

# Encaixe 2D

---

Jogos de encaixe proporcionam uma ótima atividade tanto para crianças como para adultos. As crianças, particularmente, são bastante beneficiadas por essa atividade, pois esse tipo de jogo estimula o desenvolvimento de orientação espacial. Diferentes modelos são desenvolvidos no mercado visando atrair o público adulto e infantil.



Uma grande empresa está desenvolvendo um novo jogo de encaixe 2D. O jogo é composto por diferentes tabuleiros e peças, sendo que os tabuleiros e as peças são retangulares e formados por células. Cada célula pode estar preenchida ou vazia. Em cada partida do jogo, o objetivo consiste em selecionar aleatoriamente um tabuleiro e uma peça e realizar o maior número possível de encaixes da peça no tabuleiro. Uma peça encaixa em uma determinada posição do tabuleiro quando todas as células preenchidas da peça estão alinhadas com células vazias do tabuleiro. Além disso, a peça deve ser encaixada sem ultrapassar as dimensões (largura e altura) do tabuleiro. O jogo permite ainda que a peça seja rotacionada para a direita em 90, 180 e 270 graus.

O engenheiro de produção da empresa possui a representação computacional de todos os tabuleiros e peças que são produzidos. Sabendo de sua habilidade com programação, ele pediu sua ajuda para determinar a quantidade máxima de encaixes que uma determinada peça pode ter em um determinado tabuleiro. Os tabuleiros e peças são representados por meio de matrizes, sendo que cada célula da matriz possui o caractere `X` ou `.`, indicando que a célula está preenchida ou vazia, respectivamente.

A seguir, temos um exemplo de um tabuleiro representado por uma matriz `5x6`.

```
. X X . X .  
X . . . . X  
X . . . . X  
. X X X X .  
X X X X X X
```

A seguir, temos um exemplo de uma peça representada por uma matriz `3x3`.

```
X . .  
. X X  
. X X
```

Perceba que só é possível encaixar a peça original (sem rotacioná-la) no tabuleiro a partir da primeira linha e primeira coluna. Além disso, rotacionando a peça em 90° também é possível encaixá-la a partir da primeira linha, tanto a partir da segunda coluna, quanto a partir da quarta. Já rotacionando 180°, podemos encaixar a peça a partir da segunda linha e quarta coluna. Por fim, rotacionando 270°, podemos encaixar a peça a partir da segunda linha e da primeira coluna.

O seu programa receberá como entrada um número inteiro `T`, que indica a altura do tabuleiro, seguido por `T` linhas que representam o tabuleiro. Em seguida, seu programa receberá um número inteiro `P`, que indica a altura da peça, seguido por `P` linhas que representam a peça. As células do tabuleiro e da peça estão separadas pelo caractere de espaço `" "`. Você pode assumir que a altura e largura da peça sempre será menor ou igual do que a altura e largura do tabuleiro.

A saída do seu programa deverá informar a quantidade de posições no tabuleiro em que é possível encaixar a peça original e rotacionada para direita em 90°, 180° e 270°.

Exemplos de entradas e saídas esperadas pelo seu programa:

## Teste 01

## Entrada

```
5
. X X . X .
X . . . . X
X . . . . X
. X X X X .
X X X X X X
3
X . .
. X X
. X X
```

## Saída

```
1, 2, 1, 1
```

## Teste 02

### Entrada

```
7
X X X X X X X
X X X X X X X
X X X . X X X
X X . . . X X
X X X X X X X
X X X X X X X
X X X X X X X
2
X X
X .
```

## Saída

```
0, 0, 1, 1
```

## Teste 03

### Entrada

```
12
. . . . . . . . . .
. . . . . . . . . .
. . . . . . . . . .
```

```

. . . X . . . X . . . X
. . . . . . . . . . .
. . . . . . . . . . .
. . . . . . . . . . .
. . . X . . . X . . . X
. . . . . . . . . . .
. . . . . . . . . . .
. . . . . . . . . . .
. . . X . . . X . . . X
4
X X X X
X X X X
X X X X
. X X X

```

## Saída

```
6,4,6,9
```

## Código Base

No arquivo auxiliar lab11.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 do tabuleiro e da peça.

```

# Leitura do tabuleiro
T = int(input())
tabuleiro = []
for _ in range(T):
    tabuleiro.append(input().split())

# Leitura da peça
P = int(input())
peca = []
for _ in range(P):
    peca.append(input().split())

```

## Orientações

- Veja [aqui](#) a página de submissão da tarefa.
- O arquivo a ser submetido deve se chamar lab11.py.
- No link "Arquivos auxiliares" há um arquivo compactado (aux11.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 20/11/2022 (domingo).