Processamento da Informação
– Teoria –

Exercícios sobre strings

Semana 04

Prof. Jesús P. Mena-Chalco

18/05/2013

Uma string é uma sequência

Uma string (cadeia) é uma sequência de caracteres.

Podemos acessar aos caracteres com o operador colchete

```
>>> fruta = "banana"
>>> letra = fruta[1]
```

Uma string é uma sequência

A primeira letra (“b”) tem a posição 0. A segunda letra (“a”) tem a posição 1, ...

```
>>> fruta = "banana"

>>> print fruta[0]
 b
>>> print fruta[1]
 a
>>> print fruta[2]
 n
```

Índices

Strings são imutáveis

É tentador usar o operador [ ] no lado esquerdo de uma atribuição, com a intenção de mudar um caractere em uma string.

Por exemplo:

```
>>> greeting = 'Hello, world!'
>>> greeting[0] = 'J'
TypeError: object does not support item assignment
```

Strings são imutáveis

Parte do conteúdo de uma string não pode ser modificado de forma direta.

Uma alternativa é a criação de uma nova string que contenha a modificação:

```
>>> greeting = 'Hello, world!'
>>> new_greeting = 'J' + greeting[1:len(greeting)-1]
>>> print new_greeting
Jello, world!
```

Atividade na aula anterior ...

Crie uma função que receba duas palavras e retorne True se uma das palavras é o reverso da outra:

```
def reverso(palavra1, palavra2):
```

Exemplo:

'pots' é reverso de 'stop'

'livres' é reverso de 'servil'

Atividade na aula anterior ...

```
1  def reverso(palavra1, palavra2):
2      if len(palavra1) != len(palavra2):
3          return False
4      i = 0
5      j = len(palavra2)-1
6      while j>=0:
7          if palavra1[i] != palavra2[j]:
8              return False
9          i = i+1
10         j = j-1
11     return True
```

Atividade na aula anterior ...

```
def inverter_palavra(palavra):
    temporal = ""
    i = len(palavra)-1
    while i>=0:
        temporal = temporal + palavra[i]
        i=i-1
    return temporal
```

Atividade na aula anterior ...

```
def reverso2(palavra1, palavra2):
    if palavra1 == inverter_palavra(palavra2):
        return True
    else:
        return False
```

Atividade na aula anterior ...

```
def reverso3(palavra1, palavra2):
    if len(palavra1) != len(palavra2):
        return False
    n = len(palavra1)
    i = 0
    while i<n:
        if palavra1[i] != palavra2[n-1-i]:
            return False
        i = i+1
    return True
```

Atividade na aula anterior ...

Crie uma função que receba duas palavras e retorne True caso a primeira palavra seja um prefixo da segunda:

Cabeçalho: def prefixo (palavra1, palavra2):

Exemplo: 'uf' é prefixo de 'ufabc'

Atividade na aula anterior ...

Solução com erro...

```
1  def prefixo(palavra1, palavra2):
2      if len(palavra1) > len(palavra2):
3          return False
4      if palavra1 in palavra2:
5          return True
6      else:
7          return False
```

```
>>> prefixo("uf","ufabc")
1
>>> prefixo("uf","abcuf")
1
```

Atividade na aula anterior ...

```
1  def prefixo(palavra1, palavra2):
2      if len(palavra1) > len(palavra2):
3          return False
4      i=0
5      while i<len(palavra1):
6          if palavra1[i]!=palavra2[i]:
7              return False
8          i=i+1
9      return True
```

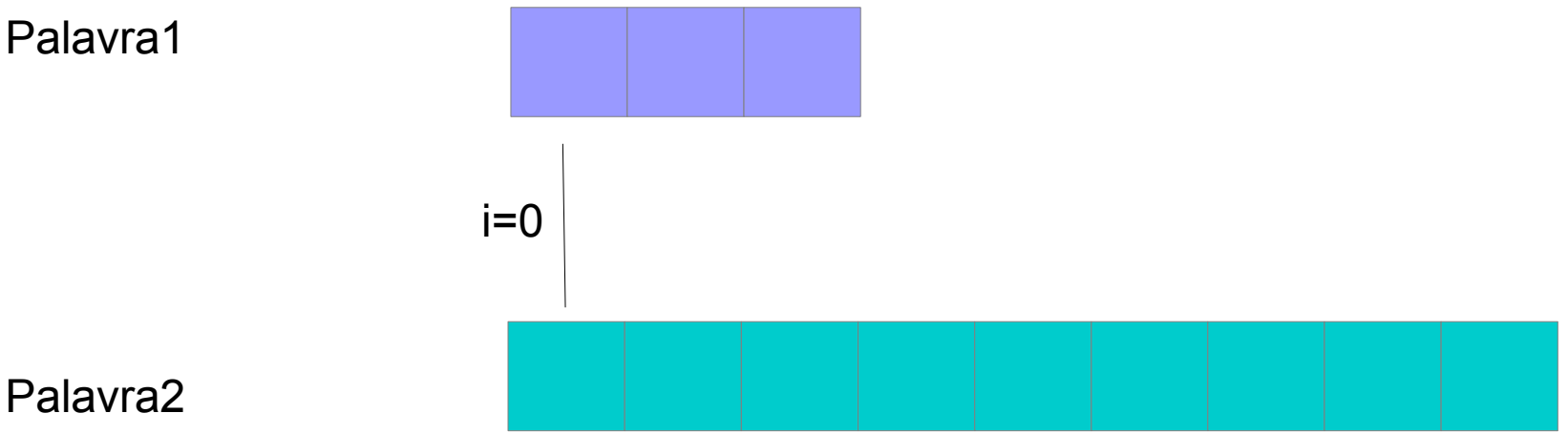
```
>>> prefixo("uf","ufabc")
1
>>> prefixo("uf","abcuf")
0
>>> prefixo("ufabc","uf")
0
```

```
1  def prefixo(palavra1, palavra2):
2      if len(palavra1) > len(palavra2):
3          return False
4      i=0
5      while i<len(palavra1):
6          if palavra1[i]!=palavra2[i]:
7              return False
8          i=i+1
9      return True
```

