

🏠 Home (/) » SCC0222 (/offerings/view/1601) » [3 - Matriz] A Esteira de Plínio

## [3 - Matriz] A Esteira de Plínio

**Disciplina:** SCC0222 - Laboratório de Introdução à Ciência da Computação I

**Prazo de Entrega:** 23/06/2021 23:59:59 Fechado

### Descrição

Plínio é um robô de manutenção que foi encarregado da tarefa de verificar o correto funcionamento das esteiras de uma fábrica. Seu trabalho é viajar em cima das esteiras verificando se elas estão indo para os devidos lugares. Existem alguns tipos de erros que podem ocorrer na esteira. A esteira pode terminar antes de chegar em seu destino, caso no qual Plínio deve alertar "Falha na esteira.", ou uma esteira pode se conectar em si mesma, criando um loop infinito. Nesse caso, Plínio deverá alertar "Loop infinito.". Caso a esteira chegue em seu devido destino, o robô diz "ok.".

Você foi encarregado de configurar Plínio. Escreva um programa que leia como entrada a disposição das esteiras da fábrica e verifique se essa esteira contém loops infinitos, possui falha, ou está correta. Imprima na saída padrão a mensagem de o robô diz.

A fábrica é representada por uma matriz de 32 x 64 caracteres, ou seja, 32 linhas e 64 colunas e é como uma imagem de cima da fábrica. Cada caractere da matriz representa uma parte de uma esteira:

- '>' - Esteira que vai para a direita.
- '<' - Esteira que vai para a esquerda.
- '^' - Esteira que vai para cima.
- 'v' - Esteira que vai para baixo.
- '#' - Junção.
- '[' - Começo da esteira.
- ']' - Fim da esteira.
- ' ' - Espaço em branco.

Por exemplo, considere a seguinte fábrica

```
[ > > > > > > v
    v
  > > > # > > > ]
  ^      v
  ^      v
  ^ < < <
```

Plínio começa no canto superior esquerdo, no caractere imediatamente à direita de onde o '[' se encontra. Sempre haverá apenas um '[' na esteira e Plínio sempre inicia na esteira a direita desse caractere.

Seguindo a esteira de exemplo, Plínio vai para a direita até que precisa ir para baixo e finalmente se encontra com uma junção ('#'). A junção é "escorregadia" e, como ainda se tem a velocidade adquirida das esteiras anteriores, os produtos da fábrica que passam por essa parte da esteira (assim como Plínio) seguem na mesma direção que estavam seguindo anteriormente. Nesse caso, Plínio segue para baixo até que faz a volta e cruza novamente a junção e dessa vez segue a direita e chega ao final.

Como o robô chegou ao final da esteira, seu programa deveria imprimir "ok." para esse caso.

Mudando apenas um caractere temos

```
[ > > > > > > v
    v
  > > > v > > > ]
  ^      v
  ^      v
  ^ < < <
```

Agora não há mais junção, e Plínio deveria detectar o loop. Mas como fazer isso? As partes da fábrica parecem tão iguais... Bem, há uma forma. O robô pode deixar uma marca pelos lugares que passar, substituindo o caractere naquele lugar da matriz por um '.'.

Fazendo isso, se o robô encontrar um '.' em seu caminho, sabe que já passou por ali antes e, portanto, há um loop! Há apenas uma exceção. Quando Plínio vê um '#' ele não substitui por '.' pois sabe que pode passar duas vezes pela mesma junção. Depois de percorrer o exemplo acima a matriz seria

```
[ . . . . . . .
    .
    . . . . . > > > ]
    .
    .
    . . . . .
```

Ou seja, o robô parou assim que percebeu que havia um loop, e então seu programa deverá imprimir "Loop infinito."

Por fim, para detectar se há falha na esteira é mais simples. Se o robô se encontra num lugar que não é esteira, não é junção e também não é o fim do percurso, significa que a esteira possui falha.

## Entradas

Seu programa receberá como entrada 32 linhas, cada uma com 64 caracteres mais um para o '\n' que indica o final da linha.

**Atenção:** Cada linha possui caracteres espaçados. Esses caracteres nas colunas de índice ímpar **não devem ser considerados**. Na verdade, se preferir, pode usar uma matriz de 32 x 32 e ignorar completamente essas colunas. Isso é feito para que as entradas e saídas sejam mais compreensíveis visualmente.

