TÉCNICO LISBOA

Fundamentos de Programação - 2020/2021 Aula Prática 04 (30 minutos) Turno 2ª feira 09:00-10:30
Nome:
Número:
Data:
Curso:

Capítulo 4 - Tuplos e Ciclos Contados

Escreva em Python a função `duplica` que recebe um tuplo e retorna como resultado um tuplo idêntico ao original, mas em que cada elemento está repetido.
Não pode usar a instrução while. Não necessita validar os argumentos.
Por exemplo,

```
>>> duplica(())
()
>>> duplica((1, 2, 3))
(1, 1, 2, 2, 3, 3)
```

Solução:

```
def duplica(tuplo):
    newtuplo = ()
    for e in tuplo:
        newtuplo = newtuplo + (e, e)
    return newtuplo
```

TÉCNICO LISBOA

Fundamentos de Programação - 2020/2021 Aula Prática 04 (30 minutos) Turno 2ª feira 10:30-12:00
Nome:
Número:
Data:
Curso:

Capítulo 4 - Tuplos e Ciclos Contados

Escreva em Python a função `insere` que recebe um tuplo e um valor, e retorna como resultado um tuplo idêntico ao original, mas em que após cada elemento é inserido o valor passado como parâmetro. Não pode usar a instrução while. Não necessita validar os argumentos.
Por exemplo,

```
>>> insere((), 2)
()
>>> insere((1, 2, 3), 7)
(1, 7, 2, 7, 3, 7)
```

Solução:

```
def insere(tuplo,v):
    newtuplo = ()
    for e in tuplo:
        newtuplo = newtuplo + (e, v)
    return newtuplo
```

TÉCNICO LISBOA

Fundamentos de Programação - 2020/2021 Aula Prática 04 (30 minutos) Turno 5ª feira 10:30-12:00
Nome:
Número:
Data:
Curso:

Capítulo 4 - Tuplos e Ciclos Contados

Escreva em Python a função `juntos` que recebe um tuplo contendo inteiros e tem como valor o número de elementos iguais adjacentes.
Não pode usar a instrução while. Não necessita validar os argumentos.
Por exemplo,

```
>>> juntos((1, 2, 2, 3, 4, 4, 4))
3
>>> juntos((1, 2, 2, 3, 4))
1
```

Solução 1:

```
def juntos(tupla):
    previous = None
    count = 0
    for v in tupla:
        if previous == v:
            count += 1
        previous = v
    return count
```

Solução 2:

```
def juntos(tuplo):
        n_juntos = 0
        for i in range(len(tuplo) - 1):
            if tuplo[i] == tuplo[i + 1]:
                n_juntos = n_juntos + 1
        return n_juntos
```

TÉCNICO LISBOA

Fundamentos de Programação - 2020/2021 Aula Prática 04 (30 minutos) Turno 5ª feira 14:00-15:30
Nome:
Número:
Data:
Curso:

Capítulo 4 - Tuplos e Ciclos Contados

Escreva uma função em Python chamada `ordenado_decrescente` que recebe um tuplo contendo números inteiros, e devolve True se os elementos do tuplo estiverem por ordem decrescente ou `False` em caso contrário. Não pode usar a instrução `while`, nem a função `sorted`.
Assuma que o tuplo contém pelo menos um elemento. Não necessita validar os argumentos.
Por exemplo,

```
>>> ordenado_decrescente((7, 5, 1))
True
>>> ordenado_decrescente((202, 33, 23, 4, 76))
False
```

Solução:

```
def ordenado_decrescente(tuplo):
    for i in range(1, len(tuplo)):
        if tuplo[i-1] < tuplo[i]:
            return False
    return True
```

TÉCNICO LISBOA

Fundamentos de Programação - 2020/2021 Aula Prática 04 (30 minutos) Turno 6ª feira 10:00-11:30
Nome:
Número:
Data:
Curso:

Capítulo 4 - Tuplos e Ciclos Contados

Escreva uma função em Python com o nome `diferentes` que recebe um tuplo contendo números inteiros e um número inteiro, e que devolve um tuplo com todos os elementos do tuplo (pela mesma ordem) que são diferentes do segundo argumento.
Não pode usar a instrução while. Não necessita validar os argumentos.
Por exemplo,

```
>>> diferentes((3, 4, 5, 6, 7), 5)
(3, 4, 6, 7)
>>> diferentes((3, 4, 5, 6, 7), 8)
(3, 4, 5, 6, 7)
```

Solução:

```
def diferentes(tuplo, num):
    newtuplo = ()
    for e in tuplo:
        if e != num:
            newtuplo = newtuplo + (e,)
    return newtuplo
```

\

TÉCNICO LISBOA

Fundamentos de Programação - 2020/2021 Aula Prática 04 (30 minutos) Turno 6ª feira 14:00-15:30
Nome:
Número:
Data:
Curso:

Capítulo 4 - Tuplos e Ciclos Contados

Escreva uma função em Python com o nome `maiores` que recebe um tuplo contendo números inteiros e um número inteiro e que devolve um tuplo com todos os elementos do tuplo (pela mesma ordem) que são maiores do que o segundo argumento.
Não pode usar a instrução while. Não necessita validar os argumentos.
Por exemplo,

```
>>> maiores((3, 4, 5, 6, 7), 5)
(6, 7)
>>> maiores((3, 4, 5, 6, 7), 8)
()
```

Solução:

```
def menores(tuplo, num):
    newtuplo = ()
    for e in tuplo:
        if e > num:
            newtuplo = newtuplo + (e,)
    return newtuplo
```

TÉCNICO LISBOA

Fundamentos de Programação - 2020/2021 Aula Prática 04 (30 minutos) Turno 6ª feira 15:30-17:00
Nome:
Número:
Data:
Curso:

Capítulo 4 - Tuplos e Ciclos Contados

Escreva uma função em Python com o nome `menores` que recebe um tuplo contendo números inteiros e um número inteiro e que devolve um tuplo com todos os elementos do tuplo (pela mesma ordem) que são menores do que o segundo argumento.
Não pode usar a instrução while. Não necessita validar os argumentos.
Por exemplo,

```
>>> menores((3, 4, 5, 6, 7), 5)
(3, 4)
>>> menores((3, 4, 5, 6, 7), 0)
()
```

Solução:

```
def menores(tuplo, num):
    newtuplo = ()
    for e in tuplo:
        if e < num:
            newtuplo = newtuplo + (e,)
    return newtuplo
```