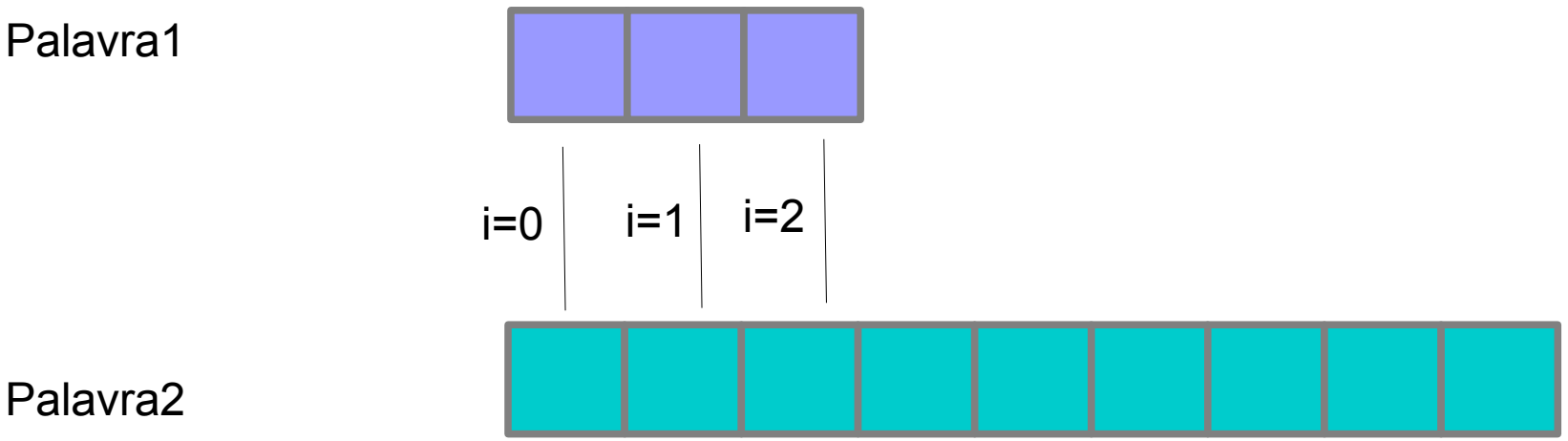
Palavra1

Palavra2

```
1  def prefixo(palavra1, palavra2):
2      if len(palavra1) > len(palavra2):
3          return False
4      i=0
5      while i<len(palavra1):
6          if palavra1[i]!=palavra2[i]:
7              return False
8          i=i+1
9      return True
```



```
def prefixo(palavra1, palavra2):
    if len(palavra1) > len(palavra2):
        return False
    i=0
    while i<len(palavra1):
        if palavra1[i]!=palavra2[i]:
            return False
        i=i+1
    return True
```



Atividade na aula anterior ...

Crie uma função que receba duas palavras e retorne True caso a primeira palavra seja um sufixo da segunda:

Cabeçalho: def sufixo (palavra1, palavra2):

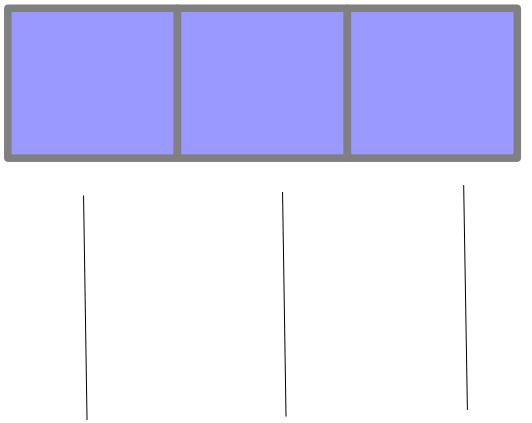
Exemplo: 'abc' é sufixo de 'ufabc'

Atividade na aula anterior ...

```
def sufixo(palavra1, palavra2):
    n1 = len(palavra1)
    n2 = len(palavra2)
    if n1 > n2:
        return False
    j = 0
    while j<n1:
        if palavra1[n1-1-j] != palavra2[n2-1-j]:
            return False
        j = j+1
    return True
```

