

2021
06/NOV



MARATONA DE PROGRAMAÇÃO

InterFatecs

Fatec
Ribeirão Preto

CP
Centro
Paula Souza

SÃO PAULO
GOVERNO DO ESTADO

CADERNO DE QUESTÕES

10a. EDIÇÃO DA INTERFATECS

ETAPA ÚNICA

APOIO

bk BANK


CANAL **PRO X**
MULTILASER

Lago
Consultoria

 **POPULOS**

WWW.INTERFATECS.COM.BR

1 Instruções

Este caderno contém 11 problemas – identificados por letras de A até K, com páginas numeradas de 3 até 7. Verifique se seu caderno está completo.

Informações gerais

1. Sobre a competição

- (a) A competição possui duração de 5 horas (início as 13:00h término as 18:00h);
- (b) É permitido a consulta a materiais já publicados anteriormente ao dia da competição
- (c) Não é permitido a comunicação com o técnico ou qualquer outra pessoa que não seja a equipe para tirar dúvidas sobre a maratona
- (d) É vedada a comunicação entre as equipes durante a competição, bem como a troca de material de consulta entre elas;
- (e) Cada integrante da equipe poderá utilizar o seu computador/notebook para resolver os problemas, porém, apenas um competidor da equipe ficará responsável pela submissão
- (f) Os problemas têm o mesmo valor na correção.

2. Sobre o arquivo de solução e submissão:

- (a) O arquivo de solução (o programa fonte) deve ter o mesmo nome que o especificado no enunciado (logo após o título do problema);
- (b) confirme se você escolheu a linguagem correta e está com o nome de arquivo correto antes de submeter a sua solução;
- (c) NÃO insira acentos no arquivo-fonte.

3. Sobre a entrada

- (a) A entrada de seu programa deve ser lida da entrada padrão (não use interface gráfica);
- (b) Seu programa será testado em vários casos de teste válidos além daqueles apresentados nos exemplos. Considere que seu programa será executado uma vez para cada caso de teste.

4. Sobre a saída

- (a) A saída do seu programa deve ser escrita na saída padrão;
- (b) Não exiba qualquer outra mensagem além do especificado no enunciado.

Problem A

Abbreviating Names

Source file: abbreviating.{ c | cpp | java | py }

Author: Anderson V. Araujo, Caio Tokunaga e Pedro Flores (UFMS)

Yves is a receptionist at an art gallery and is organizing the opening of a new collection by a famous painter in the city. To save space on the pages and speed up the entry of each guest, he wants to put only an abbreviation of the full names of the guests in the entry list. For each guest, the first and last name must be inserted in full and the others in short form. Help Yves to create a program to accomplish this task.

Input

The entry contains several test cases, each case is a guest's name, containing from 2 to 100 characters, without special characters. The entry ends with EOF.

Output

The program must print the guest list so that it is ordered by the names with the middle names of each guest abbreviated.

Example of Input 1

```
Alda Gusmao Antonia Naves
Ismael Novais Silveira Da Silva De Melo Santos
Prince Uria Noite
Ismael Castelo Gorjao
Jeferson Vargas Capistrano
Layra Bogado
Kyara Ramalho Cavaco
Telmo Lagos Ourique
Ariele Lousa
Nicolae Chaves Fitas
Tania Nascimento Montenegro
Aayush Areosa Sintra
Gastao Saraiva Guilherme
Ariane
Lina Madureira Saloio
```

Example of Output 1

```
Aayush A. Sintra  
Alda G. A. Naves  
Ariane  
Ariele Lousa  
Gastao S. Guilherme  
Ismael C. Gorjao  
Ismael N. S. D. S. D. M. Santos  
Jeferson V. Capistrano  
Kyara R. Cavaco  
Layra Bogado  
Lina M. Saloio  
Nicolae C. Fitas  
Prince U. Noite  
Tania N. Montenegro  
Telmo L. Ourique
```

Problema B

Agrupamentos

Arquivo fonte: agrupamentos.{ c | cpp | java | py }

Autor: Leandro Luque (Fatec Mogi das Cruzes)

Você está trabalhando num projeto que envolve a exibição em um mapa da localização de prestadores de serviço de uma grande empresa. Entre os desafios do projeto está uma lentidão cada vez maior no carregamento do mapa, dado o grande volume de prestadores exibidos - que hoje passa dos 50 mil. Ainda, a visualização do mapa está muito poluída, pois muitos pontos são exibidos em uma área pequena.

