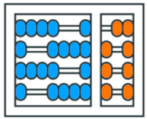


Instituto de Computação

UNIVERSIDADE ESTADUAL DE CAMPINAS

UNICAMP

# MC102 - Aula 08

## Exemplos sobre Listas e Tuplas

Algoritmos e Programação de Computadores

Turmas

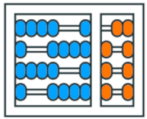
**OVXZ**

Prof. Lise R. R. Navarrete

lrommel@ic.unicamp.br

Terça-feira, 12 de abril de 2022

21:00h - 23:00h (CB06)



Instituto de Computação

UNIVERSIDADE ESTADUAL DE CAMPINAS

UNICAMP

# MC102 – Algoritmos e Programação de Computadores

Turmas

**OVXZ**

https://ic.unicamp.br/~mc102/

Site da Coordenação de MC102

Aulas teóricas:

Terça-feira, 21:00h - 23:00h (CB06)

Quinta-feira, 19:00h - 21:00h (CB06)

# Conteúdo

- Exercício 1
- Exercício 2
- Exercício 3
- Exercício 4
- Exercício 5

# Exercício 1

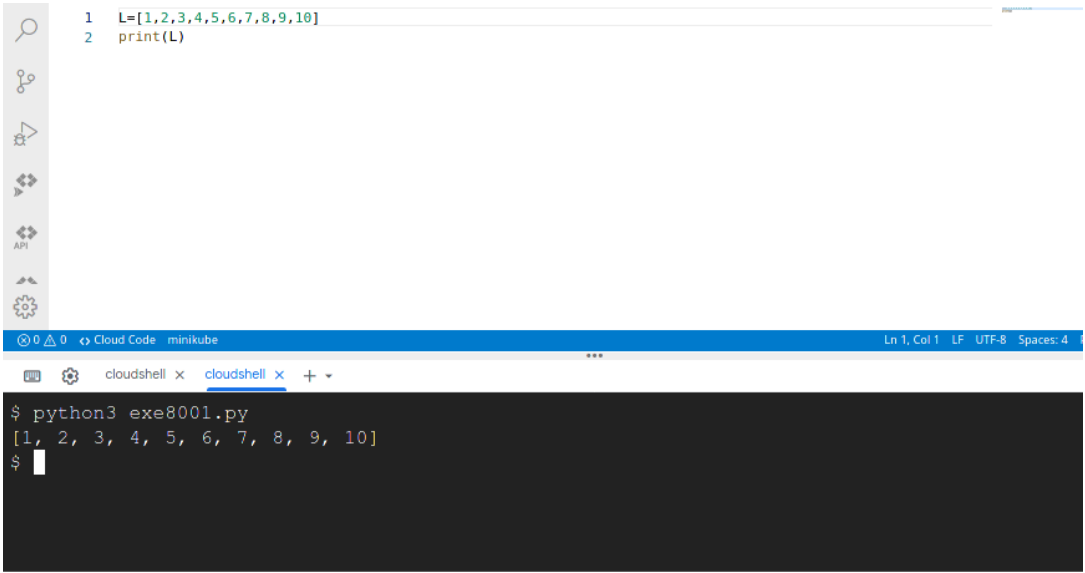
https://ic.unicamp.br/~mc102/aulas/aula05.pdf

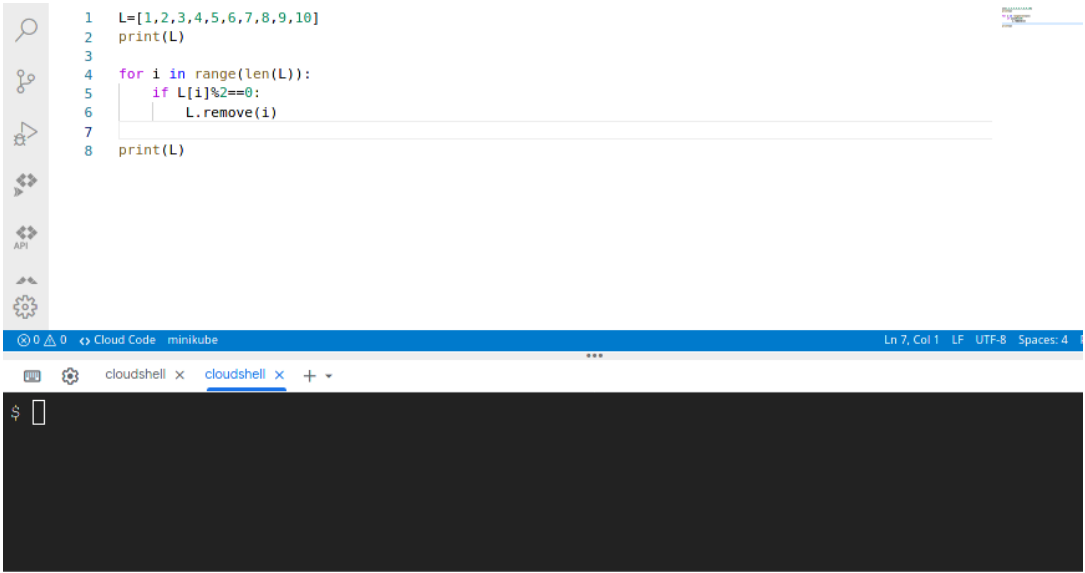
**Descrição**

Dada uma lista $L$ de $n$ valores inteiros, escreva um programa que remova todos os números pares da lista.

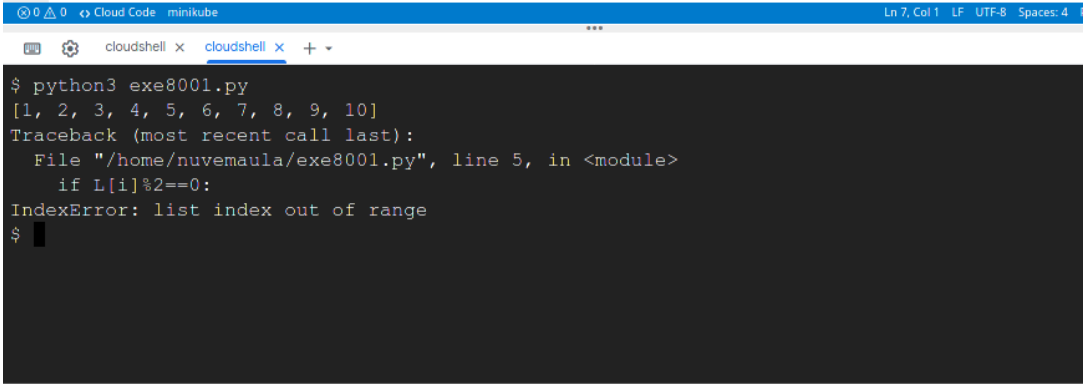
Exemplo:
Tamanho da lista $L$: 10
$L$: 1 2 3 4 5 6 7 8 9 10

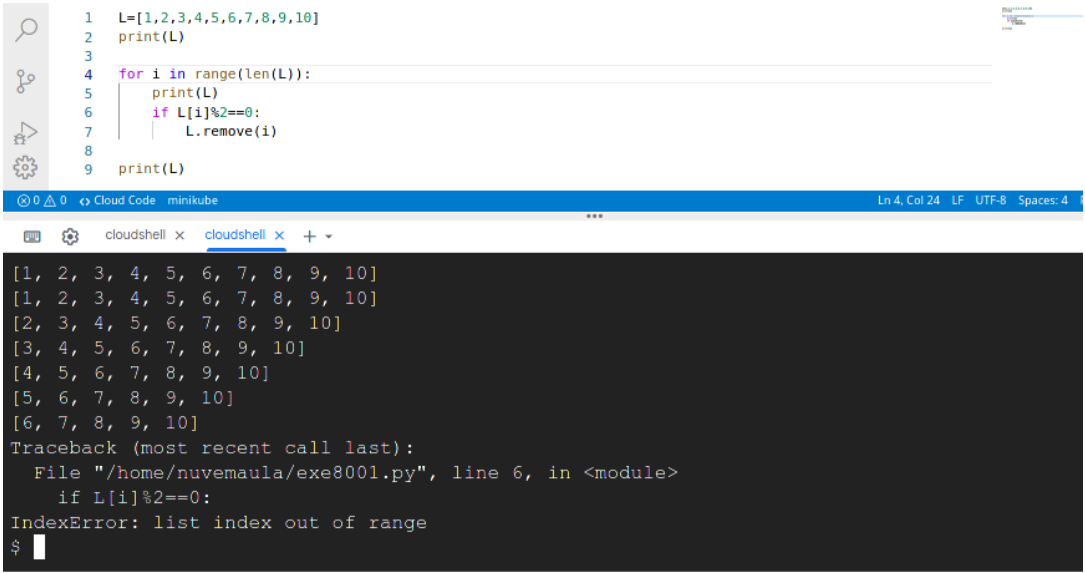
Resposta:
1 3 5 7 9

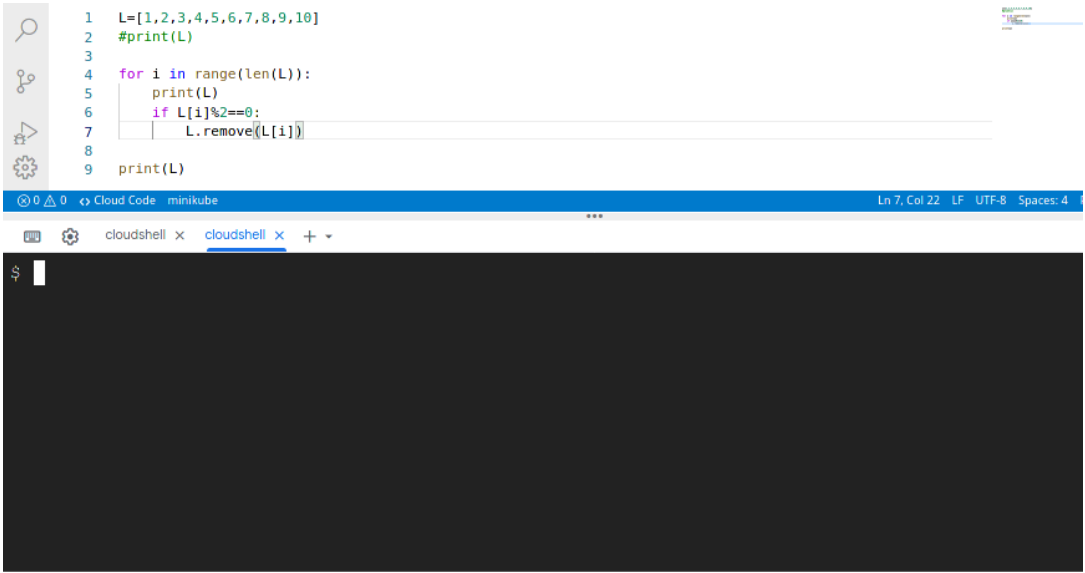

1 L=[1,2,3,4,5,6,7,8,9,10]
2 print(L)
API
0 0 Cloud Code minikube
Ln 1, Col 1 LF UTF-8 Spaces: 4
cloudshell cloudshell
$ python3 exe8001.py
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
$