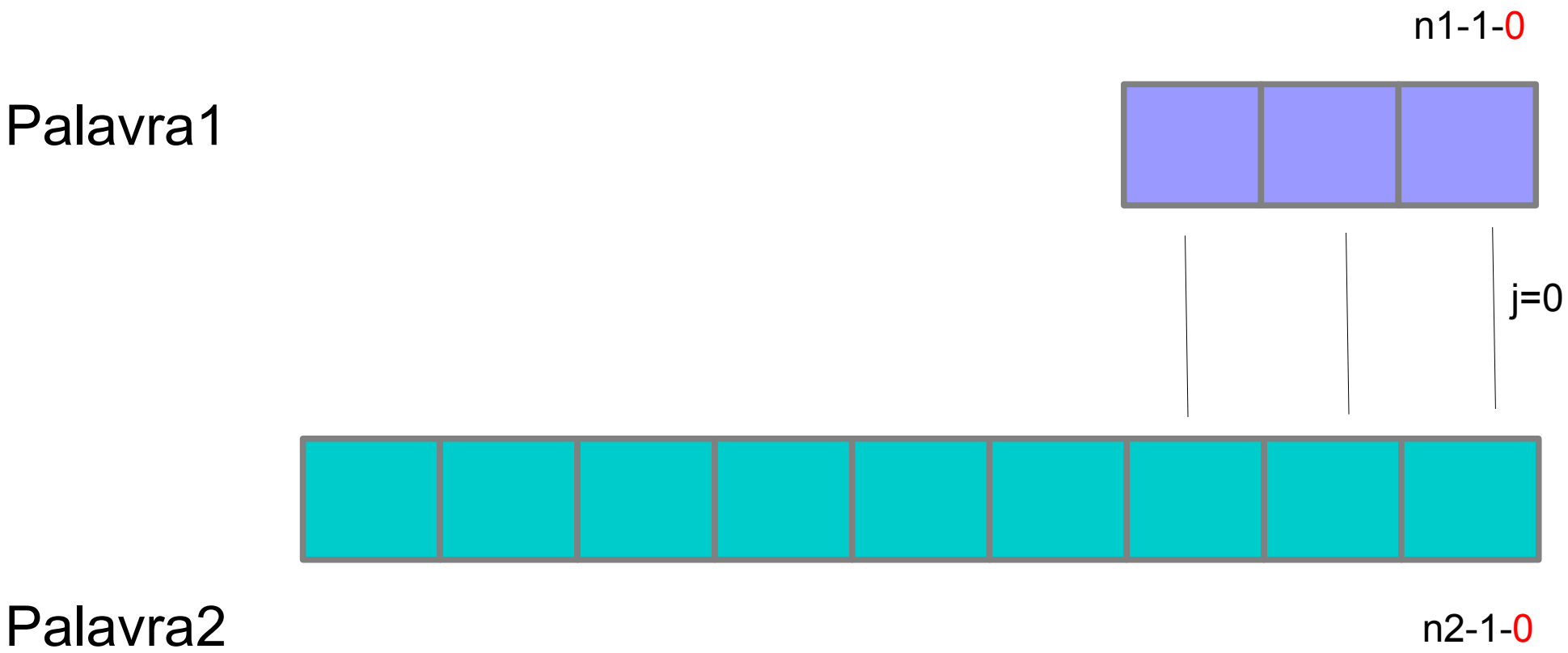
Atividade na aula anterior ...

