Caça ao Tesouro

Você e alguns amigos vão participar de um jogo de caça ao tesouro. Nesse jogo, dois times competem para encontrar o maior número de tesouros escondidos em uma sala. Com as suas habilidades de programação você decidiu implementar um programa para determinar a pontuação de cada time e consequentemente qual time foi o vencedor.

O jogo funciona com as seguintes regras. Inicialmente, vários tesouros são espalhados por uma sala e os jogadores são separados em dois times, o time azul e o time vermelho. Em seguida, cada jogador entra na sala e procura por tesouros por um tempo predeterminado. Apenas um jogador por vez pode entrar na sala. Os jogadores dos times devem estar intercalados, ou seja, depois de um jogador do time vermelho deve entrar um jogador do time azul e depois de um jogador do time azul deve entrar um jogador do time vermelho. O primeiro time a entrar na sala é escolhido por sorteio. Após todos os jogadores terem tido a oportunidade de entrar na sala, os tesouros recolhidos pelos jogadores de cada time são somados e vence o time que recolheu mais tesouros.

No seu programa, a sala será representada por uma matriz com uma marcação na posição de cada tesouro e você receberá o caminho realizado por cada jogador. Você pode supor que sempre que um jogador passa por um tesouro esse tesouro será recolhido, ou seja, será computado na pontuação do time desse jogador e não poderá mais ser encontrado pelos próximos jogadores.

Como entrada o seu programa receberá inicialmente um inteiro `n` indicando o tamanho da matriz, seguido de `n` linhas representado a matriz. Cada linha conterá `n` caracteres separados por espaços, se o caractere for `*` a posição correspondente da matriz possui um tesouro. Caso contrário, o caractere será `.` e a posição não possui um tesouro. Em seguida, o seu programa receberá uma linha com uma string indicando o time do primeiro jogador a entrar no labirinto ("vermelho" ou "azul"), seguida por uma linha com um inteiro `M` indicando o número de jogadores totais (soma dos jogadores dos dois times), seguida de `M` linhas com o caminho feito por cada jogador. A entrada da sala fica no canto superior esquerdo da matriz e o caminho é representado por uma sequência de caracteres. Os seguintes caracteres são usados para descrever o caminho:

- `N`: representa uma movimentação para o norte, ou seja para a linha acima da posição atual.
- `S`: representa uma movimentação para o sul, ou seja para a linha abaixo da posição atual.

- `O`: representa uma movimentação para o oeste, ou seja para a coluna à esquerda da posição atual.
- `L`: representa uma movimentação para o leste, ou seja para a coluna à direita da posição atual.

Como saída o seu programa deve imprimir duas linhas com as mensagens `Tesouros encontrados pelo time azul: X` e `Tesouros encontrados pelo time vermelho: Y`, onde `X` é a quantidade de tesouros encontrados pelo time azul e Y é a quantidade de tesouros encontrados pelo time vermelho. Em seguida você deve imprimir `Vitoria do time azul`, caso a pontuação final do time azul seja maior que a do vermelho, `Vitoria do time vermelho`, caso a pontuação final do time vermelho seja maior que a do azul, ou `Empate`, caso as pontuações finais sejam iguais.

Exemplos de entradas e saídas esperadas pelo seu programa:

Teste 01

Entrada

```
5
. . . * .
. * . . .
. . . * .
. * . . .
. . * . .
azul
2
LSSSSL
LLLSSS
```

Saída

```
Tesouros encontrados pelo time azul: 3
Tesouros encontrados pelo time vermelho: 2
Vitoria do time azul
```

No caso desse teste, o primeiro jogador (pertencente ao time azul) se moveu de acordo com o seguinte caminho, indicado com os caracteres `>`, `V` e `+`. Note que ele encontrou 3 tesouros.

```
> V . * .
. V . . .
. V . * .
```

```
. V . . .
. > + . .
```

O segundo jogador (pertencente ao time vermelho), por sua vez, se moveu de acordo com o seguinte caminho. Note que ele encontrou 2 tesouros. Como cada time só tinha um jogador, o time azul foi o vencedor.

```
> > > V .
. . . V .
. . . V .
. . . + .
. . . . .
```

Teste 03

Entrada

```
6
. . * . * .
. . . . . .
. . * . * .
. . . . . .
* . * . * *
. . . . . .
azul
2
LLLLSSOONN
SSSSLLLLL
```

Saída

```
Tesouros encontrados pelo time azul: 4
Tesouros encontrados pelo time vermelho: 4
Empate
```

Teste 04

Entrada

```
6
. . . . . .
* . * . . .
. . . . . .
. . * . * .
* . . . . .
```

```
. . . . . .
vermelho
4
SSSSLLLL
SSLLNNOO
LLLLSSSS
LLSSSSLL
```

Saída

```
Tesouros encontrados pelo time azul: 2
Tesouros encontrados pelo time vermelho: 3
Vitoria do time vermelho
```

Código Base

No arquivo auxiliar lab10.py você irá encontrar um código base para dar início ao processo de elaboração deste laboratório. Para facilitar a implementação do seu programa, o código base realiza a leitura da matriz.

```
n = int(input())
matriz = [input().split() for _ in range(n)]
```

Orientações

- Veja aqui a página de submissão da tarefa.
- O arquivo a ser submetido deve se chamar lab10.py.
- No link "Arquivos auxiliares" há um arquivo compactado (aux10.zip) que contém todos os arquivos de testes abertos (entradas e saídas esperadas).
- O laboratório é composto de 10 testes abertos e 10 testes fechados.
- O limite máximo será de 20 submissões.
- Acesse o sistema SuSy com seu RA (apenas números) e a senha que você utiliza para fazer acesso ao sistema da DAC.
- Você deve seguir as instruções de submissão descritas no enunciado.
- Serão considerados apenas os resultados da última submissão.
- Esta tarefa tem peso 3.
- O prazo final para submissão é dia 13/11/2022 (domingo).