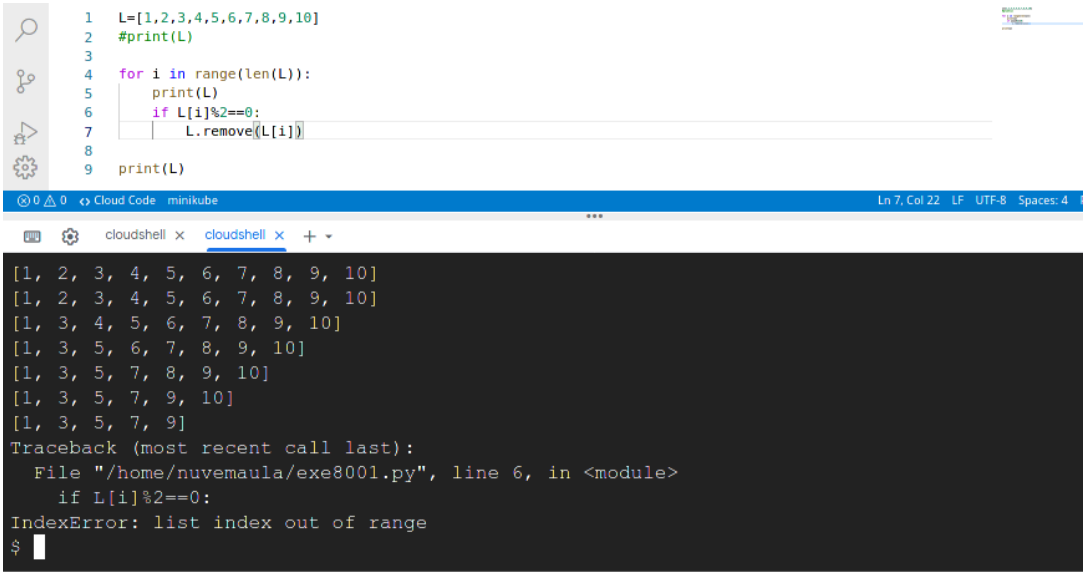

1 L=[1,2,3,4,5,6,7,8,9,10]
2 print(L)
3
4 for i in range(len(L)):
5 if L[i]%2==0:
6 L.remove(i)
7
8 print(L)
API
0 0 Cloud Code minikube
Ln 7, Col 1 LF UTF-8 Spaces: 4
cloudshell cloudshell
$

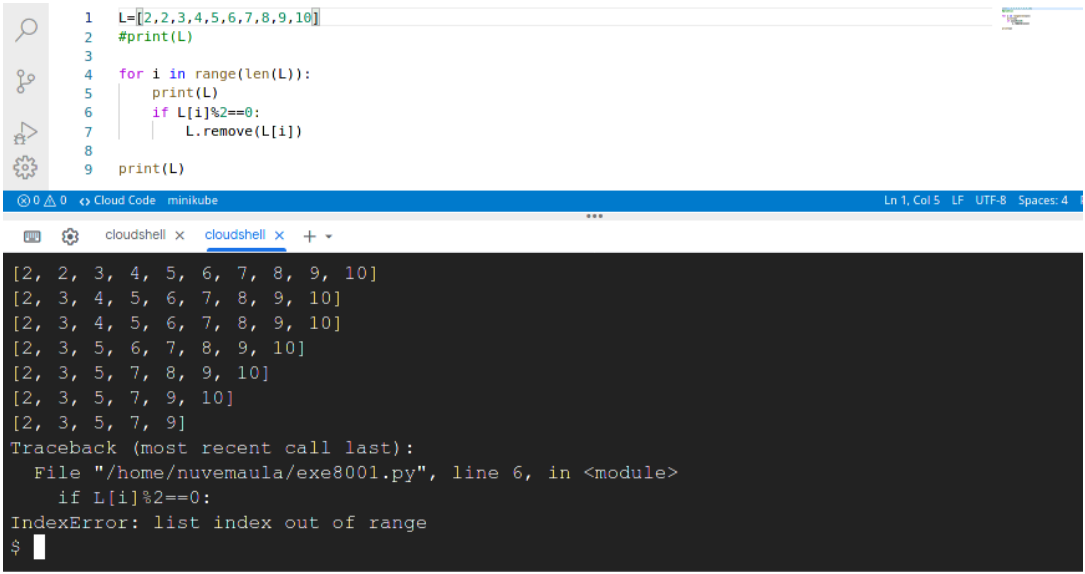
```python
L=[1,2,3,4,5,6,7,8,9,10]
print(L)

for i in range(len(L)):
    if L[i]%2==0:
        L.remove(i)

print(L)
```

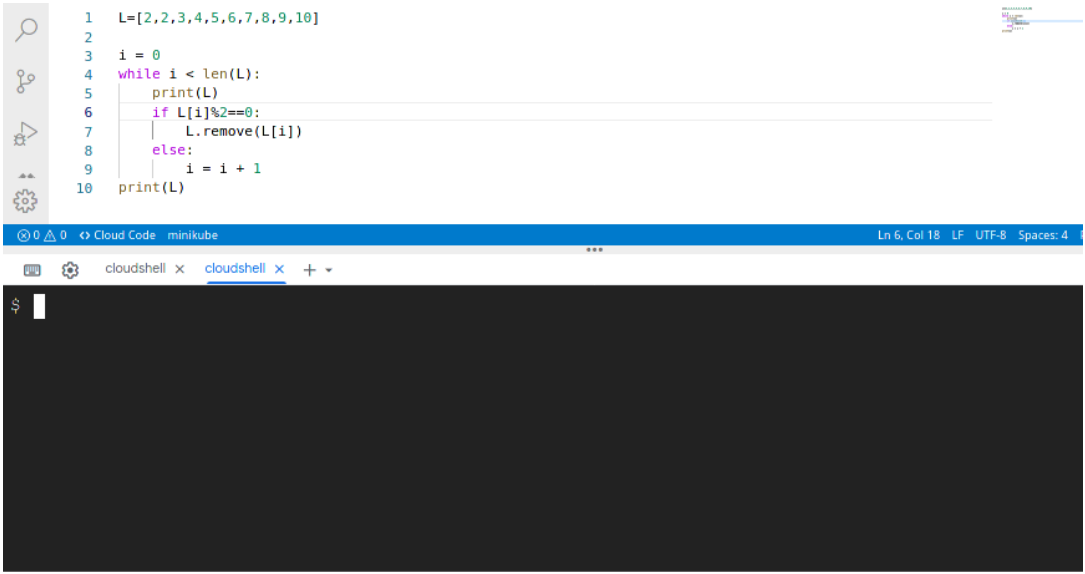


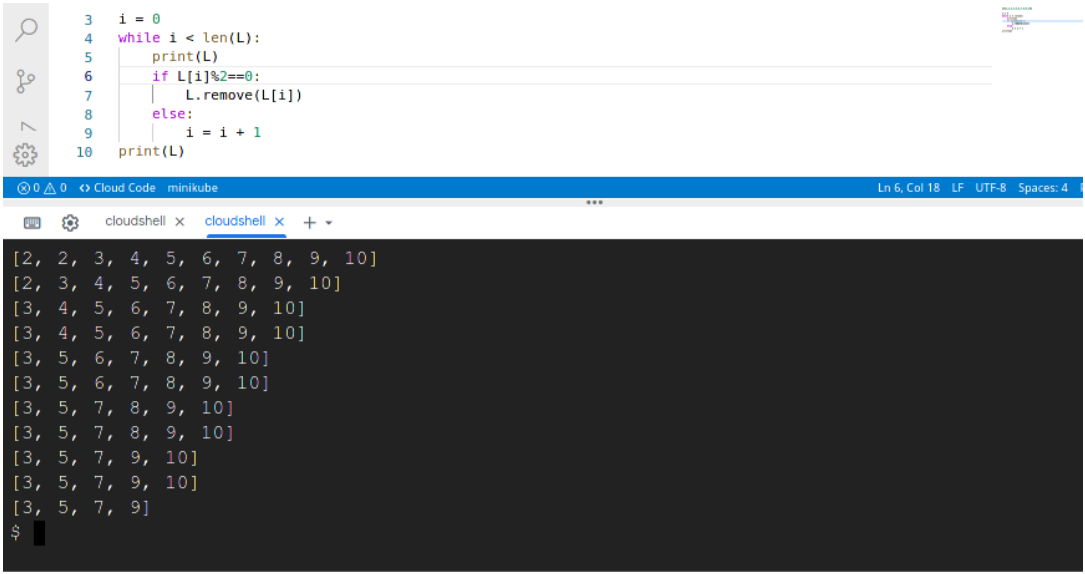

1 L=[1,2,3,4,5,6,7,8,9,10]
2 print(L)
3
4 for i in range(len(L)):
5 print(L)
6 if L[i]%2==0:
7 L.remove(i)
8
9 print(L)
0 0 Cloud Code minikube
Ln 4, Col 24 LF UTF-8 Spaces: 4
cloudshell cloudshell
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
[2, 3, 4, 5, 6, 7, 8, 9, 10]
[3, 4, 5, 6, 7, 8, 9, 10]
[4, 5, 6, 7, 8, 9, 10]
[5, 6, 7, 8, 9, 10]
[6, 7, 8, 9, 10]
Traceback (most recent call last):
File "/home/nuvemaula/exe8001.py", line 6, in <module>
if L[i]%2==0:
IndexError: list index out of range
$

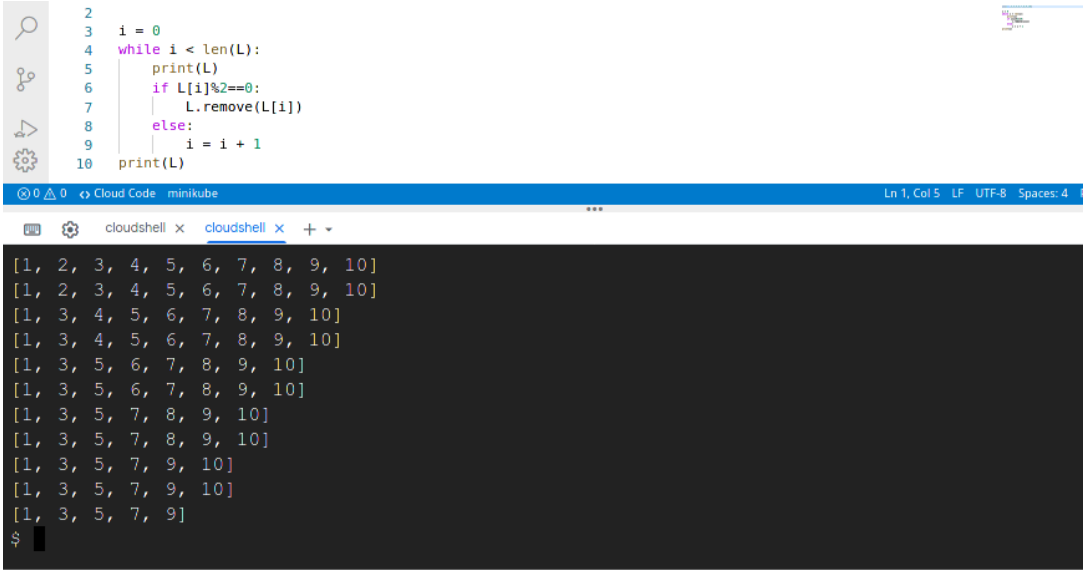

1 L=[1,2,3,4,5,6,7,8,9,10]
2 #print(L)
3
4 for i in range(len(L)):
5 print(L)
6 if L[i]%2==0:
7 L.remove(L[i])
8
9 print(L)
0 0 Cloud Code minikube
Ln 7, Col 22 LF UTF-8 Spaces: 4
cloudshell
cloudshell
$

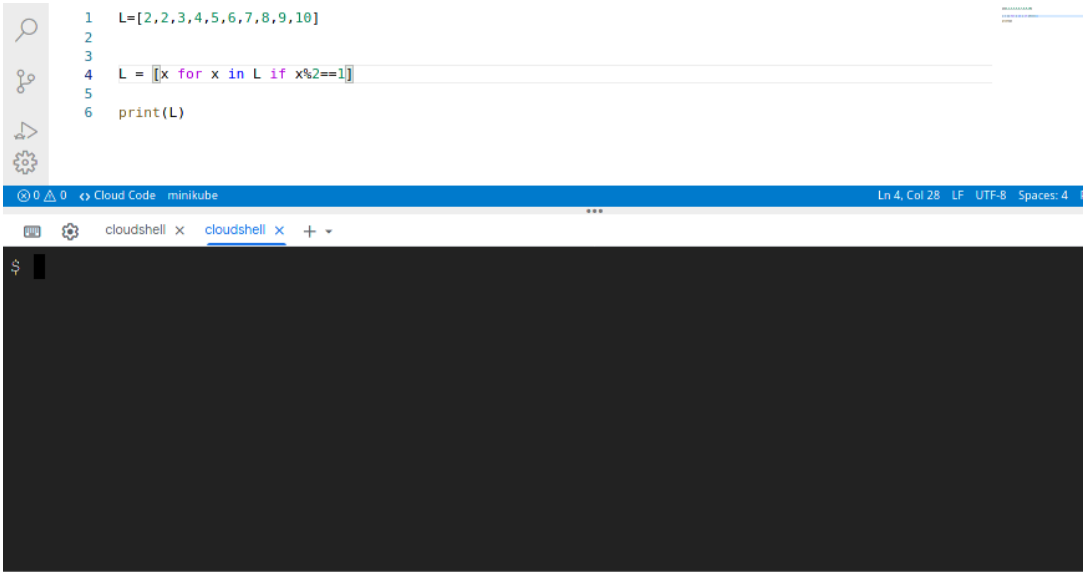

1 L=[1,2,3,4,5,6,7,8,9,10]
2 #print(L)
3
4 for i in range(len(L)):
5 print(L)
6 if L[i]%2==0:
7 L.remove(L[i])
8
9 print(L)
0 0 Cloud Code minikube
Ln 7, Col 22 LF UTF-8 Spaces: 4
cloudshell cloudshell
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
[1, 3, 4, 5, 6, 7, 8, 9, 10]
[1, 3, 5, 6, 7, 8, 9, 10]
[1, 3, 5, 7, 8, 9, 10]
[1, 3, 5, 7, 9, 10]
[1, 3, 5, 7, 9]
Traceback (most recent call last):
File "/home/nuvemaula/exe8001.py", line 6, in <module>
if L[i]%2==0:
IndexError: list index out of range
$