Palavra1

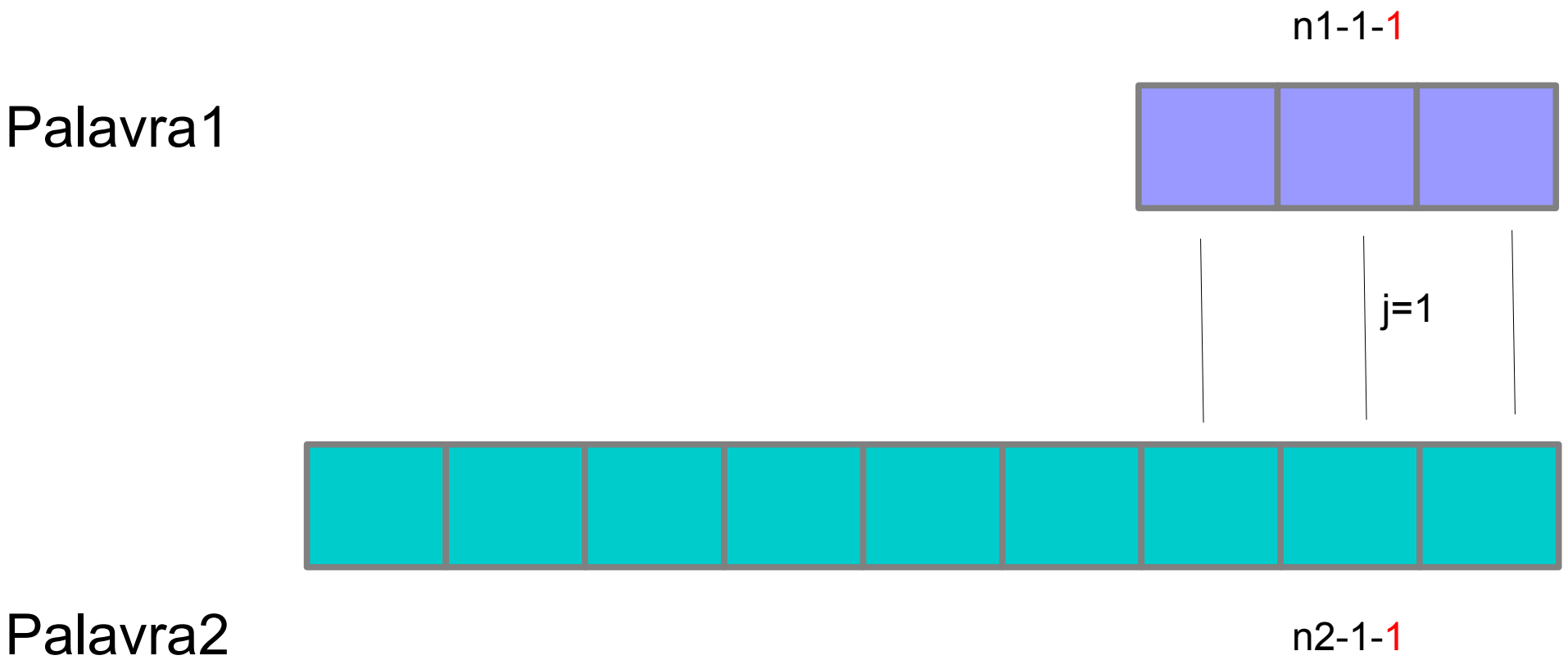


Palavra2

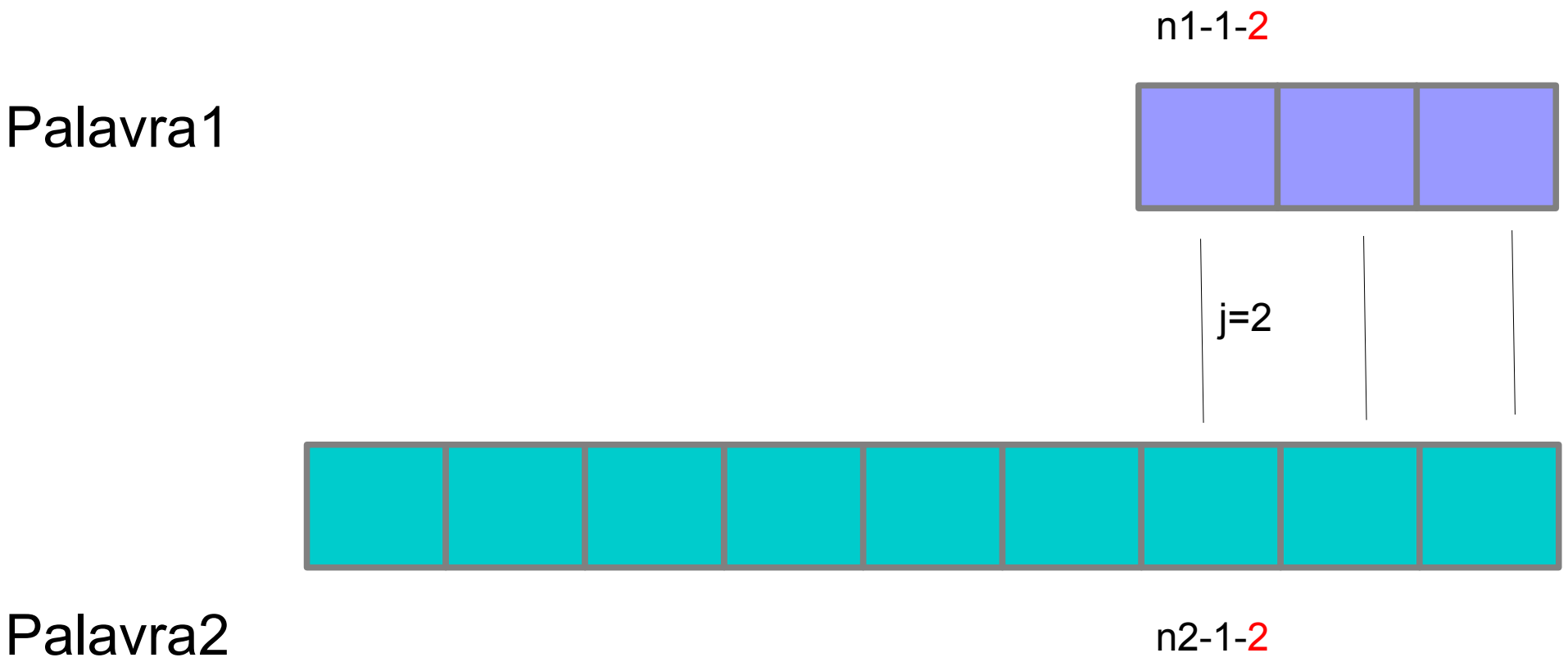
Atividade na aula anterior ...



Atividade na aula anterior ...



Atividade na aula anterior ...