## Saídas

A saída deve ser a mensagem de Plínio numa linha e em seguida uma impressão da matriz com as marcações feitas pelo robô.

A saída do programa deve ter **exatamente** 32 linhas cada qual com 65 caracteres (64 da matriz + 1 para o '\n').

## Restrições

Uma esteira nunca vai levar o robô para fora dos limites da matriz.

O caractere imediatamente à direita do começo sempre será uma esteira.

Se o robô chegar no caractere final, sempre será pela direita.

**Atenção:** Os exemplos abaixo estão truncados para a facilidade de visualização. Se quiser as saídas completas, baixe os casos de teste.

### Exemplo 1:

Entrada:

```
[ > > > > > > v
    v
    > > > v > > > ]
    ^      v
    ^      v
    ^ < < <
```

Saída:

Loop infinito.

```
[ . . . . . . .
    .
    . . . . . > > > ]
    .
    .
    .
    . . . . .
```

### Exemplo 2:

Entrada:

```
[ > > > > > > ]
```

Saída:

```
Falha na esteira.  
[ . . . . . ]
```

**Exemplo 3:**

Entrada:

```
< < < < < [ > < ]
```

Saída:

```
Loop infinito.  
< < < < < [ . . ]
```

## Quaisquer dúvidas ou problemas a relatar

Envie uma mensagem para o monitor Gabriel Dertoni via telegram @GabrielDertoni (<https://t.me/GabrielDertoni>) ou no Discord da disciplina (<https://discord.gg/9gQ5YQfFsA>). Se preferir, também pode enviar um email para o professor Leonardo [leonardop@usp.br](mailto:leonardop@usp.br) (<mailto:leonardop@usp.br>) ou para o Gabriel [gab.dertoni@usp.br](mailto:gab.dertoni@usp.br) (<mailto:gab.dertoni@usp.br>)

[Esconder Descrição](#)**Este exercício aceita os seguintes tipos de arquivos:**[📄 Baixar Casos de Teste \(/Exercises/downloadCases/19907\)](/Exercises/downloadCases/19907)**Novo Envio**[📅 \(/Exercises/exportExerciseToGoogleCalendar/19907\)](/Exercises/exportExerciseToGoogleCalendar/19907)

O exercício está fechado

**23/06/2021 23:59:59**

Fechado

**Meu Último Envio**[📄 Download \(/Commits/download/1350840\)](/Commits/download/1350840)

status

**Finalizado**

compilado

**Sim**

casos corretos

**9/9**

pontuação

10.00

Caso	Status	Tempo de CPU	Tam. de Memória Utilizado	Mensagem
Caso 1	Correto	0.0020 s	-1 Kb	Resposta Correta
Caso 2	Correto	0.0018 s	-1 Kb	Resposta Correta
Caso 3	Correto	0.0017 s	-1 Kb	Resposta Correta
Caso 4	Correto	0.0014 s	-1 Kb	Resposta Correta
Caso 5	Correto	0.0014 s	-1 Kb	Resposta Correta
Caso 6	Correto	0.0015 s	-1 Kb	Resposta Correta
Caso 7	Correto	0.0031 s	-1 Kb	Resposta Correta
Caso 8	Correto	0.0015 s	-1 Kb	Resposta Correta
Caso 9	Correto	0.0016 s	-1 Kb	Resposta Correta

Detalhes dos Casos de Teste

Selecione um caso de teste...

▼

Histórico de Entregas

Data	Status	Corretos	Notas	Ações
08/06/2021 15:32:47	Finalizado	9/9	10.00	<div><div>Download (/Commits/download/1350840)</div><div>Detalhes (/commits/details/1350840)</div></div>
08/06/2021 15:32:11	Erro	0/9	0.00	<div><div>Download (/Commits/download/1350839)</div><div>Detalhes (/commits/details/1350839)</div></div>
03/06/2021 11:38:01	Incompleto	6/9	6.67	<div><div>Download (/Commits/download/1342925)</div><div>Detalhes (/commits/details/1342925)</div></div>
03/06/2021 11:36:02	Erro	0/9	0.00	<div><div>Download (/Commits/download/1342922)</div><div>Detalhes (/commits/details/1342922)</div></div>
03/06/2021 11:33:45	Incompleto	2/9	2.22	<div><div>Download (/Commits/download/1342916)</div><div>Detalhes (/commits/details/1342916)</div></div>