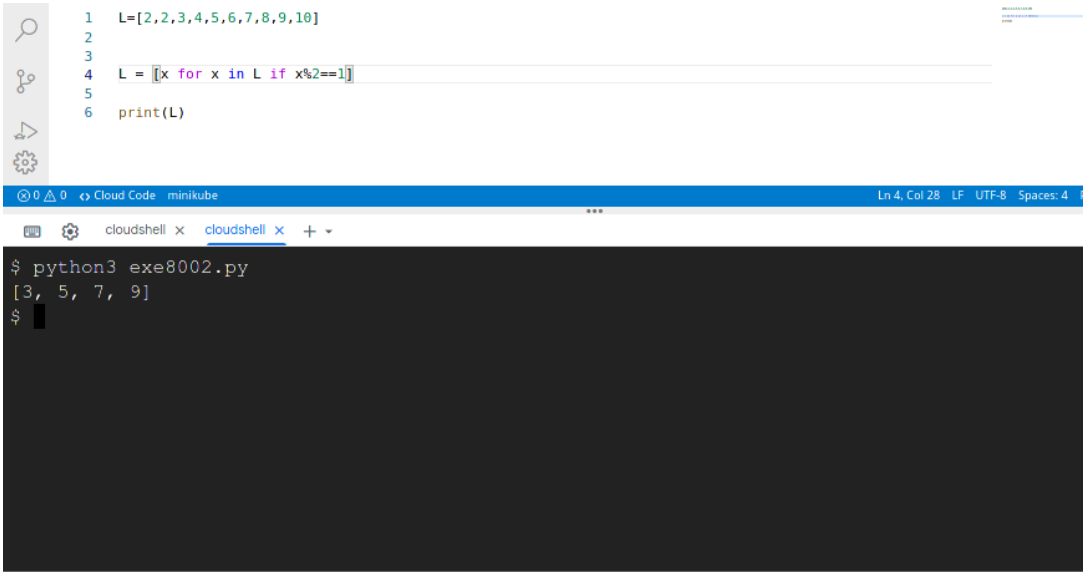

L=[2,2,3,4,5,6,7,8,9,10]
#print(L)
for i in range(len(L)):
print(L)
if L[i]%2==0:
L.remove(L[i])
print(L)
0 0 Cloud Code minikube
Ln 1, Col 5 LF UTF-8 Spaces: 4
cloudshell
cloudshell
[2, 2, 3, 4, 5, 6, 7, 8, 9, 10]
[2, 3, 4, 5, 6, 7, 8, 9, 10]
[2, 3, 4, 5, 6, 7, 8, 9, 10]
[2, 3, 5, 6, 7, 8, 9, 10]
[2, 3, 5, 7, 8, 9, 10]
[2, 3, 5, 7, 9, 10]
[2, 3, 5, 7, 9]
Traceback (most recent call last):
File "/home/nuvemaula/exe8001.py", line 6, in <module>
if L[i]%2==0:
IndexError: list index out of range
$

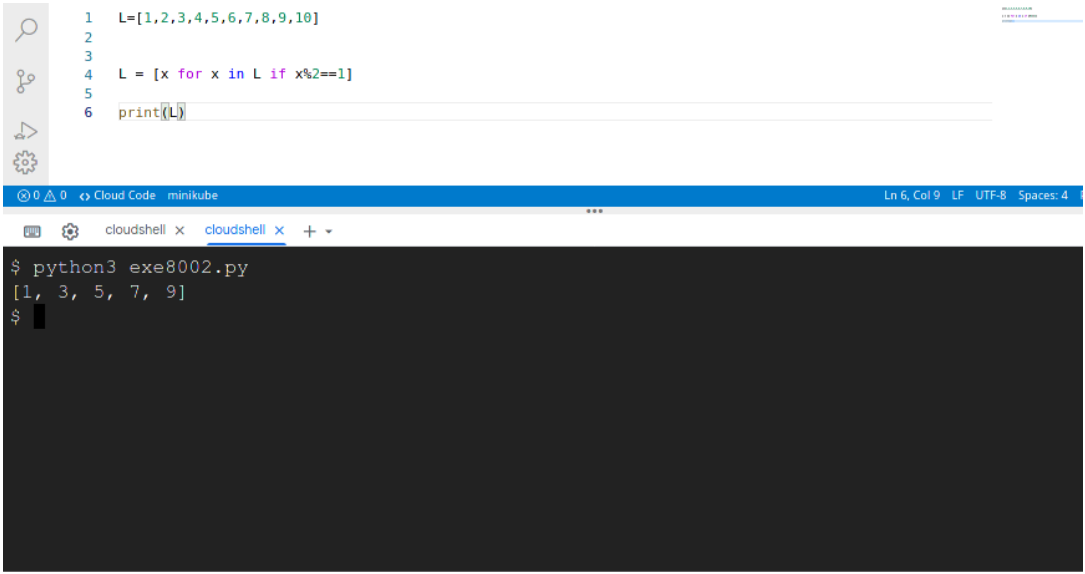

1 L=[2,2,3,4,5,6,7,8,9,10]
2
3 i = 0
4 while i < len(L):
5 print(L)
6 if L[i]%2==0:
7 L.remove(L[i])
8 else:
9 i = i + 1
10 print(L)
⊗ 0 △ 0 Cloud Code minikube
Ln 6, Col 18 LF UTF-8 Spaces: 4
cloudshell
cloudshell
$


3 i = 0
4 while i < len(L):
5 print(L)
6 if L[i]%2==0:
7 L.remove(L[i])
8 else:
9 i = i + 1
10 print(L)
⊗0 △0 Cloud Code minikube
Ln 6, Col 18 LF UTF-8 Spaces: 4
cloudshell × cloudshell ×
[2, 2, 3, 4, 5, 6, 7, 8, 9, 10]
[2, 3, 4, 5, 6, 7, 8, 9, 10]
[3, 4, 5, 6, 7, 8, 9, 10]
[3, 4, 5, 6, 7, 8, 9, 10]
[3, 5, 6, 7, 8, 9, 10]
[3, 5, 6, 7, 8, 9, 10]
[3, 5, 7, 8, 9, 10]
[3, 5, 7, 8, 9, 10]
[3, 5, 7, 9, 10]
[3, 5, 7, 9, 10]
[3, 5, 7, 9]
$


2
3 i = 0
4 while i < len(L):
5 print(L)
6 if L[i]%2==0:
7 L.remove(L[i])
8 else:
9 i = i + 1
10 print(L)
0 0 Cloud Code minikube
Ln 1, Col 5 LF UTF-8 Spaces: 4
cloudshell
cloudshell
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
[1, 3, 4, 5, 6, 7, 8, 9, 10]
[1, 3, 4, 5, 6, 7, 8, 9, 10]
[1, 3, 5, 6, 7, 8, 9, 10]
[1, 3, 5, 6, 7, 8, 9, 10]
[1, 3, 5, 7, 8, 9, 10]
[1, 3, 5, 7, 8, 9, 10]
[1, 3, 5, 7, 9, 10]
[1, 3, 5, 7, 9, 10]
[1, 3, 5, 7, 9]
$


1 L=[2,2,3,4,5,6,7,8,9,10]
2
3
4 L = [x for x in L if x%2==1]
5
6 print(L)
0 0 Cloud Code minikube
Ln 4, Col 28 LF UTF-8 Spaces: 4
cloudshell
cloudshell
$


1 L=[2,2,3,4,5,6,7,8,9,10]
2
3
4 L = [x for x in L if x%2==1]
5
6 print(L)
0 0 Cloud Code minikube
Ln 4, Col 28 LF UTF-8 Spaces: 4
cloudshell
cloudshell
$ python3 exe8002.py
[3, 5, 7, 9]
$


1 L=[1,2,3,4,5,6,7,8,9,10]
2
3
4 L = [x for x in L if x%2==1]
5
6 print(L)
0 0 Cloud Code minikube
Ln 6, Col 9 LF UTF-8 Spaces: 4
cloudshell
cloudshell
$ python3 exe8002.py
[1, 3, 5, 7, 9]
$

# Exercício 2

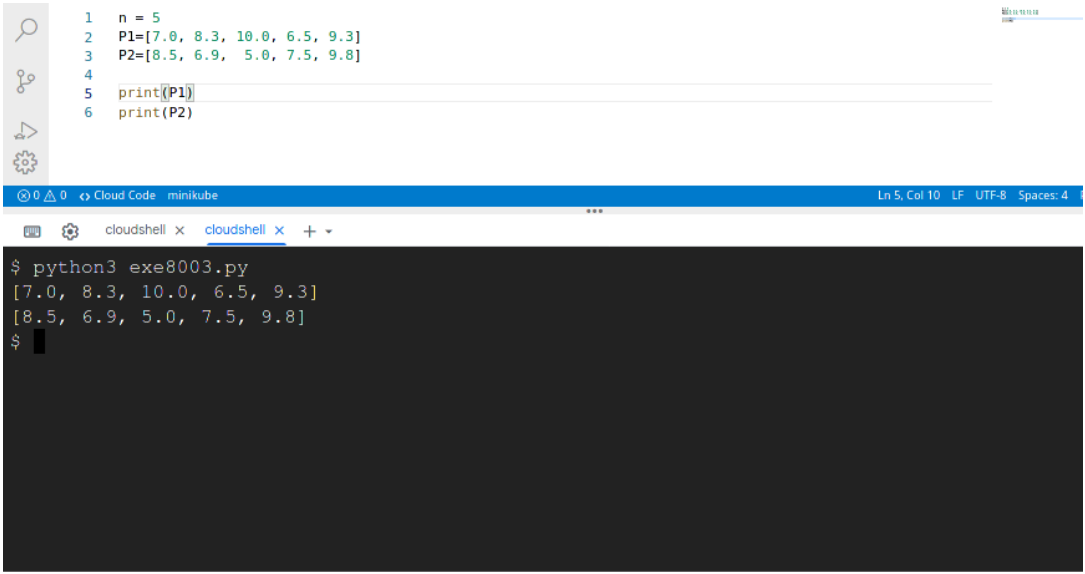
https://ic.unicamp.br/~mc102/aulas/aula05.pdf