Atividade em Aula

Questão 1: Indique a mensagem que apresentará a execução da seguintes função. Considere como parâmetro de entrada a string “abracadabra”

```
def funcao1(frase):
    str1 = ""
    str2 = ""
    k = len(frase)-1
    while k>=0:
        str1 = str1 + frase[k]
        str2 = frase[k] + str2
        k = k-1
    print str1
    print str2
```

Atividade em Aula

Questão 2: Indique a mensagem que apresentará a execução da seguintes função. Considere como parâmetro de entrada a string “UFABC” e “123”

```
def funcao2(palavra):
    str = ''
    k = 0
    while k<len(palavra):
        str = str + palavra[k]
        k = k+1
    while k>0:
        k = k-1
        str = str + palavra[k]
    print str
```

Atividade em Aula

Questão 3: Indique a mensagem que apresentará a execução da seguintes função. Considere como parâmetro de entrada a string “um dois tres”.

```
def funcao3(frase):
    contador = 0
    k = 0
    while k<len(frase)/2:
        if frase[k]==" ":
            contador = contador+1
    print contador
```

Atividade em Aula

Questão 1: Indique a mensagem que apresentará a execução da seguintes função. Considere como parâmetro de entrada a string “abracadabra”

```
def funcao1(frase):
    str1 = ""
    str2 = ""
    k = len(frase)-1
    while k>=0:
        str1 = str1 + frase[k]
        str2 = frase[k] + str2
        k = k-1
    print str1
    print str2
```

Resposta:

```
>>> funcao1("abracadabra")
arbadacarba
abracadabra
```

Atividade em Aula

Questão 2: Indique a mensagem que apresentará a execução da seguintes função. Considere como parâmetro de entrada a string “UFABC” e “123”

```
def funcao2(palavra):
    str = ''
    k = 0
    while k<len(palavra):
        str = str + palavra[k]
        k = k+1
    while k>0:
        k = k-1
        str = str + palavra[k]
    print str
```

Resposta:

```
>>> funcao2("UFABC")
UFABCCBAFU

>>> funcao2("123")
123321
```

Atividade em Aula

Questão 3: Indique a mensagem que apresentará a execução da seguintes função. Considere como parâmetro de entrada a string “um dois tres”.

```
def funcao3(frase):
    contador = 0
    k = 0
    while k<len(frase)/2:
        if frase[k]==" ":
            contador = contador+1
    print contador
```

Resposta:

Loop infinito, sem resposta.

Atividade em Aula

Questão 3: Indique a mensagem que apresentará a execução da seguintes função. Considere como parâmetro de entrada a string “um dois tres”.

```
def funcao3(frase):
    contador = 0
    k = 0
    while k<len(frase)/2:
        if frase[k]==" ":
            contador = contador+1
        k = k+1
    print contador
```

Resposta:

```
>>> funcao3("um dois tres")
1
```

String palíndroma

Fazer uma função que receba como parâmetro uma string e verifique se ela é palíndroma, isto é, se ela é igual lida da esquerda para a direita e vice-versa.

Exemplos:
- "RADAR" é palíndroma
- "B123321B" é palíndroma
- "oaio" não é palíndroma

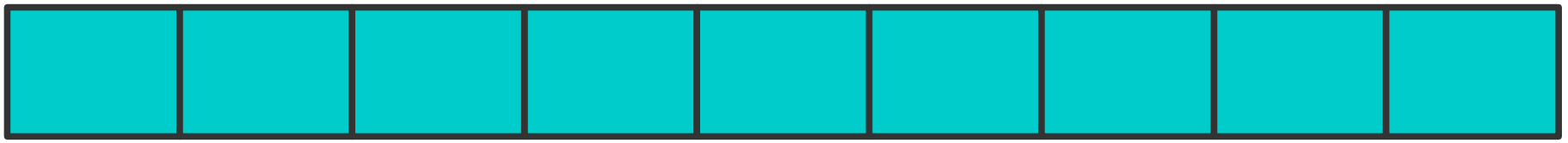
Cabeçalho: def palindroma(palavra):

