se pode obter? Como existem várias fitas do mesmo tipo para analisar, umas mais largas do que outras, umas mais compridas do que outras, vamos tentar automatizar o processo.

Tarefa

Escrever um programa que, dada a descrição da fita a cortar, indique o número máximo de pedaços iguais que se pode obter por aplicação de cortes transversais (na direcção da largura, ou seja, perpendiculares à linha de remate).

Input

Na primeira linha dois inteiros **L** e **C** representando a largura da fita e o seu comprimento respectivamente. A largura e o comprimento correspondem ao número de filas de símbolos. Pode assumir que o valor de **L** nunca excede 8 e **C** não excede 10000. Nas linhas seguintes tem a descrição de cada linha da fita (segundo a largura). O caracter T existe apenas no fim de cada linha e não contará para a largura da fita.

Output

Escreverá um inteiro que indica o número de pedaços que obteria (seguido de mudança de linha).

Exemplo

Input

```
7 15
**?****T
**?****T
****?*T
**?****T
**?****T
****?*T
**?****T
**?****T
****?*T
**?****T
**?****T
****?*T
**?****T
**?****T
****?*T
```

Output

```
5
```