PROBLEMAS DE TREINO

Escola Secundária Francisco de Holanda

MARÇO/2018

<u>Nota:</u> No dia 20 de março, a partir das 14.30h poderá comparecer na sala TIC 3 para treinar para o Campeonato de Programação.

NÍVEL I

<u>Problema A</u> – Árvore de Asteriscos

Escreva um programa que coloque no ecrã uma árvore com asteriscos. O número de ramos deverá ser introduzido pelo utilizador.

Exemplos com 3, 4 e 5 ramos:

```
  *
 ***
*****
```

```
   *
  ***
 *****
*******
```

```
    *
   ***
  *****
 *******
*********
```

Altere o programa anterior de forma que, em vez de asteriscos, sejam escritas letras em cada nível, começando o nível inicial com a letra 'A'

```
  A
 B B B
C C C C C
```

```
   A
  B B B
 C C C C C
D D D D D D D
```

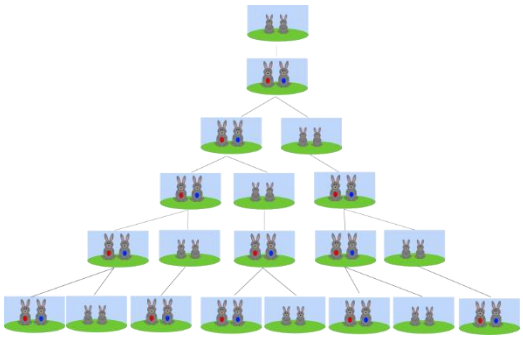
```
    A
   B B B
  C C C C C
 D D D D D D D
E E E E E E E E E
```

<u>Problema B</u> – Divisores de um Número

Dado um número inteiro determinar os seus divisores.

<u>Problema C</u> - Sucessão de Fibonacci

Na matemática, a Sucessão de Fibonacci (também Sequência de Fibonacci), é uma sequência de números inteiros, começando normalmente por 0 e 1, na qual, cada termo subsequente corresponde à soma dos dois anteriores. A sequência recebeu o nome do matemático italiano Leonardo de Pisa, mais conhecido por Fibonacci, que descreveu, no ano de 1202, o crescimento de uma população de coelhos, a partir desta. Esta sequência já era, no entanto, conhecida na antiguidade.



Os números de Fibonacci são, portanto, os números que compõem a seguinte sequência:

0,1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, …

Em termos matemáticos, a sequência é definida recursivamente pela fórmula abaixo, sendo o primeiro termo F1= 1:

$$F_n = F_{n-1} + F_{n-2},$$

e valores iniciais

$$F_1 = 1,\ F_2 = 1.$$

Fonte: https://pt.wikipedia.org/wiki/Sequ%C3%AAncia_de_Fibonacci

Mostre os primeiros números da sucessão de Fibonacci. Crie com e sem recurso a funções.

NÍVEL II

<u>Problema A</u> – Peso dos jogadores de Basquetebol

Crie um programa que armazene os pesos de 5 jogadores de basquetebol e devolva os números dos jogadores cujo peso é superior à média.

<u>Problema B</u> – Decompor um número em fatores primos.

Crie um programa que leia um número inteiro e no final apresente a sua decomposição em fatores primos. Por exemplo: 12=3 x 2 x 2.

<u>Problema C</u> – Estatísticas do Vitória

Crie um programa para auxiliar os técnicos do Vitória na estatística dos seus jogos. O programa deve pedir para cada jornada o número de golos marcados e sofridos. No final de ler as 34 jornadas, deve apresentar para cada jornada se ganhou ('G'), Perdeu ('P') ou empatou ('E') e o resultado. Apresente sempre o resultado, como se os jogos do Vitória fossem todos em casa.

No final deve apresentar também o número de pontos conquistados, o número total de golos marcados e sofridos, a média de golos marcados e o número de vitórias, empates e derrotas.

NÍVEL III

<u>Problema A</u> - Uma Questão de Divisores

A Ana e o Aniceto, dois grandes amigos, gostam muito de tudo o que envolve matemática. Recentemente, estavam a jogar um novo jogo que consistia em descobrir muito rapidamente quantos divisores tem um dado número inteiro positivo. **Um divisor é um número inteiro positivo que divide exatamente outro número inteiro positivo, ou seja, *d* é divisor de *n* se existir um inteiro *q* tal que *d vezes q* é igual a *n*.** Por exemplo, o número 10 tem exatamente 4 divisores (1, 2, 5 e o próprio 10).