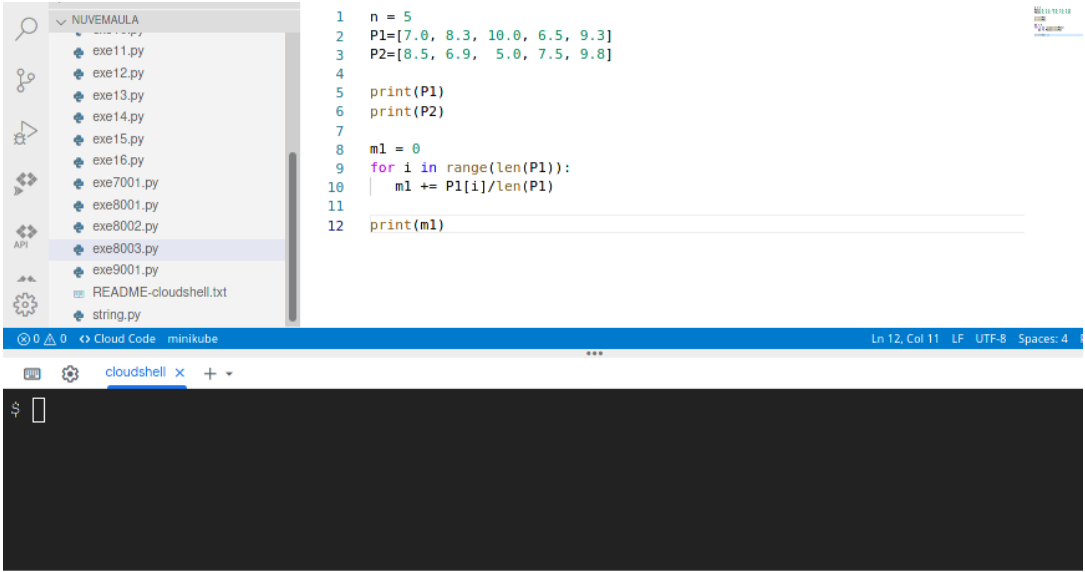
## Descrição

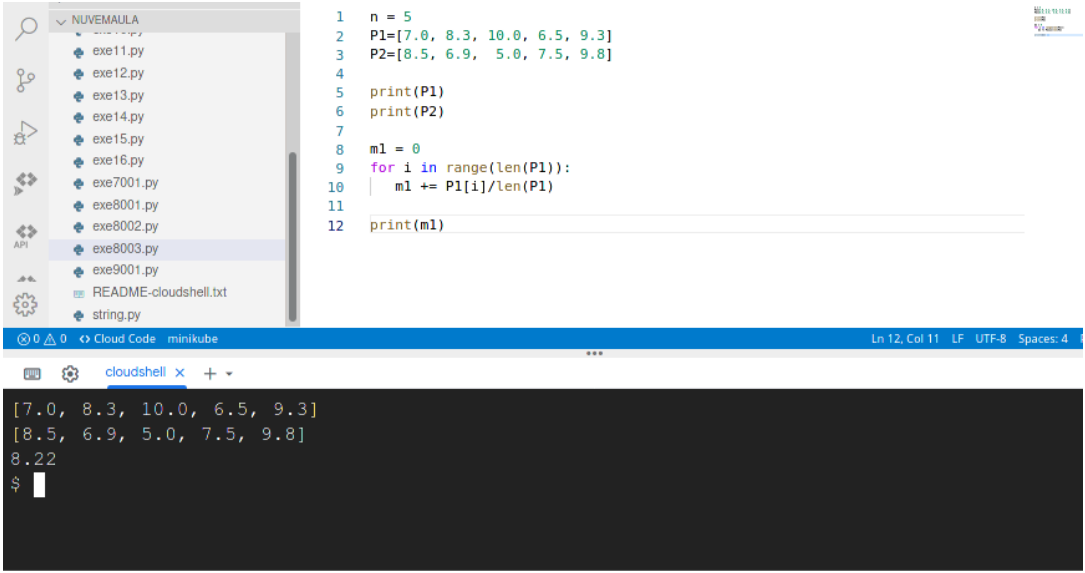
Dadas duas listas $P1$ e $P2$, ambas com $n$ valores reais que representam as notas de uma turma na prova 1 e na prova 2, respectivamente, escreva um programa que calcule a média da turma nas provas 1 e 2, imprimindo em qual das provas a turma obteve a melhor média.

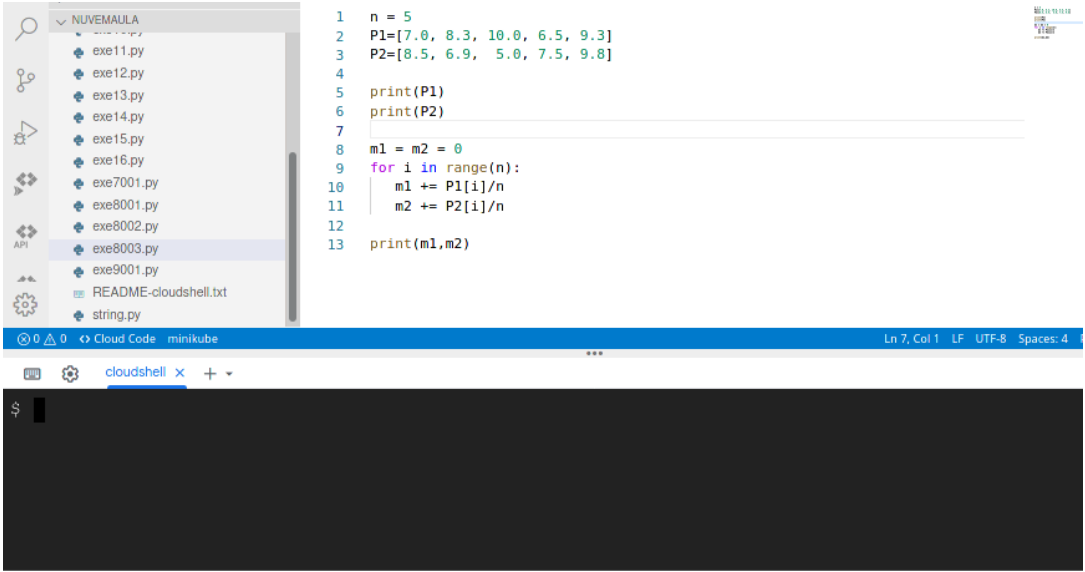
Exemplo:
Tamanho da turma: 5
$P1$: 7.0 8.3 10.0 6.5 9.3
$P2$: 8.5 6.9 5.0 7.5 9.8

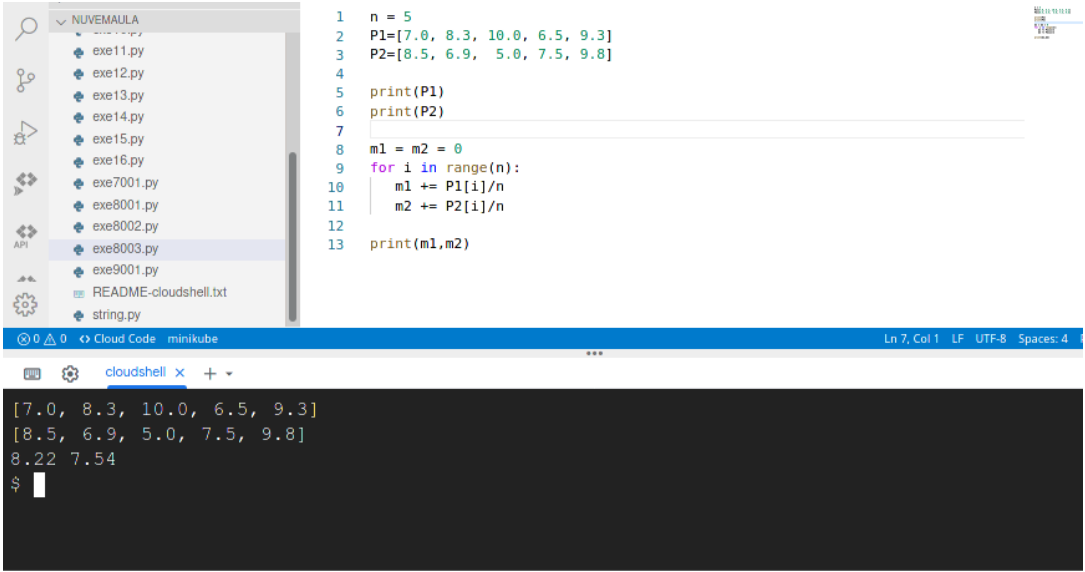
Resposta:
Média da turma na prova 1: 8.22
Média da turma na prova 2: 7.54
A turma obteve a melhor média na prova 1.

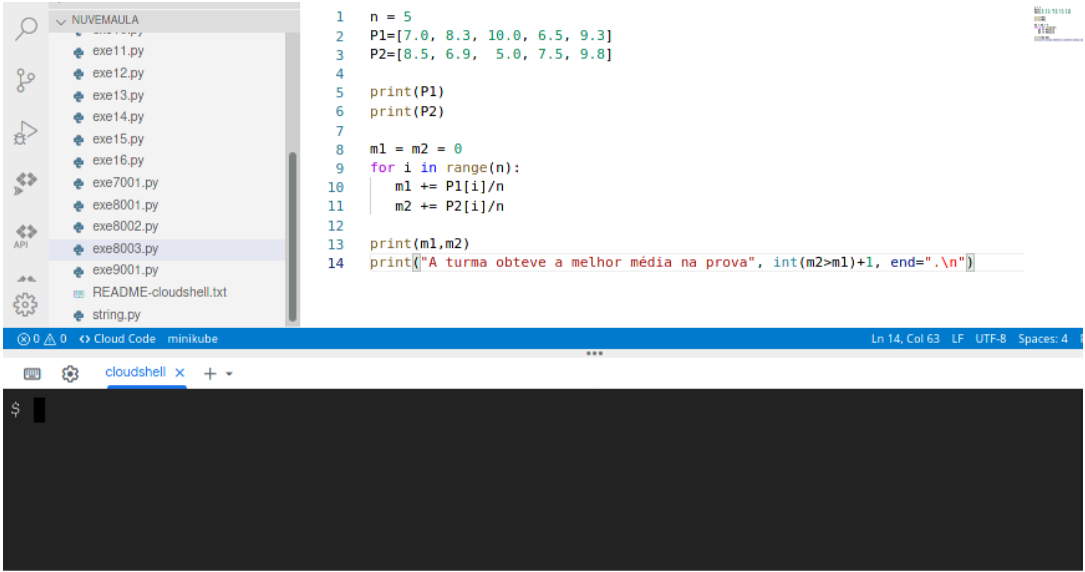

1 n = 5
2 P1=[7.0, 8.3, 10.0, 6.5, 9.3]
3 P2=[8.5, 6.9, 5.0, 7.5, 9.8]
4
5 print(P1)
6 print(P2)
0 0 Cloud Code minikube
Ln 5, Col 10 LF UTF-8 Spaces: 4
cloudshell
cloudshell
$ python3 exe8003.py
[7.0, 8.3, 10.0, 6.5, 9.3]
[8.5, 6.9, 5.0, 7.5, 9.8]
$

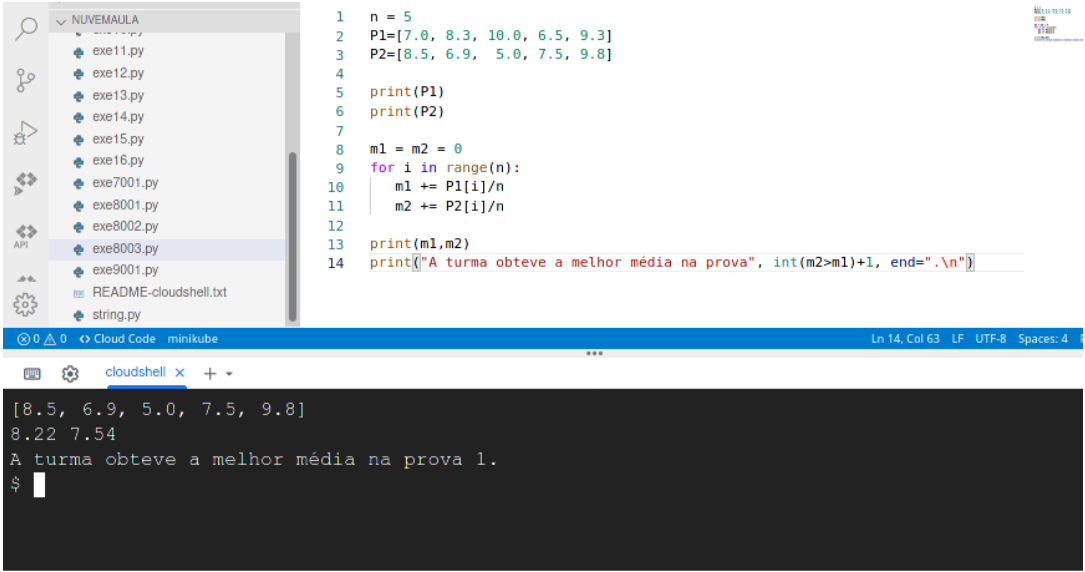

NUVEMAULA
exe11.py
exe12.py
exe13.py
exe14.py
exe15.py
exe16.py
exe7001.py
exe8001.py
exe8002.py
exe8003.py
exe9001.py
README-cloudshell.txt
string.py
n = 5
P1=[7.0, 8.3, 10.0, 6.5, 9.3]
P2=[8.5, 6.9, 5.0, 7.5, 9.8]
print(P1)
print(P2)
m1 = 0
for i in range(len(P1)):
m1 += P1[i]/len(P1)
print(m1)
0 0 Cloud Code minikube
Ln 12, Col 11 LF UTF-8 Spaces: 4
cloudshell
$

