V - Maratona Doméstica da UDESC (Edileuza)
Departamento de Ciência da Computação
Warmup (Aquecimento) - set/2007

Problema A: Vermelho ou Preto

Uma sala retangular tem o piso coberto de ladrilhos quadrados. Cada ladrilho pode ser da cor preta ou da cor vermelha. Um homem está em pé sobre um ladrilho vermelho. De um ladrilho, ele pode se mover para um dos quatro ladrilhos adjacentes. Entretanto, ele não pode se mover sobre ladrilhos vermelhos (representado pelo caracter '#'). Ou seja, ele pode se mover apenas sobre os ladrilhos pretos (representado pelo caracter '.').
Escreva um programa que conte o número de ladrilhos pretos, os quais podem ser alcançados repetindo os movimentos descritos acima.

Arquivo Entrada:

A entrada consiste de múltiplos conjuntos de dados. Um conjunto de dados inicia com uma linha contendo dois inteiros positivos W (*width*: largura) e H (*height*: altura). W e H são os números de ladrilhos nas direções ***x*** e ***y***, respectivamente. W e H não são mais do que 20. O conjunto de dados contém mais H linhas, cada uma contendo W caracteres. Cada caractere representa a cor do ladrilho como mostrado a seguir.

'.' – um ladrilho preto
'#' – um ladrilho vermelho
'@' – um homem sobre um ladrilho preto (aparece apenas uma vez em cada conjunto de dados)

O fim do arquivo é indicado por uma linha contendo dois zeros.

Arquivo de Saída:

Para cada conjunto de dados o seu programa deverá ter, no arquivo de saída, uma linha contendo o número de ladrilhos que podem ser alcançados a partir do ladrilho inicial(incluindo ele mesmo)

Exemplo de arquivo de entrada:

```
6 9
....#.
.....#
```

```
......
......
......
......
......
#@...#
.#..#.
11 9
.#.........
.#.#######.
.#.#.....#.
.#.#.###.#.
.#.#..@#.#.
.#.#####.#.
.#.......#.
.#########.
...........
11 6
..#..#..#..
..#..#..#..
..#..#..###
..#..#..#@.
..#..#..#..
..#..#..#..
7 7
..#.#..
..#.#..
###.###
...@...
###.###
..#.#..
..#.#..
0 0
```

Exemplo de Arquivo de Saída:

```
45
59
6
13
```

Problema B: Círculo e Pontos (*Circle and Points*)

Lhes são dados N pontos no plano x-y. Você tem um círculo de raio um(01) e o move no plano x-y de modo a incluir o máximo possível de pontos no círculo. Ache o número máximo de pontos que podem ser incluídos no círculo. Um ponto é considerado como sendo incluído no círculo quando está no interior ou sobre o círculo.

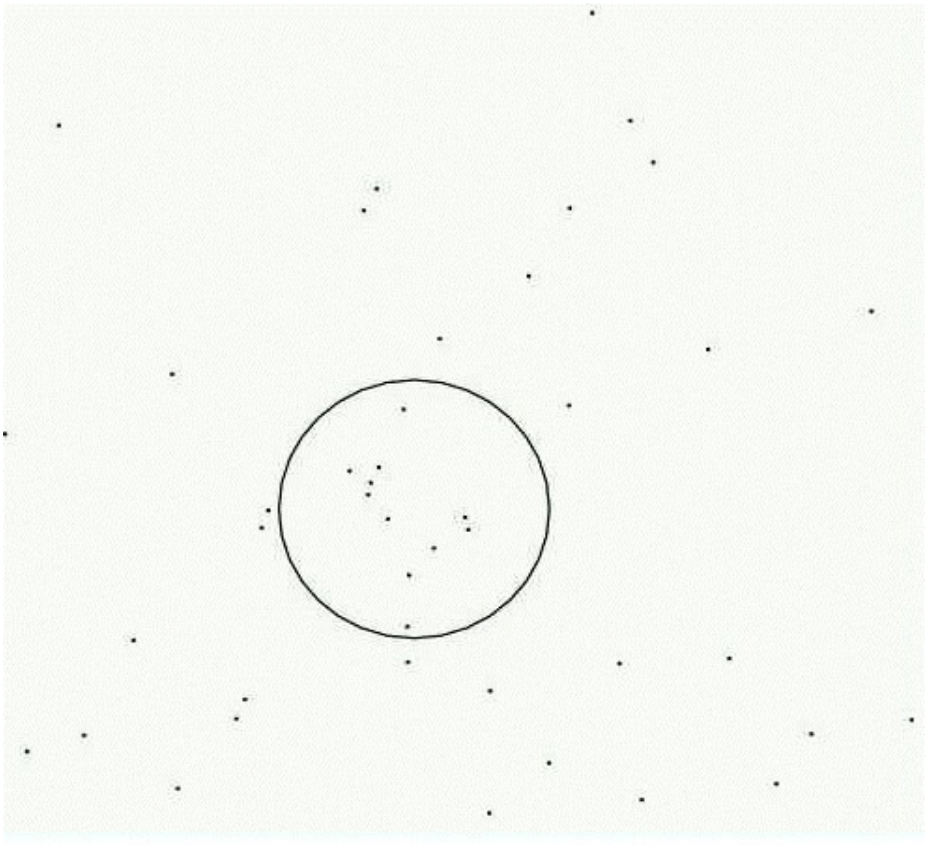

Fig 1. Círculo e Pontos

Arquivo de Entrada:

O arquivo de entrada consiste numa série de conjuntos de dados, seguidos por uma linha que contem apenas o caractere ‘0’, que indica o final da entrada. Cada conjunto de pontos inicia com uma linha que contem um inteiro N, que indica o número de pontos no conjunto de dados. Esta é seguida por N linhas descrevendo as coordenadas dos pontos. Cada uma das N linhas contem dois decimais X e Y, que descrevem respectivamente as coordenadas x e y de um ponto. Estas são dados com 5 dígitos após o ponto decimal.
Você pode assumir que 1<= N <= 300, 0.0 <= X <=10.0 e 0.0 <= Y <= 10.0. Nunca dois pontos estão a distância menor do que 0.0001. Dois pontos num conjunto de dados nunca estão aproximadamente a uma distância de 2.0. Mais precisamente, para qualquer dois pontos num conjunto de dados, a distância ***d*** entre os dois nunca satisfará a 1.9999<=d<=2.0001. Finalmente, num conjunto de dados, nunca três pontos são simultaneamente muito próximos de um único círculo de raio 1. Mais precisamente, sendo P1, P2 e P3 quaisquer pontos num conjunto de dados e; d1, d2 e d3 as respectivas distâncias de cada um até um ponto selecionado arbitrariamente no plano x-y. Então, este ponto nunca poderá ter uma situação tal que: 0.9999 <= di <= 1.0001 (i =1,2,3).

Arquivo de Saída:

Para cada conjunto de dados, imprima n o arquivo uma única linha contendo o número máximo de pontos que podem ser incluídos simultaneamente num círculo de raio um.

Nenhum outro caractere, inclusive espaços em branco antes ou depois, podem ser impressos.

Exemplo do Arquivo de Entrada:

```
3
6.47634 7.69628
5.16828 4.79915
6.69533 6.20378
6
7.15296 4.08328
6.50827 2.69466
5.91219 3.86661
5.29853 4.16097
6.10838 3.46039
6.34060 2.41599
8
7.90650 4.01746
4.10998 4.18354
4.67289 4.01887
6.33885 4.28388
4.98106 3.82728
5.12379 5.16473
7.84664 4.67693
4.02776 3.87990
20
6.65128 5.47490
6.42743 6.26189
6.35864 4.61611
6.59020 4.54228
4.43967 5.70059
4.38226 5.70536
5.50755 6.18163
7.41971 6.13668
6.71936 3.04496
5.61832 4.23857
5.99424 4.29328
5.60961 4.32998
6.82242 5.79683
5.44693 3.82724
6.70906 3.65736
7.89087 5.68000
6.23300 4.59530
5.92401 4.92329
6.24168 3.81389
6.22671 3.62210
0
```

Exemplo de Arquivo de Saída para esta Entrada:

```
2
5
5
11
```

Problema C: Esquerda, Volver!

Este ano o sargento está tendo mais trabalho do que de costume para treinar osrecrutas. Um deles é muito atrapalhado, e de vez em quando faz tudo errado – por exemplo, ao invés de virar à direita quando comandado, vira à esquerda, causando grande confusão no batalhão.

O sargento tem fama de durão e não vai deixar o recruta em paz enquanto este não aprender a executar corretamente os comandos. No sábado à tarde, enquanto todos os outros recrutas estão de folga, ele obrigou o recruta a fazer um treinamento extra. Com o recruta marchando parado no mesmo lugar, o sargento emitiu uma sèrie de comandos "esquerda volver!" e "direita volver!". A cada comando, o recruta deve girar sobre o mesmo ponto e dar um quarto de volta na direção correspondente ao comando. Por exemplo, se o recruta está inicialmente com o rosto voltado para a direção norte, após um comando de "esquerda volver!" ele deve ficar com o rosto voltado para a direção oeste. Se o recruta está inicialmente com o rosto voltado para o leste, após um comando "direita, volver!" ele deve ter o rosto voltado para o sul.

No entanto, durante o treinamento, em que o recruta tinha inicialmente o rosto voltado para o norte, o sargento emitiu uma série tão extensa de comandos, e tão rapidamente, que até ele ficou confuso, e não sabe mais para qual direção o recruta deve ter seu rosto voltado após executar todos os comandos. Você pode ajudar o sargento?

Entrada:
A entrada contém vários casos de teste. A primeira linha de um caso de teste contém um inteiro N que indica o número de comandos emitidos pelo sargento ($1 \leq N \leq 1000$). A segunda linha contém N caracteres, descrevendo a série de comandos emitidos pelo sargento. Cada comando é representado por uma letra: 'E' (para "esquerda, volver!") e 'D' (para "direita, volver!"). O final da entrada é indicado por N = 0.

A entrada deve ser lida da entrada padrão.

Saída:
Para cada caso de teste da entrada seu programa deve produzir uma unica linha da saída, indicando a direção para a qual o recruta deve ter sua face voltada após executar a série de comandos, considerando que no início o recruta tem a face voltada para o norte. A linha deve conter uma letra entre 'N', 'L', 'S' e 'O', representando respectivamente as direção norte, leste, sul e oeste.

A saída deve ser escrita na saída padrão.

Exemplo de entrada:

```
3
DDE
2
EE
0
```

Saída para o exemplo de entrada acima:

```
L
S
```

PS: *este último problema é o mais fácil.*