Você fez uma PoC (prova de conceito) com uma biblioteca de agrupamento (*clusterização* - que agrupa pontos próximos) do componente de mapas *frontend* que usa e percebeu que o problema de visualização é resolvido. No entanto, agrupar no *frontend* não resolve o problema de ter que carregar milhares de pontos a partir do *backend*. Portanto, você decidiu implementar um algoritmo de clusterização no *backend*. O algoritmo é uma variante do implementado em diversas bibliotecas de mapas disponíveis no mercado.

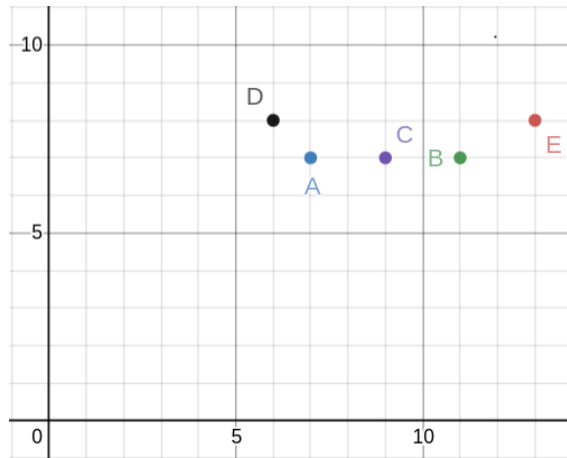
Em resumo, o algoritmo consiste nos seguintes passos:

1. Selecione um ponto (no nosso caso, selecionaremos o primeiro ponto na entrada);
2. Verifique qual é o *cluster* mais próximo:
 - Se não existir *cluster* mais próximo ou se o *cluster* mais próximo estiver a mais de x unidades de distância (considerando a distância euclidiana entre o ponto e o centro do *cluster*), crie um novo *cluster* com o ponto como seu centro;
 - Caso contrário, adicione o ponto ao *cluster* mais próximo e recalcule o centro do *cluster* para ser igual a média aritmética das posições de seus pontos constituintes. Caso um ponto esteja à mesma distância de dois ou mais *clusters*, ele deverá ser adicionado ao *cluster* criado primeiro.
3. Selecione o próximo ponto e repita (2) até passar por todos os pontos.

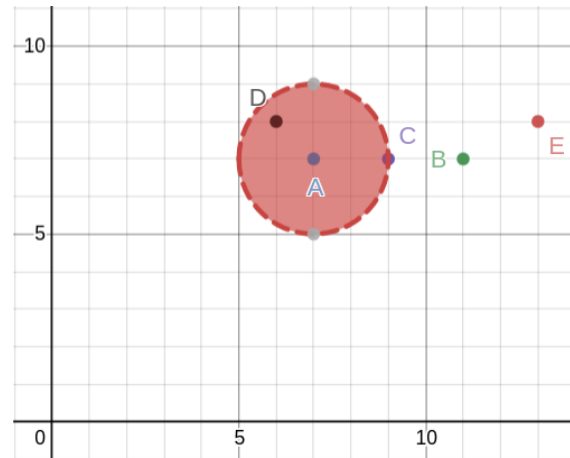
Considere um mapa com os seguintes pontos: A (7, 7), B (11, 7), C (9, 7), D (6, 8), E (13, 8) - Figura 2a; e raio do *cluster* igual a 2.

1. Começaremos pelo primeiro ponto informado: A (7,7);
2. Como ainda não existem *clusters*, será criado um novo com centro em (7, 7) - Figura 2b;
3. O próximo ponto (B) está à quatro unidades de distância do primeiro *cluster* e, portanto, será criado um novo *cluster* com (11, 7) como centro - Figura 2c;
4. O terceiro ponto (C) está a duas unidades de distância do primeiro e do segundo *clusters*. Como as distâncias são iguais, deve-se incluí-lo no primeiro cluster criado (A). Após adicioná-lo, o novo centro do *cluster* deverá ser atualizado para $((x_a + x_c) / 2, (y_a + y_c) / 2) = ((7 + 9) / 2, (7 + 7) / 2) = (8, 7)$ - Figura 2d;
5. O mesmo processo segue até o fim, resultando em 4 clusters - Figura 2e.