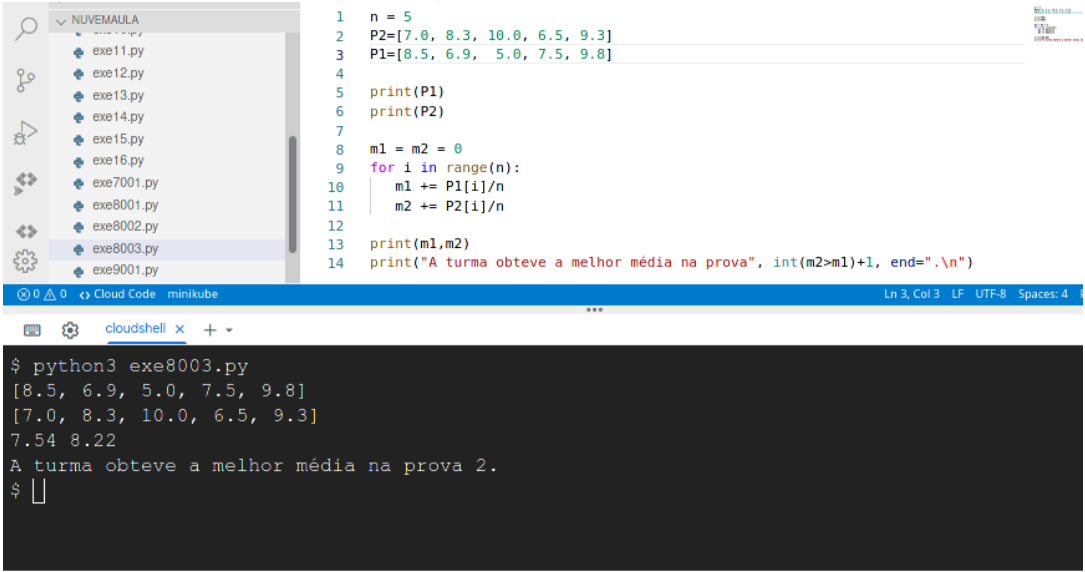

NUVEMAULA
exe11.py
exe12.py
exe13.py
exe14.py
exe15.py
exe16.py
exe7001.py
exe8001.py
exe8002.py
exe8003.py
exe9001.py
README-cloudshell.txt
string.py
API
1 n = 5
2 P1=[7.0, 8.3, 10.0, 6.5, 9.3]
3 P2=[8.5, 6.9, 5.0, 7.5, 9.8]
4
5 print(P1)
6 print(P2)
7
8 m1 = 0
9 for i in range(len(P1)):
10 m1 += P1[i]/len(P1)
11
12 print(m1)
0 0 Cloud Code minikube
Ln 12, Col 11 LF UTF-8 Spaces: 4
cloudshell
[7.0, 8.3, 10.0, 6.5, 9.3]
[8.5, 6.9, 5.0, 7.5, 9.8]
8.22
$


NUVEMAULA
exe11.py
exe12.py
exe13.py
exe14.py
exe15.py
exe16.py
exe7001.py
exe8001.py
exe8002.py
exe8003.py
exe9001.py
README-cloudshell.txt
string.py
API
1 n = 5
2 P1=[7.0, 8.3, 10.0, 6.5, 9.3]
3 P2=[8.5, 6.9, 5.0, 7.5, 9.8]
4
5 print(P1)
6 print(P2)
7
8 m1 = m2 = 0
9 for i in range(n):
10 m1 += P1[i]/n
11 m2 += P2[i]/n
12
13 print(m1,m2)
0 0 Cloud Code minikube
Ln 7, Col 1 LF UTF-8 Spaces: 4
cloudshell
$


NUVEMAULA
exe11.py
exe12.py
exe13.py
exe14.py
exe15.py
exe16.py
exe7001.py
exe8001.py
exe8002.py
exe8003.py
exe9001.py
README-cloudshell.txt
string.py
API
1 n = 5
2 P1=[7.0, 8.3, 10.0, 6.5, 9.3]
3 P2=[8.5, 6.9, 5.0, 7.5, 9.8]
4
5 print(P1)
6 print(P2)
7
8 m1 = m2 = 0
9 for i in range(n):
10 m1 += P1[i]/n
11 m2 += P2[i]/n
12
13 print(m1,m2)
0 0 Cloud Code minikube
Ln 7, Col 1 LF UTF-8 Spaces: 4
cloudshell
[7.0, 8.3, 10.0, 6.5, 9.3]
[8.5, 6.9, 5.0, 7.5, 9.8]
8.22 7.54
$


NUVEMAULA
exe11.py
exe12.py
exe13.py
exe14.py
exe15.py
exe16.py
exe7001.py
exe8001.py
exe8002.py
exe8003.py
exe9001.py
README-cloudshell.txt
string.py
n = 5
P1=[7.0, 8.3, 10.0, 6.5, 9.3]
P2=[8.5, 6.9, 5.0, 7.5, 9.8]
print(P1)
print(P2)
m1 = m2 = 0
for i in range(n):
m1 += P1[i]/n
m2 += P2[i]/n
print(m1,m2)
print("A turma obteve a melhor média na prova", int(m2>m1)+1, end=".\n")
Cloud Code minikube
Ln 14, Col 63 LF UTF-8 Spaces: 4
cloudshell
$


NUVEMAULA
exe11.py
exe12.py
exe13.py
exe14.py
exe15.py
exe16.py
exe7001.py
exe8001.py
exe8002.py
exe8003.py
exe9001.py
README-cloudshell.txt
string.py
1 n = 5
2 P1=[7.0, 8.3, 10.0, 6.5, 9.3]
3 P2=[8.5, 6.9, 5.0, 7.5, 9.8]
4
5 print(P1)
6 print(P2)
7
8 m1 = m2 = 0
9 for i in range(n):
10 m1 += P1[i]/n
11 m2 += P2[i]/n
12
13 print(m1,m2)
14 print("A turma obteve a melhor média na prova", int(m2>m1)+1, end=".\n")
Cloud Code minikube
Ln 14, Col 63 LF UTF-8 Spaces: 4
cloudshell
[8.5, 6.9, 5.0, 7.5, 9.8]
8.22 7.54
A turma obteve a melhor média na prova 1.
$


NUVEMAULA
exe11.py
exe12.py
exe13.py
exe14.py
exe15.py
exe16.py
exe7001.py
exe8001.py
exe8002.py
exe8003.py
exe9001.py
n = 5
P2=[7.0, 8.3, 10.0, 6.5, 9.3]
P1=[8.5, 6.9, 5.0, 7.5, 9.8]
print(P1)
print(P2)
m1 = m2 = 0
for i in range(n):
m1 += P1[i]/n
m2 += P2[i]/n
print(m1,m2)
print("A turma obteve a melhor média na prova", int(m2>m1)+1, end=".\n")
Cloud Code minikube
Ln 3, Col 3 LF UTF-8 Spaces: 4
cloudshell
$ python3 exe8003.py
[8.5, 6.9, 5.0, 7.5, 9.8]
[7.0, 8.3, 10.0, 6.5, 9.3]
7.54 8.22
A turma obteve a melhor média na prova 2.
$

# Exercício 3

https://ic.unicamp.br/~mc102/aulas/aula05.pdf

## Descrição

Dadas duas listas $L1$ e $L2$, com $n$ e $m$ valores inteiros, respectivamente, escreva um programa que concatene as listas $L1$ e $L2$ em uma nova lista $L3$. Em seguida, imprima a lista $L3$ ordenada de maneira crescente e decrescente.

Exemplo:
Tamanho da lista $L1$: 3
Tamanho da lista $L2$: 4
$L1$: 7 2 9
$L2$: 2 5 1 3

Resposta:
1 2 2 3 5 7 9
9 7 5 3 2 2 1


1 L1 = [7, 2, 9, ]
2 L2 = [2, 5, 1, 3, ]
3
4 print(L1)
5 print(L2)
6
⊗0 △0 Cloud Code minikube
Ln 6, Col 1 LF UTF-8 Spaces: 4
cloudshell
$ python3 exe8004.py
[7, 2, 9]
[2, 5, 1, 3]
$


1 L1 = [7, 2, 9, ]
2 L2 = [2, 5, 1, 3, ]
3
4 print(L1)
5 print(L2)
6
7 L3 = L1 + L2
8
9 print(sorted(L3))
10 print(L3.sort(reverse = True))
11
⊗0 △0 Cloud Code minikube
Ln 11, Col 1 LF UTF-8 Spaces: 4
cloudshell
$ python3 exe8004.py
[7, 2, 9]
[2, 5, 1, 3]
[1, 2, 2, 3, 5, 7, 9]
None
$


1 L1 = [7, 2, 9, ]
2 L2 = [2, 5, 1, 3, ]
3
4 print(L1)
5 print(L2)
6
7 L3 = L1 + L2
8
9 print(sorted(L3))
10 L3.sort(reverse = True)
11
12 print(L3)
13
⊗0 △0 Cloud Code minikube
Ln 13, Col 1 LF UTF-8 Spaces: 4
cloudshell
$ python3 exe8004.py
[7, 2, 9]
[2, 5, 1, 3]
[1, 2, 2, 3, 5, 7, 9]
[9, 7, 5, 3, 2, 2, 1]
$

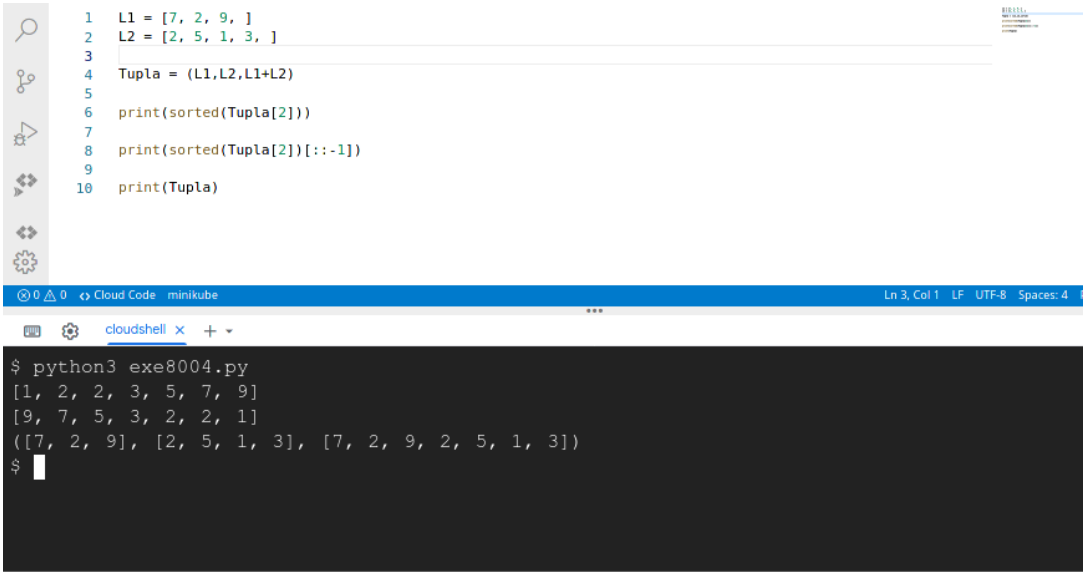
```python
L1 = [7, 2, 9, ]
L2 = [2, 5, 1, 3, ]

print(L1)
print(L2)

L3 = L1 + L2

print(sorted(L3))

print(sorted(L3)[::-1])
```