String palíndroma: solução 01

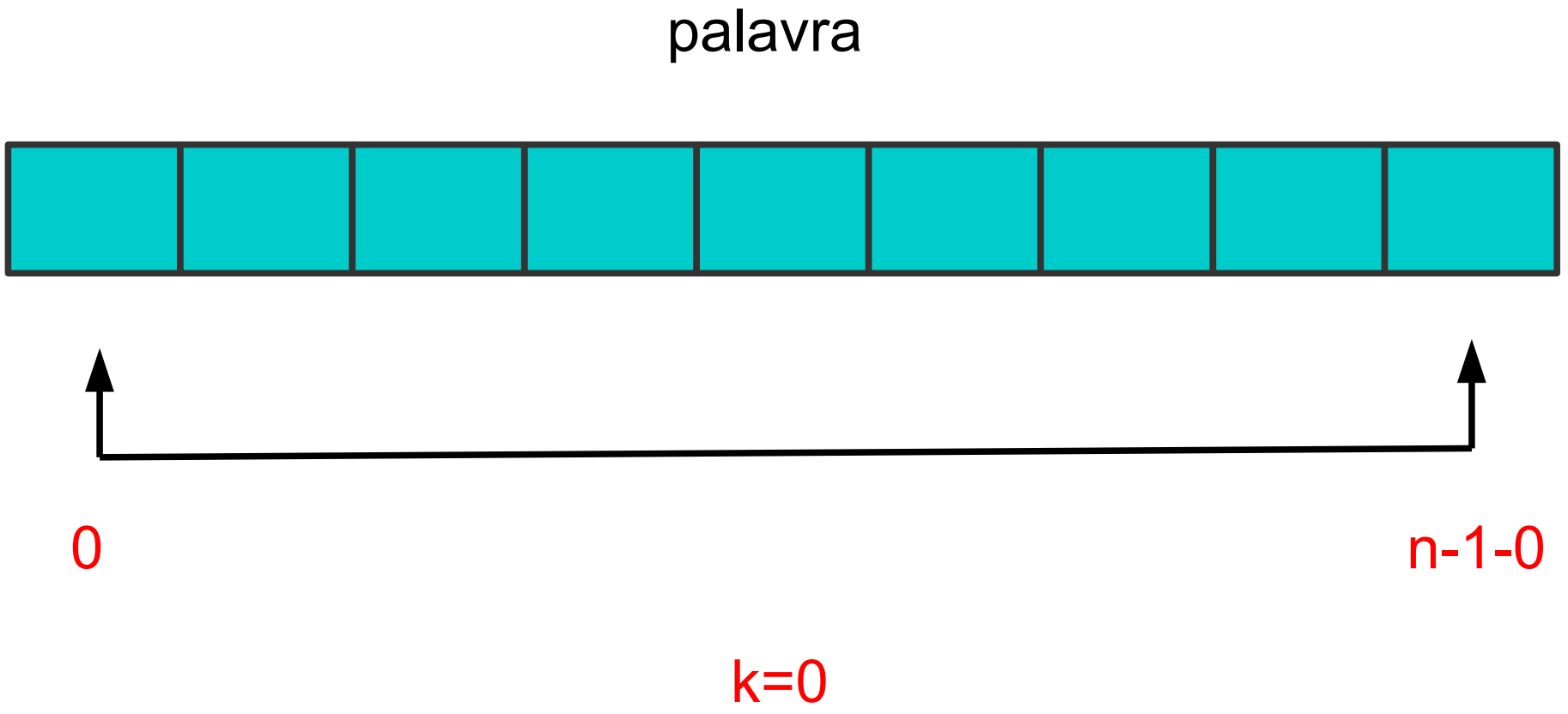
```
def inverter_palavra(palavra):
    temporal = ""
    i = len(palavra)-1
    while i>=0:
        temporal = temporal + palavra[i]
        i=i-1
    return temporal

def palindroma1(str):
    if str == inverter_palavra(str):
        return True
    else:
        return False
```