Uma vez que são muito rápidos nos cálculos, resolveram complicar um pouco mais o jogo, para o tornar mais desafiante. Em vez de apenas dizerem quantos divisores tem um dado número, têm de dizer quantos números dum dado intervalo têm uma determinada quantidade de divisores. Por exemplo, o jogo pode consistir em dizer rapidamente quantos números entre 1 e 15 têm exatamente 4 divisores. Neste caso a resposta seria 5, pois existem 5 números entre 1 e 15 que têm 4 divisores: 6, 8, 10, 14 e 15.

```
Número:   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
Quantidade de divisores:   1   2   2   3   2   4   2   4   3   4   2   6   2
4   4
```

Depois de selecionarem previamente a quantidade de divisores, a Ana e o Aniceto definem os intervalos que lhes interessam. O jogo consiste em repetidamente um deles ir definindo um novo intervalo de números e o outro ir respondendo quantos números nesse intervalo têm a quantidade pretendida de divisores.

Tens de ajudar a Ana e o Aniceto fazendo um programa para jogar este jogo!

O Problema

Dado um número **D** (a quantidade de divisores), **N** pares de números $\mathbf{A_i}$ e $\mathbf{B_i}$, a tua tarefa é descobrir para cada um destes pares quantos números entre $\mathbf{A_i}$ e $\mathbf{B_i}$ (inclusive, ou seja, no intervalo $\mathbf{[A_i,B_i]}$) têm exatamente **D** divisores.

Input

Na primeira linha do *input* vem um único número inteiro **D** indicando a quantidade de divisores a considerar. Na segunda linha vem um número inteiro **N** indicando o número de intervalos a considerar.

Seguem-se exatamente **N** linhas, cada uma contendo exatamente dois números inteiros separados por um espaço: $\mathbf{A_i}$ e $\mathbf{B_i}$, indicando que deves considerar números entre esses limites. Os limites do intervalo pertencem ao intervalo.

Output

O *output* deve ser constituído exatamente por **N** linhas, cada uma contendo um único número inteiro indicando a quantidade de números no intervalo [$\mathbf{A_i,B_i}$] que têm exatamente **D** divisores. Estas linhas devem vir pela mesma ordem em que os intervalos aparecem no *input*.

Restrições

São garantidos os seguintes limites em todos os casos de teste que irão ser colocados ao programa:

$\mathbf{1 \leq D \leq 1000}$	Número de divisores
1 ≤ N ≤ 1 00	Número de intervalos
$\mathbf{1 \leq A_i \leq B_i \leq 1000}$	Intervalos de números

Exemplo de Input

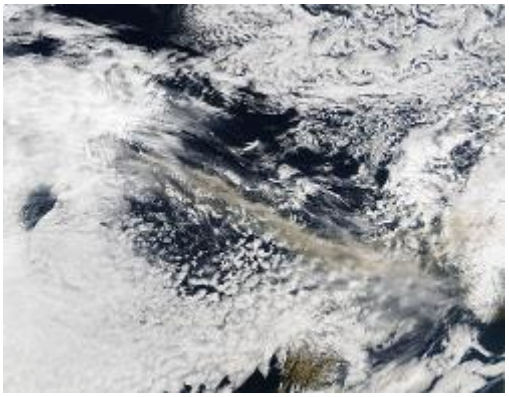
```
4
5
1 10
1 15
11 15
1 5
20 30
```

Exemplo de Output

```
3
5
2
0
4
```

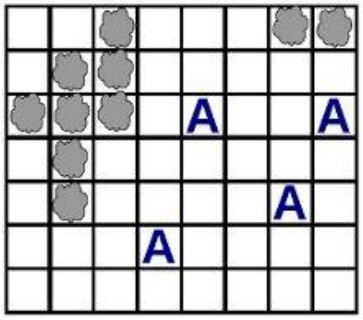
(PROBLEMA RETIRADO DAS OLIMPÍADAS NACIONAIS DE INFORMÁTICA)