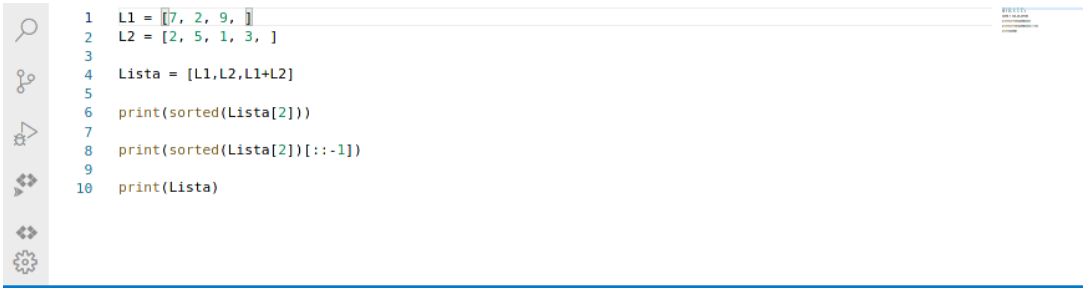


```
$ python3 exe8004.py
[7, 2, 9]
[2, 5, 1, 3]
[1, 2, 2, 3, 5, 7, 9]
[9, 7, 5, 3, 2, 2, 1]
$
```

```python
L1 = [7, 2, 9, ]
L2 = [2, 5, 1, 3, ]

print(L1)
print(L2)

L3 = L1 + L2

print(sorted(L3))

print(sorted(L3)[::-1])

print(L3)
```

```
$ python3 exe8004.py
[7, 2, 9]
[2, 5, 1, 3]
[1, 2, 2, 3, 5, 7, 9]
[9, 7, 5, 3, 2, 2, 1]
[7, 2, 9, 2, 5, 1, 3]
$
```


1 L1 = [7, 2, 9, ]
2 L2 = [2, 5, 1, 3, ]
3
4 Tupla = (L1,L2,L1+L2)
5
6 print(sorted(Tupla[2]))
7
8 print(sorted(Tupla[2])[::-1])
9
10 print(Tupla)
⊗0 △0 Cloud Code minikube
Ln 3, Col 1 LF UTF-8 Spaces: 4
cloudshell
$ python3 exe8004.py
[1, 2, 2, 3, 5, 7, 9]
[9, 7, 5, 3, 2, 2, 1]
([7, 2, 9], [2, 5, 1, 3], [7, 2, 9, 2, 5, 1, 3])
$



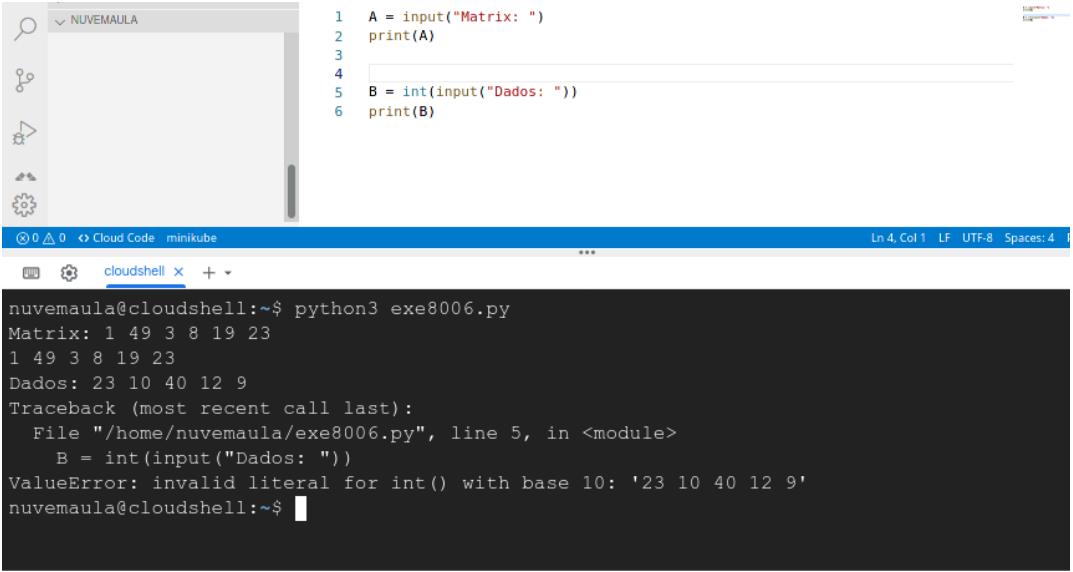
```
1   L1 = [7, 2, 9, ]
2   L2 = [2, 5, 1, 3, ]
3
4   Lista = [L1,L2,L1+L2]
5
6   print(sorted(Lista[2]))
7
8   print(sorted(Lista[2])[::-1])
9
10  print(Lista)
```

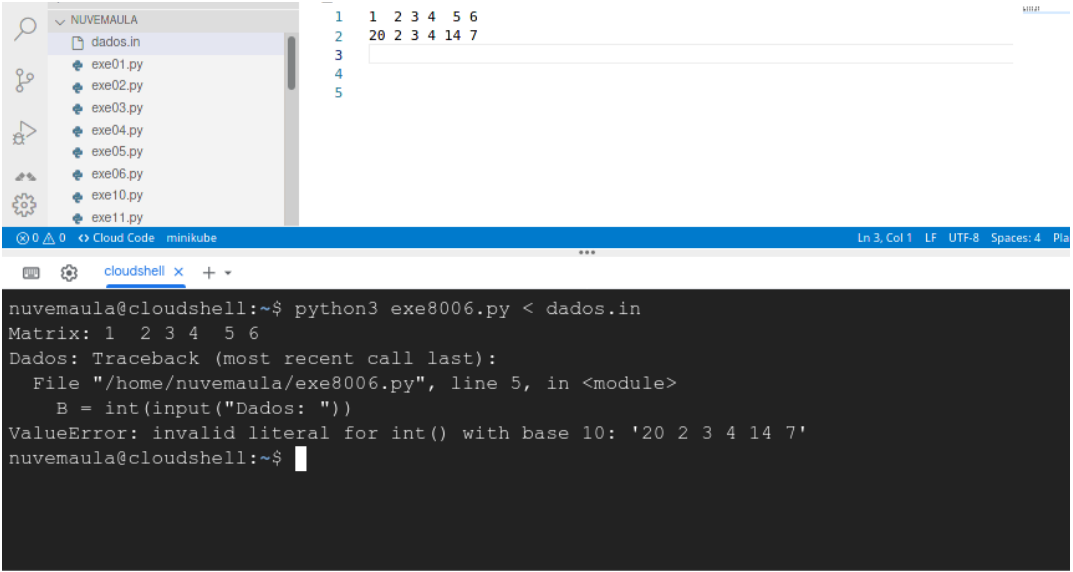
```
⊗0 △0  Cloud Code  minikube                    Ln 1, Col 17  LF  UTF-8  Spaces: 4
```

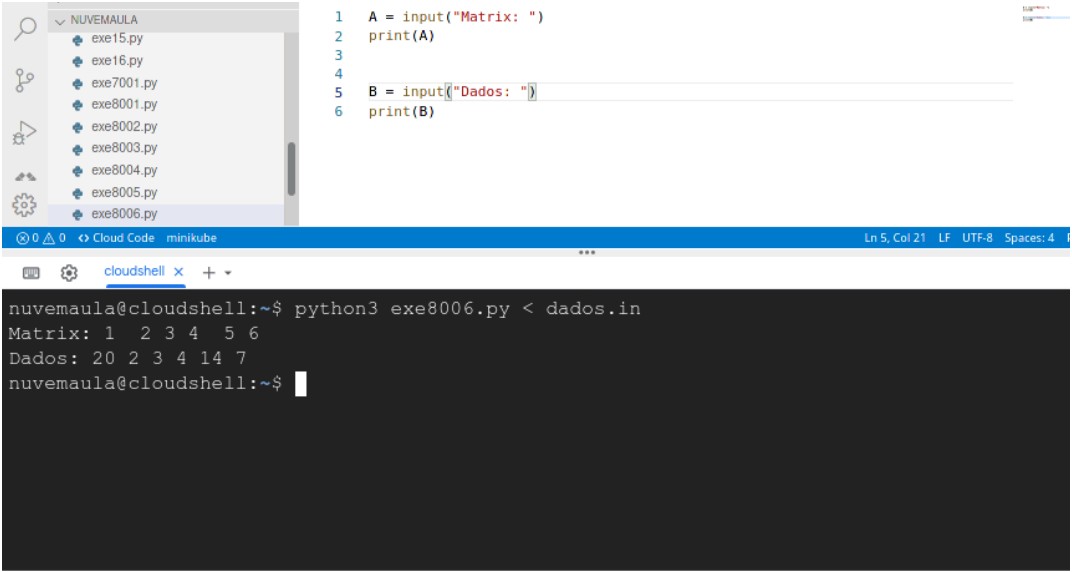
```
cloudshell ×
$ python3 exe8004.py
[1, 2, 2, 3, 5, 7, 9]
[9, 7, 5, 3, 2, 2, 1]
[[7, 2, 9], [2, 5, 1, 3], [7, 2, 9, 2, 5, 1, 3]]
$
```

# Exercício 4

## Leitura de uma matriz

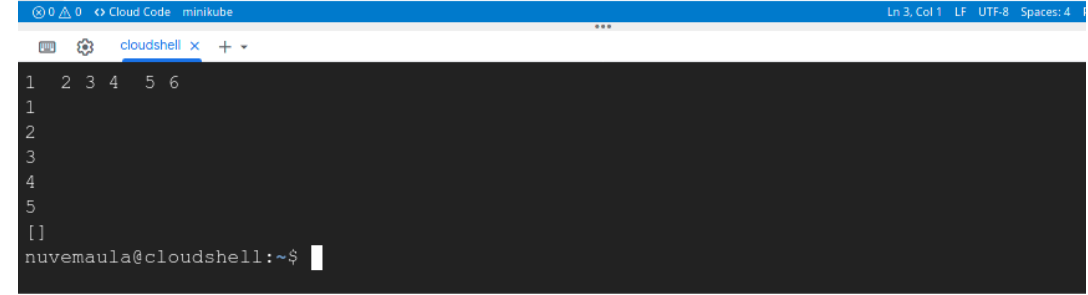

NUVEMAULA
1 A = input("Matrix: ")
2 print(A)
3
4
5 B = int(input("Dados: "))
6 print(B)
0 0 Cloud Code minikube
Ln 4, Col 1 LF UTF-8 Spaces: 4
cloudshell
nuvemaula@cloudshell:~$ python3 exe8006.py
Matrix: 1 49 3 8 19 23
1 49 3 8 19 23
Dados: 23 10 40 12 9
Traceback (most recent call last):
File "/home/nuvemaula/exe8006.py", line 5, in <module>
B = int(input("Dados: "))
ValueError: invalid literal for int() with base 10: '23 10 40 12 9'
nuvemaula@cloudshell:~$


NUVEMAULA
dados.in
exe01.py
exe02.py
exe03.py
exe04.py
exe05.py
exe06.py
exe10.py
exe11.py
1 2 3 4 5 6
20 2 3 4 14 7
Cloud Code minikube
Ln 3, Col 1 LF UTF-8 Spaces: 4
cloudshell
nuvemaula@cloudshell:~$ python3 exe8006.py < dados.in
Matrix: 1 2 3 4 5 6
Dados: Traceback (most recent call last):
File "/home/nuvemaula/exe8006.py", line 5, in <module>
B = int(input("Dados: "))
ValueError: invalid literal for int() with base 10: '20 2 3 4 14 7'
nuvemaula@cloudshell:~$


NUVEMAULA
exe15.py
exe16.py
exe7001.py
exe8001.py
exe8002.py
exe8003.py
exe8004.py
exe8005.py
exe8006.py
1 A = input("Matrix: ")
2 print(A)
3
4
5 B = input("Dados: ")
6 print(B)
0 0 Cloud Code minikube
Ln 5, Col 21 LF UTF-8 Spaces: 4
cloudshell
nuvemaula@cloudshell:~$ python3 exe8006.py < dados.in
Matrix: 1 2 3 4 5 6
Dados: 20 2 3 4 14 7
nuvemaula@cloudshell:~$


