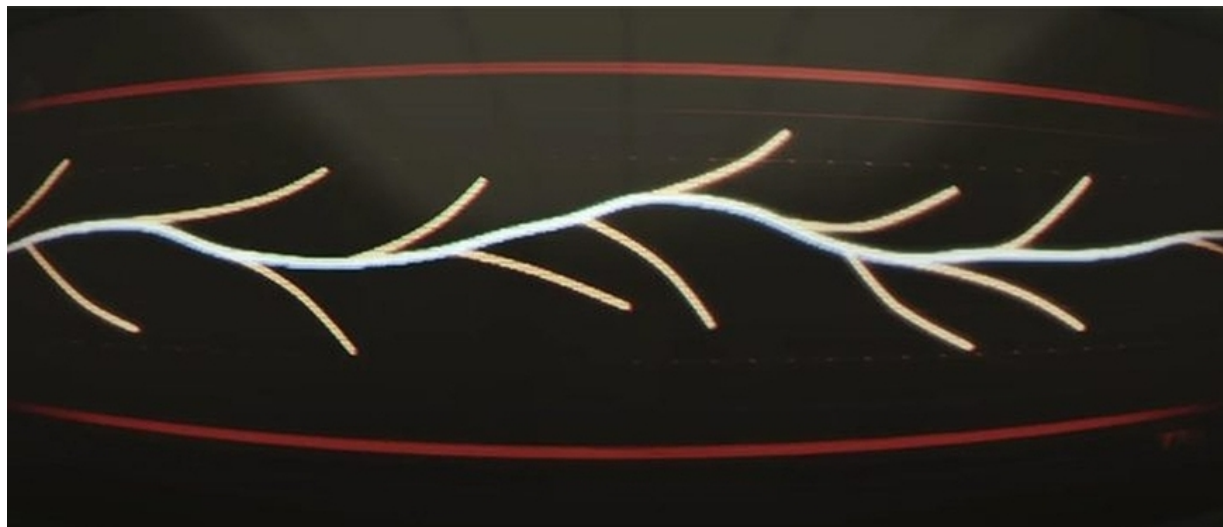


Linha do Tempo Sagrada



Devido a suas habilidades como programador você foi chamado para trabalhar para a Autoridade de Variância Temporal (AVT). O trabalho da AVT é proteger a linha do tempo sagrada removendo ramificações que possam criar linhas do tempo paralelas. A linha do tempo sofre uma ramificação quando um evento ocorre de forma diferente do esperado, se essa ramificação tiver consequências significativas para os acontecimentos da linha do tempo ela cria uma nova linha temporal. Eventos que chegam a esse ponto são chamados de eventos nexus.

O sua primeira tarefa na AVT é criar um código para filtrar os eventos que causam bifurcação na linha do tempo sagrada e determinar quais desses eventos são eventos nexus.

A linha do tempo sagrada é representada pela linha 5 de uma matriz de tamanho 11×50 (com os índices começando em 0), uma sequência de caracteres `#` marca essa linha. Uma bifurcação na linha do tempo sagrada é representada por uma sequência de caracteres `+` partindo da linha 5. As bifurcações são identificadas pela coluna onde elas ocorreram e a borda da matriz indica a fronteira a partir da qual as bifurcações se tornam permanentes (eventos nexus).

Você pode assumir as seguintes características com relação à representação utilizada:

- No máximo uma bifurcação ocorre em cada coluna.
- Nenhuma bifurcação acontece na primeira ou última coluna (colunas 0 e 49, respectivamente).
- Duas linhas do tempo não se interceptam e não tem elementos adjacentes, com exceção do momento da bifurcação, quando a linha do tempo sagrada

e a linha criada pela bifurcação são adjacentes.

- Uma linha do tempo não intercepta a si mesma e não tem elementos não consecutivos adjacentes.
- Toda expansão em uma linha do tempo ocorre sempre na vertical ou na horizontal.

Seu programa receberá a matriz 11×50 como entrada. Como saída, para cada bifurcação i , seu programa deve imprimir `Bifurcacao i:` seguido de `Evento Nexus`, quando a linha iniciada pela bifurcação se torna adjacente à borda da matriz, ou `Instavel`, quando a linha acabou antes de alcançar a borda da matriz.

Exemplos de entradas e saídas esperadas pelo seu programa:

Teste 03

Entrada

```
.....+......+.
.....+......+......+.
.....+......+......+.
.....+......+......+.
.....+......+......+.
.....+......+......+.
#####
.....+......+......+.
.....+......+......+.
.....+......+......+.
.....+......+......+.
.....+......+......+
```

Saída

```
Bifurcacao 6: Instavel
Bifurcacao 11: Evento Nexus
Bifurcacao 18: Instavel
Bifurcacao 25: Evento Nexus
Bifurcacao 29: Instavel
Bifurcacao 35: Evento Nexus
Bifurcacao 41: Instavel
Bifurcacao 45: Evento Nexus
```

Teste 06

Entrada

```
.....+......+......+.
.....++.....+.....+++.....++++.....+++.....+.
.....+
```

```
....++.....++.....++.....+.....+++.....++++..++..
...++.....++.....++.....++.....++.....+.....++...
...+.....+.....+.....+.....+.....+.....+.....+.....
#####
.....+.....+.....+.....+.....+.....
.....+.....+++++++.....+.....+++++...
.....+++++.....+.....+.....+.....+.....
.....+++++++.....+.....+++++.....+.....
.....+.....+.....+.....+.....+.....
```

Saída

```
Bifurcacao 3: Evento Nexus
Bifurcacao 10: Instavel
Bifurcacao 14: Evento Nexus
Bifurcacao 16: Evento Nexus
Bifurcacao 23: Instavel
Bifurcacao 24: Evento Nexus
Bifurcacao 31: Evento Nexus
Bifurcacao 39: Instavel
Bifurcacao 40: Instavel
Bifurcacao 42: Evento Nexus
Bifurcacao 45: Evento Nexus
```

Teste 09

Entrada

```
.....+.....+.....
.....++.....++.....+++++++
.....+++++.....+.....+++.....+.....
+++++.....+.....++++.....+.+.....++.....++++...
.....+.....+.....+.....+.....+.....
#####
.....+.....+.....+.....
.....++.....+.....+++.....+.....
.....+.....+++++.....+.+++.....
.....+++++.....+.....+.....
.....+.....+.....
```

Saída

```
Bifurcacao 10: Evento Nexus
Bifurcacao 15: Instavel
Bifurcacao 21: Evento Nexus
Bifurcacao 29: Evento Nexus
Bifurcacao 31: Instavel
Bifurcacao 39: Evento Nexus
```

Bifurcacao 43: Evento Nexus

Bifurcacao 46: Evento Nexus

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 dos dados da matriz.

```
matriz = []  
for i in range(11):  
    matriz.append(list(input()))
```

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 07/11/2021 (domingo).