String palíndroma: solução 02

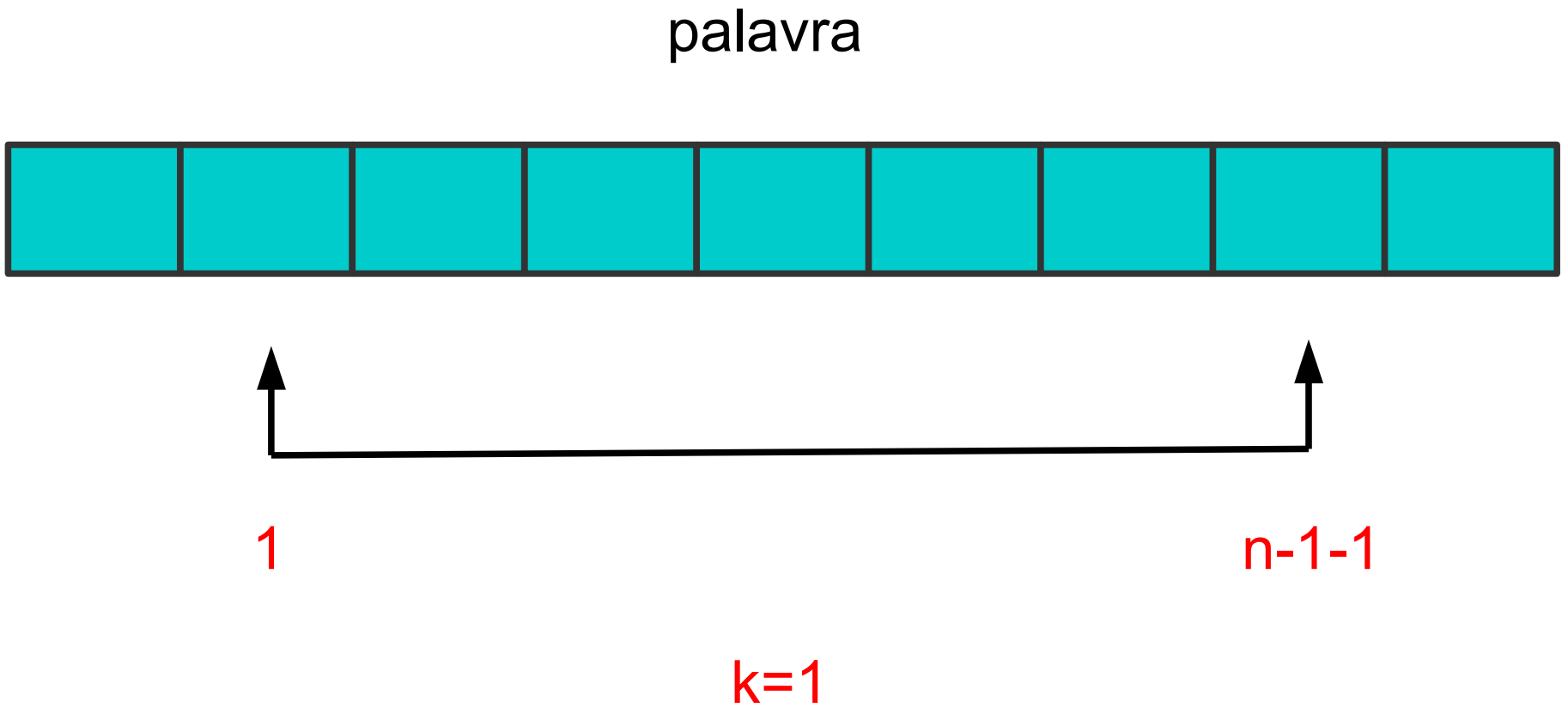
palavra



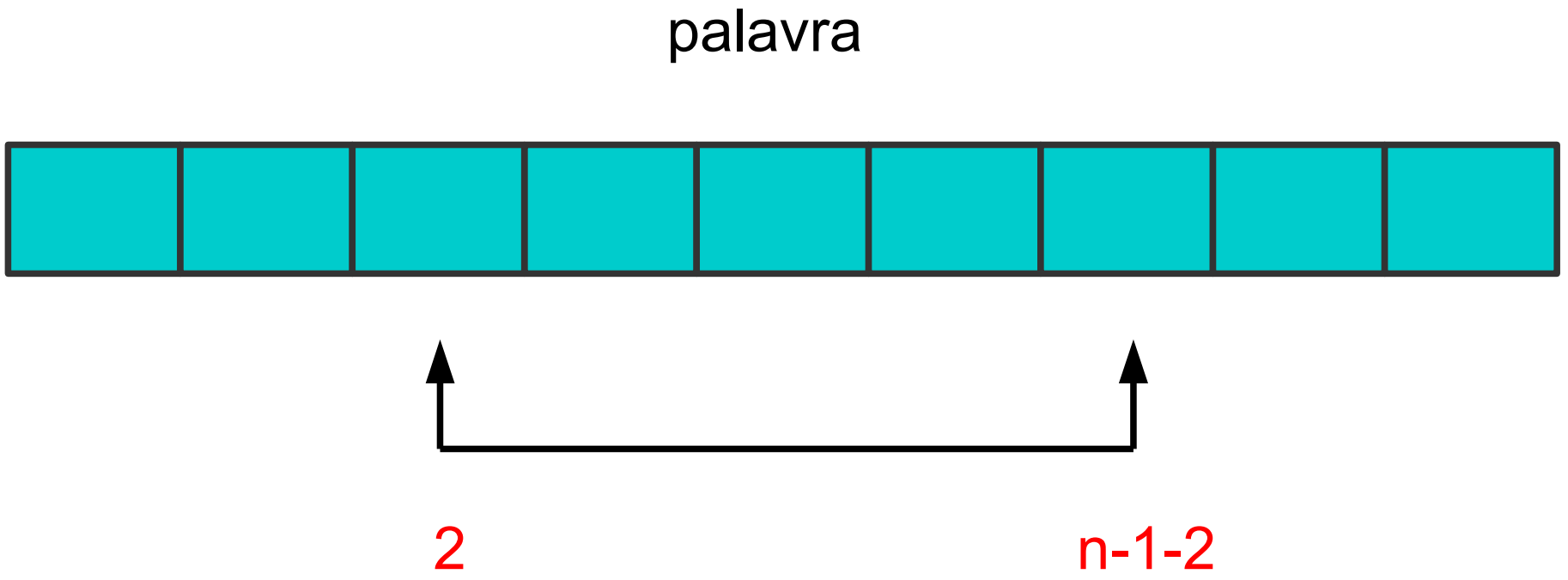
String palíndroma: solução 02



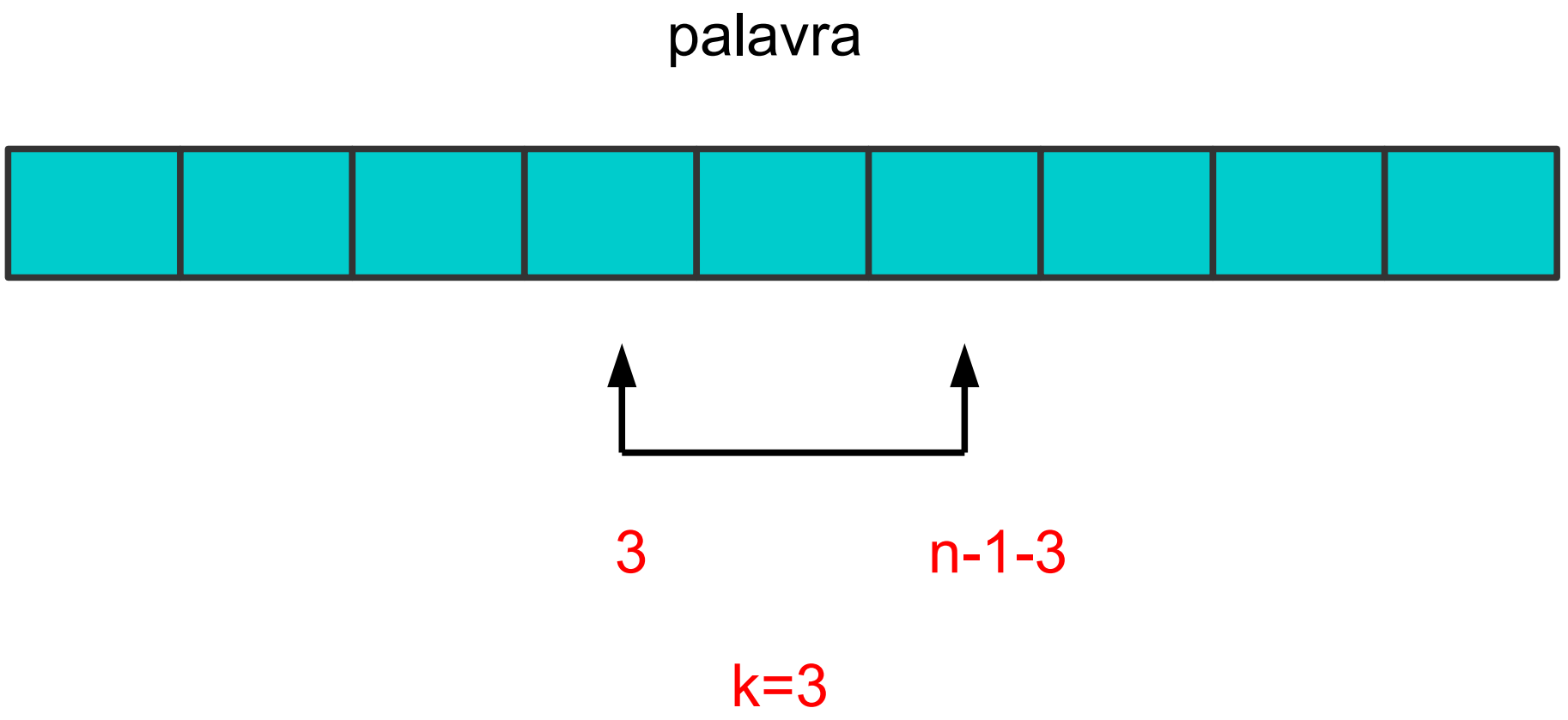
String palíndroma: solução 02



String palíndroma: solução 02



String palíndroma: solução 02



String palíndroma: solução 02

```
def palindroma2(str):
    n = len(str)
    k = 0
    while k<n/2:
        if str[k] != str[n-k-1]:
            return False
        k = k+1
    return True
```

String palíndroma: solução 02

```
def palindroma2(str):
    n = len(str)
    k = 0
    while k<n/2:
        if str[k] != str[n-k-1]:
            return False
        k = k+1
    return True
```

```
>>> palindroma2("3")
1
>>> palindroma2("33")
1
>>> palindroma2("333")
1
>>> palindroma2("3333")
1
>>> palindroma2("luzazul")
1
>>> palindroma2("sodedos")
1
>>> palindroma2("sodeedos")
1
>>> palindroma2("sodeedo")
0
>>> palindroma2("sode1451edos")
0
```

Última palavra

Crie uma função que receba, como parâmetro, uma string e imprima somente a última palavra da mesma.

Exemplo: Se a string for "José da Silva", deverá ser impresso na tela a substring "Silva".

Cabeçalho: def ultima_palavra(frase):

Última palavra

```
def ultima_palavra(frase):
    palavra = ""
    k = len(frase)-1