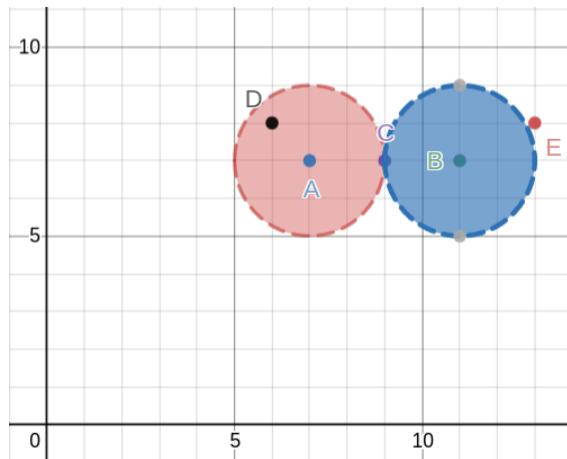
Seu trabalho é implementar o algoritmo explicado.



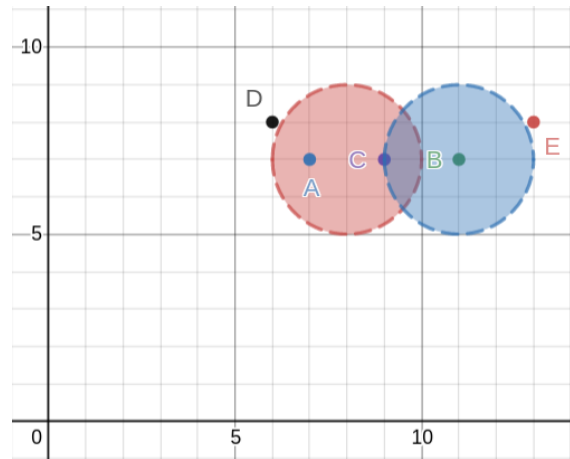
(a)



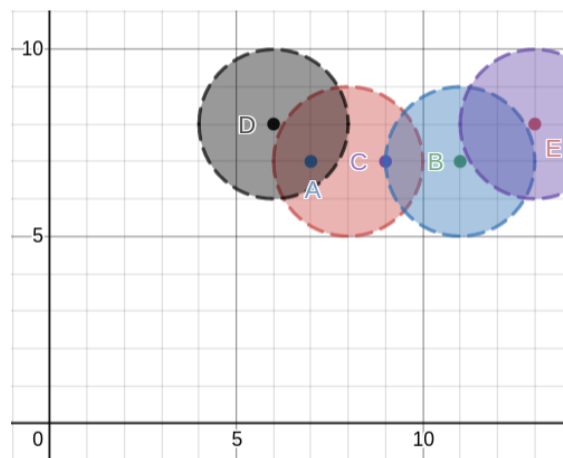
(b)



(c)



(d)



(e)

Figura B.1: Passo a passo do algoritmo

Entrada

A entrada começa com uma linha com dois inteiros: N ($1 \leq N \leq 100$) indicando o número de prestadores de serviço e R ($1 \leq R \leq 100$) representando o raio que será usado para os *clusters*. As próximas N linhas contêm coordenadas cartesianas $X Y$ inteiras ($-1000 \leq X, Y \leq 1000$) representando a localização de um prestador de serviço. A última linha de entrada termina com uma quebra de linha.

Saída

Como saída, imprima uma linha com um número C indicando o número de *clusters* criados pelo algoritmo. Em seguida, imprima C linhas, cada uma com coordenadas cartesianas inteiras (truncando o resultado se necessário) $X Y$ indicando o centro de cada *cluster*, na ordem em que foram criados. A última linha de saída termina com uma quebra de linha.

Exemplo de Entrada 1

```
5 2
7 7
11 7
9 7
6 8
13 8
```

Exemplo de Saída 1

```
4
8 7
11 7
6 8
13 8
```

Exemplo de Entrada 2

```
7 38
-966 -351
-10 899
517 524
-376 -594
-595 -696
362 -943
978 945
```

Exemplo de Saída 2

```
7
-966 -351
-10 899
517 524
-376 -594
-595 -696
362 -943
978 945
```