

```
1  """
2  Matheus Peixoto Ribeiro Vieira - 22.1.4104
3  """
4
5  import numpy as np
6
7  def maior_palindromo(palavra: str):
8      n = len(palavra)
9      if n == 1: return 1
10     if n == 2 :
11         if palavra[0] == palavra[1]: return 2
12         else: return 0
13
14     # A linha representa o comeco e a coluna o fim da substring
15     tabela = np.zeros((n,n))
16     tamanho_max = 1
17
18     # Substring de tamanho 1 sempre sera um palindromo
19     for i in range(n):
20         tabela[i][i] = 1
21
22     # Verificando se os dois primeiros elementos formam um palindromo
23     for i in range(n-1):
24         if palavra[i] == palavra[i+1]:
25             tabela[i][i+1] = 1
26             tamanho_max = 2
27
28     # vai variar para cada tamanho possivel de palavra >= 3
29     for tamanho in range(3, n + 1):
30         # Vai passando na diagonal superior que ainda nao foi verificada
31         for inicio in range(n - tamanho + 1):
32             fim = inicio + tamanho - 1
33
34             # Verifica se palavra e um palindromo
35             if palavra[inicio] == palavra[fim] and tabela[inicio + 1][fim - 1] == 1:
36                 tabela[inicio][fim] = 1
37
38                 if tamanho > tamanho_max:
39                     tamanho_max = tamanho
40
41     print(tabela)
42     print(tamanho_max)
43
44
45 palavra = "aaaabbbaa"
46 maior_palindromo(palavra)
```

```
47
48 """
49 Execucao:
50 [[1. 1. 1. 1. 0. 0. 0. 0.]
51  [0. 1. 1. 1. 0. 0. 0. 0.]
52  [0. 0. 1. 1. 0. 0. 0. 1.]
53  [0. 0. 0. 1. 0. 0. 1. 0.]
54  [0. 0. 0. 0. 1. 1. 0. 0.]
55  [0. 0. 0. 0. 0. 1. 0. 0.]
56  [0. 0. 0. 0. 0. 0. 1. 1.]
57  [0. 0. 0. 0. 0. 0. 0. 1.]]
58
59 6
60 """
```