1 A = input()
2 print(A)
3
4 L = []
5 i=j=0
6 while j<len(A):
7 if A[i]==" ":
8 i = j = i+1
9 elif A[j]!=" ":
10 j = j+1
11 else:
12 print( A[i:j+1])
13 i = j
14 print(L)
0 0 Cloud Code minikube
Ln 3, Col 1 LF UTF-8 Spaces: 4
cloudshell
nuvemaula@cloudshell:~$

```python
A = input()
print(A)

L = []
i=j=0
while j<len(A):
    if A[i]==" ":
        i = j = i+1
    elif A[j]!=" ":
        j = j+1
    else:
        print( A[i:j+1])
        i = j
print(L)
```



```python
A = input()
print(A)

L = []
i=j=0
while j<=len(A):
    if A[i]==" ":
        i = j = i+1
    elif j<len(A) and A[j]!=" ":
        j = j+1
    else:
        print( A[i:j+1])
        i = j
print(L)
```

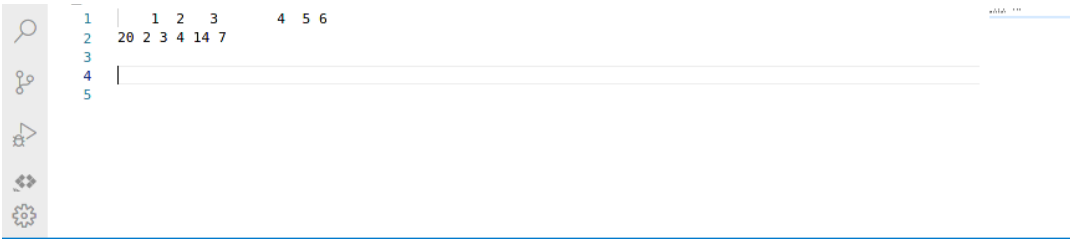
```
4
5
6
Traceback (most recent call last):
  File "/home/nuvemaula/exe8006.py", line 7, in <module>
    if A[i]==" ":
IndexError: string index out of range
nuvemaula@cloudshell:~$
```

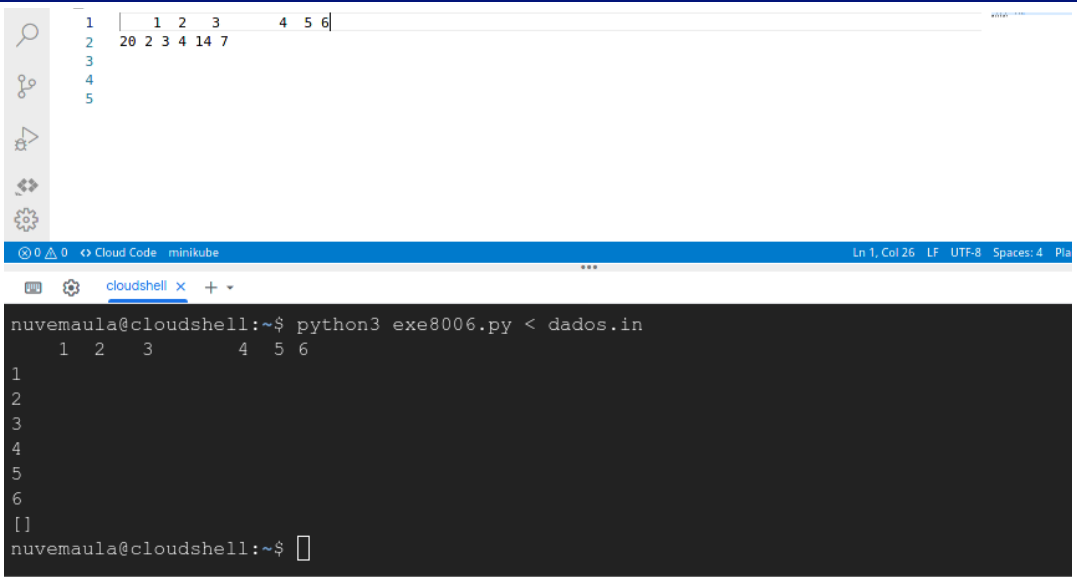
```
4   L = []
5   i=j=0
6   while j<=len(A):
7       if A[i]==" ":
8           i = j = i+1
9       elif j<len(A) and A[j]!=" ":
10          j = j+1
11      else:
12          print( A[i:j+1])
13          i = j = j+1
14  print(L)
```

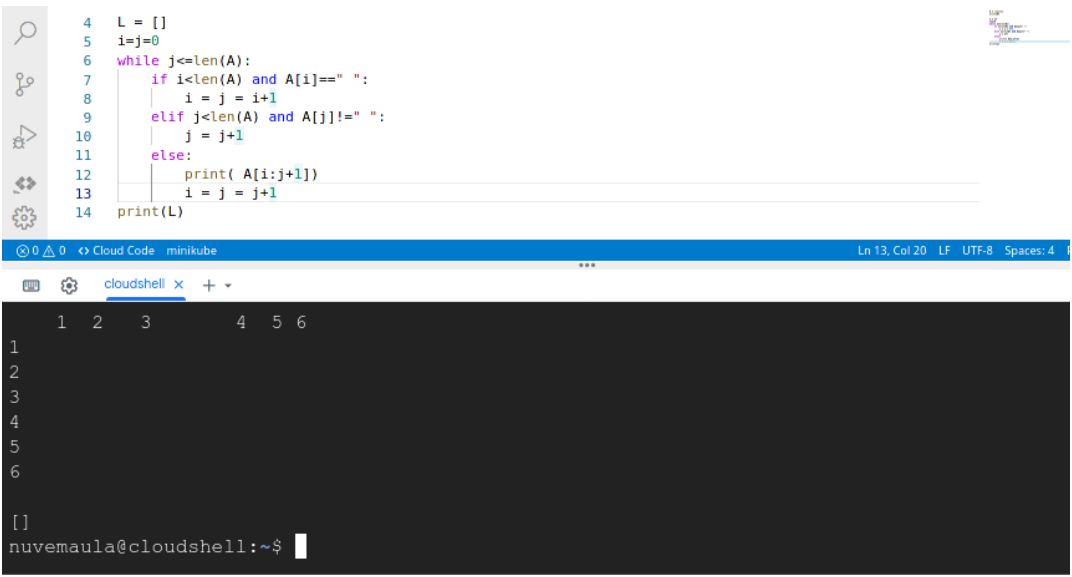
⊗0 △0 Cloud Code minikube Ln 11, Col 10 LF UTF-8 Spaces: 4

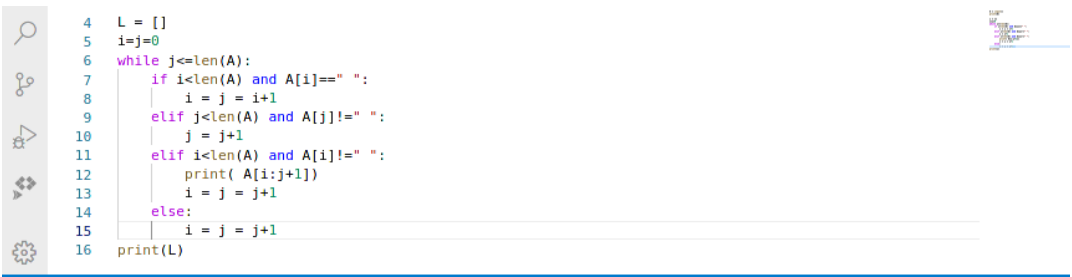
cloudshell

```
nuvemaula@cloudshell:~$ python3 exe8006.py < dados.in
1   2 3 4   5 6
1
2
3
4
5
6
[]
nuvemaula@cloudshell:~$
```


1 2 3 4 5 6
20 2 3 4 14 7
Cloud Code minikube
Ln 4, Col 1 LF UTF-8 Spaces: 4
cloudshell
2
3
4
5
6
Traceback (most recent call last):
File "/home/nuvemaula/exe8006.py", line 7, in <module>
if A[i]==" ":
IndexError: string index out of range
nuvemaula@cloudshell:~$


1 1 2 3 4 5 6
2 0 2 3 4 1 4 7
⊗ 0 △ 0 Cloud Code minikube
Ln 1, Col 26 LF UTF-8 Spaces: 4
cloudshell
nuvemaula@cloudshell:~$ python3 exe8006.py < dados.in
1 2 3 4 5 6
1
2
3
4
5
6
[]
nuvemaula@cloudshell:~$

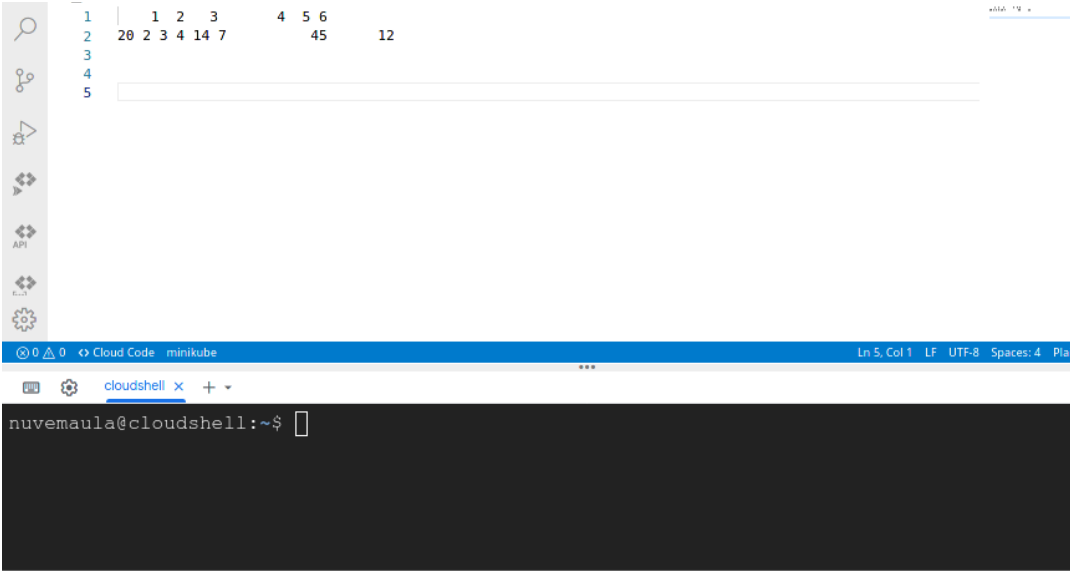

4 L = []
5 i=j=0
6 while j<=len(A):
7 if i<len(A) and A[i]==" ":
8 i = j = i+1
9 elif j<len(A) and A[j]!=" ":
10 j = j+1
11 else:
12 print( A[i:j+1])
13 i = j = j+1
14 print(L)
0 0 Cloud Code minikube
Ln 13, Col 20 LF UTF-8 Spaces: 4
cloudshell
1 2 3 4 5 6
1
2
3
4
5
6
[]
nuvemaula@cloudshell:~$

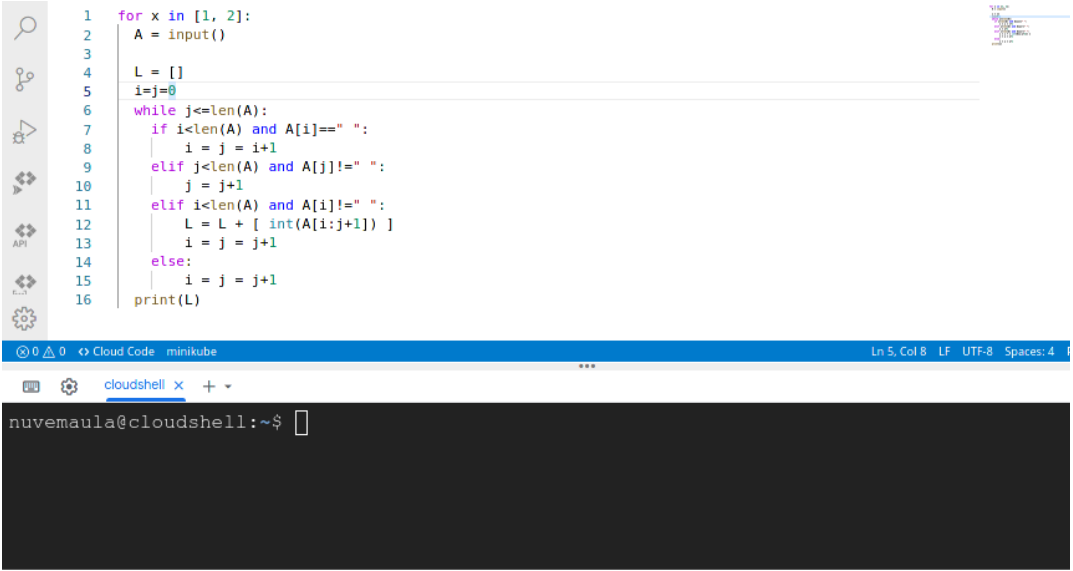

4 L = []
5 i=j=0
6 while j<=len(A):
7 if i<len(A) and A[i]==" ":
8 i = j = i+1
9 elif j<len(A) and A[j]!=" ":
10 j = j+1
11 elif i<len(A) and A[i]!=" ":
12 print( A[i:j+1])
13 i = j = j+1
14 else:
15 i = j = j+1
16 print(L)
0 0 Cloud Code minikube
Ln 15, Col 20 LF UTF-8 Spaces: 4
cloudshell
1
2
3
4
5
6
[]
nuvemaula@cloudshell:~$

```
4   L = []
5   i=j=0
6   while j<=len(A):
7       if i<len(A) and A[i]==" ":
8           i = j = i+1
9       elif j<len(A) and A[j]!=" ":
10          j = j+1
11      elif i<len(A) and A[i]!=" ":
12          L = L + [ int(A[i:j+1]) ]
13          i = j = j+1
14      else:
15          i = j = j+1
16  print(L)
```