    while k>=0:
        if frase[k]!=" ":
            palavra = frase[k]+palavra
        else:
            break
        k = k-1
    print palavra
```

Última palavra: Solução

```
>>> ultima_palavra("Jose da Silva")
Silva
>>> ultima_palavra("Jose da")
da
>>> ultima_palavra("Jose")
Jose
>>> ultima_palavra("Universidade Federal do ABC")
ABC
>>> ultima_palavra("Universidade Federal do ABC, PI")
PI
```

Lista 04: Número de palavras

Questão única: Crie uma função para ler uma frase (string) e contar o número de palavras dessa frase. Considere que as palavras estão separadas por espaços brancos ou vírgulas.

Exemplos:
“Processamento” contém 1 palavra.
“Processamento da informação” contém 3 palavras.
“computador, caderno e caneta” contém 4 palavras.
“ linux ” contém 1 palavra.
“ ” não contém palavras.
“ , , , ” não contém palavras.

Cabeçalho: def conta_palavras(frase):

Lista 04: Número de palavras

A entrega da Lista 04 deverá ser enviada através do Tidia-ae.

Seção Atividades/lista-04. Até 24/05 (23h50) – Sexta-feira.

Deve ser enviado um arquivo PDF contendo a solução da questão. Também deverão ser apresentadas (no mesmo documento) as respostas para cada um dos exemplos descritos no enunciado.

O documento deve ter o seguinte nome:
RA-SeuNomeCompleto-Lista-04.pdf