<u>Problema B</u> - Nuvem de Cinzas

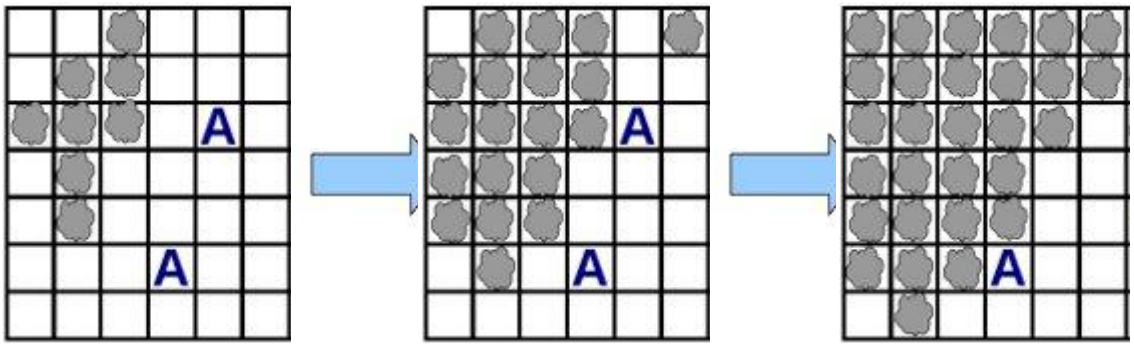


É o caos nos aeroportos! Um vulcão acaba de entrar em erupção provocando uma nuvem de cinzas que se alastra e impede a circulação aérea. O governo da Onilândia está muito preocupado e quer saber quando é que a nuvem de cinzas irá atingir os aeroportos onilandeses.

O governo tem acesso a um mapa obtido via satélite que detalha a situação corrente. O mapa é um retângulo que está dividido em quadrículas mais pequenas. Tendo em conta a situação em análise, apenas são distinguidos três tipos de quadrículas: nuvem (indicando que esse sector do mapa está neste momento coberto por uma nuvem de cinzas), aeroporto (letra 'A', indicando que esse sector do mapa contém um aeroporto) e todas as outras (que não têm neste momento nem uma nuvem nem um aeroporto). Um exemplo de um mapa seria o indicado na figura seguinte:



À medida que o tempo vai passando a situação vai piorando. De facto, por cada dia que passa, a nuvem expande-se uma quadrícula na horizontal e na vertical. Dito de outro modo, ao fim de um dia, todas as quadrículas que estavam adjacentes (vertical ou horizontalmente) a uma quadrícula com nuvem, passam também elas a conter nuvens. Exemplificando a evolução da situação ao fim de dois dias, teríamos o seguinte:



Hoje | Amanhã (1 dia depois) | 2 dias depois

Para preparar convenientemente os planos de contingência, o governo necessita de saber duas coisas: quantos dias demorará até pelo menos um aeroporto ficar coberto pela nuvem e daqui a quantos dias os aeroportos estarão todos eles cobertos pela nuvem. Tens de ajudar!

O Problema

Dado um quadriculado de **L** linhas por **C** colunas indicando a posição atual da nuvem e dos aeroportos, a tua tarefa é descobrir $\mathbf{N_{min}}$, o número de dias até um primeiro aeroporto ficar debaixo da nuvem de cinzas e $\mathbf{N_{max}}$, o número de dias até todos os aeroportos ficarem cobertos pelas cinzas.

Input

Na primeira linha do *input* vem dois números inteiros **L** e **C**, separados por um espaço, indicando prospectivamente o número de linhas e o número de colunas do mapa.

Seguem-se exatamente **L** linhas, cada uma contendo exatamente **C** caracteres, descrevendo o mapa. Cada um dos caracteres pode ser:

- **'#'**, indicando que a quadrícula tem presentemente uma nuvem
- **'A'**, indicando que a quadrícula tem um aeroporto
- **'.'**, indicando que a quadrícula não tem neste momento uma nuvem nem um aeroporto

Existe sempre pelo menos uma quadrícula com nuvem e uma quadrícula com um aeroporto, mas não deves assumir à partida mais nada sobre as outras quadrículas.

Output

O *output* deve ser constituído exatamente por uma única linha contendo dois números inteiros $\mathbf{N_{min}}$ e $\mathbf{N_{max}}$, separados por um único espaço, indicando prospectivamente o número de dias até que um primeiro aeroporto fique coberto pela nuvem e o número de dias até que todos os aeroportos fiquem cobertos.

Restrições

São garantidos os seguintes limites em todos os casos de teste que irão ser colocados ao programa:

$1 \leq L,C \leq 50$ Dimensões do mapa

Exemplo de Input

```
7 8
..#...##
.##.....
###.A..A
.#......
.#....A.
...A....
........
```

Exemplo de Output

```
2 4
```

(PROBLEMA RETIRADO DAS OLIMPÍADAS NACIONAIS DE INFORMÁTICA)

Bom trabalho!