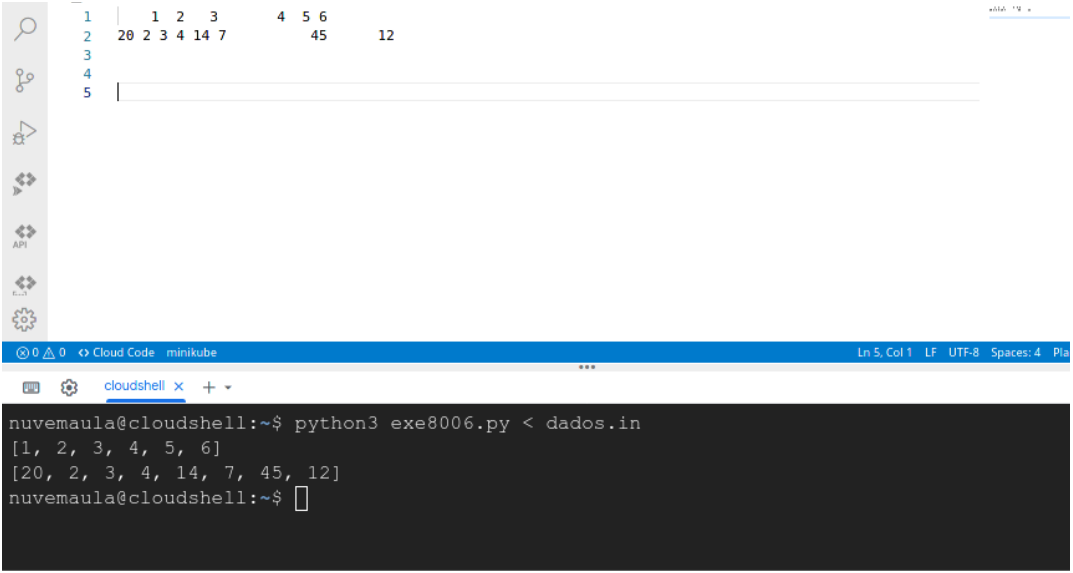
```
nuvemaula@cloudshell:~$ python3 exe8006.py < dados.in
    1  2   3        4  5 6
[1, 2, 3, 4, 5, 6]
nuvemaula@cloudshell:~$
```

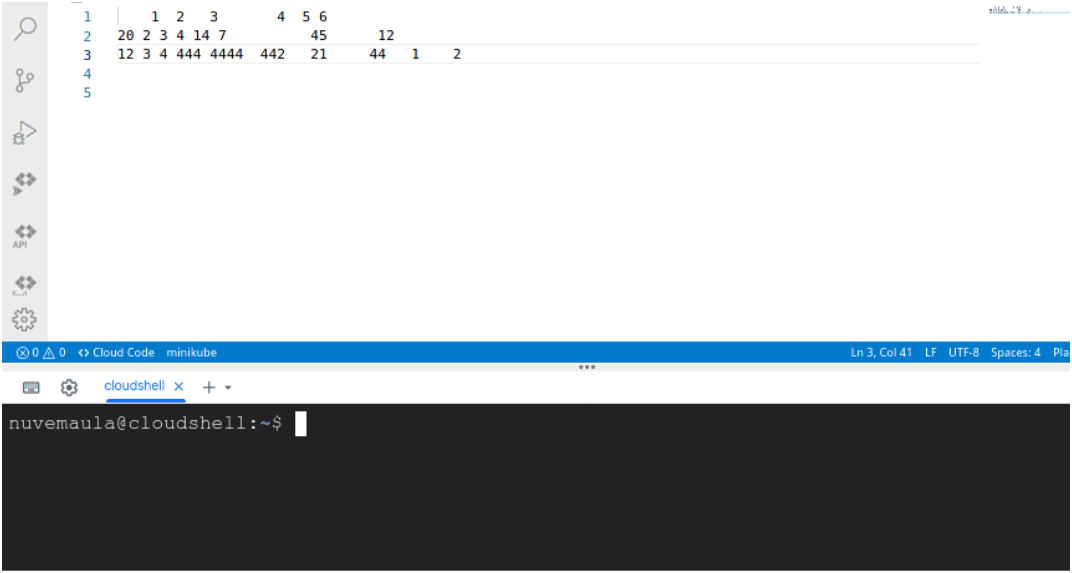

1 2 3 4 5 6
20 2 3 4 14 7 45 12
API
0 0 Cloud Code minikube
Ln 5, Col 1 LF UTF-8 Spaces: 4
cloudshell
nuvemaula@cloudshell:~$


for x in [1, 2]:
A = input()
L = []
i=j=0
while j<=len(A):
if i<len(A) and A[i]==" ":
i = j = i+1
elif j<len(A) and A[j]!=" ":
j = j+1
elif i<len(A) and A[i]!=" ":
L = L + [ int(A[i:j+1]) ]
i = j = j+1
else:
i = j = j+1
print(L)
Cloud Code
minikube
Ln 5, Col 8
LF
UTF-8
Spaces: 4
cloudshell
nuvemaula@cloudshell:~$

```
for x in [1, 2]:
  A = input()

  L = []
  i=j=0
  while j<=len(A):
    if i<len(A) and A[i]==" ":
        i = j = i+1
    elif j<len(A) and A[j]!=" ":
        j = j+1
    elif i<len(A) and A[i]!=" ":
        L = L + [ int(A[i:j+1]) ]
        i = j = j+1
    else:
        i = j = j+1
  print(L)
```

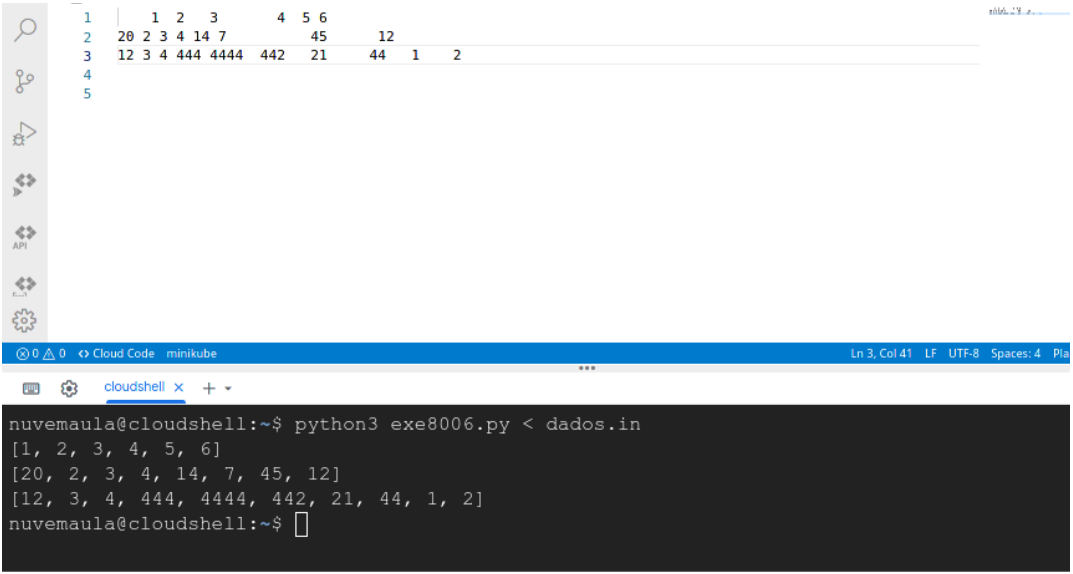
```
nuvemaula@cloudshell:~$ python3 exe8006.py < dados.in
[1, 2, 3, 4, 5, 6]
[20, 2, 3, 4, 14, 7, 45, 12]
nuvemaula@cloudshell:~$
```


1 2 3 4 5 6
20 2 3 4 14 7 45 12
Cloud Code minikube
Ln 5, Col 1 LF UTF-8 Spaces: 4
cloudshell
nuvemaula@cloudshell:~$ python3 exe8006.py < dados.in
[1, 2, 3, 4, 5, 6]
[20, 2, 3, 4, 14, 7, 45, 12]
nuvemaula@cloudshell:~$


1 2 3 4 5 6
20 2 3 4 14 7 45 12
12 3 4 444 4444 442 21 44 1 2
0 0 Cloud Code minikube
Ln 3, Col 41 LF UTF-8 Spaces: 4
cloudshell
nuvemaula@cloudshell:~$

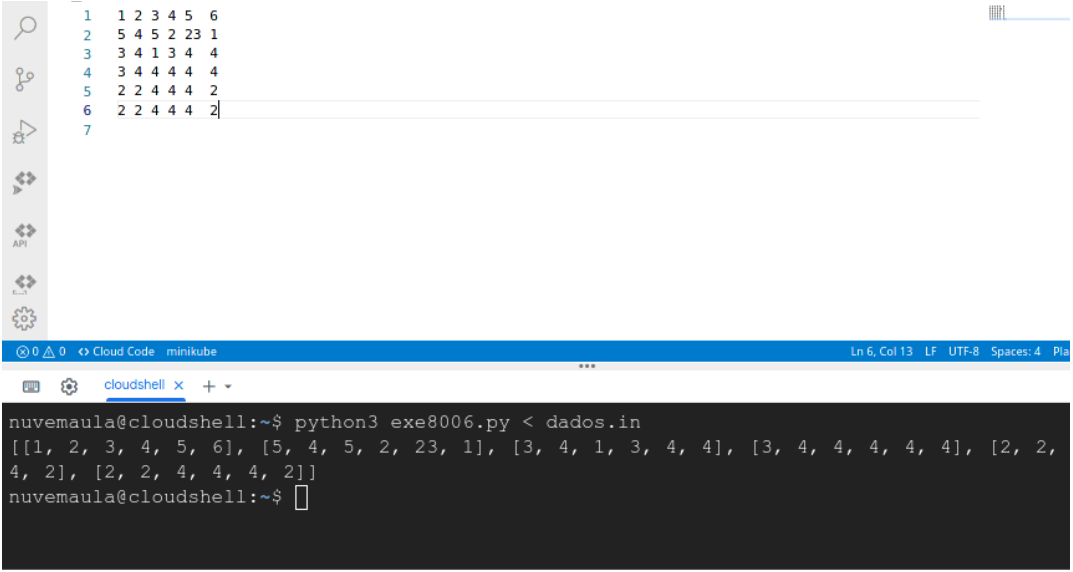
```python
for x in [1, 2, 3]:
  A = input()

  L = []
  i=j=0
  while j<=len(A):
    if i<len(A) and A[i]==" ":
        i = j = i+1
    elif j<len(A) and A[j]!=" ":
        j = j+1
    elif i<len(A) and A[i]!=" ":
        L = L + [ int(A[i:j+1]) ]
        i = j = j+1
    else:
        i = j = j+1
  print(L)
```

```
nuvemaula@cloudshell:~$ python3 exe8006.py < dados.in
[1, 2, 3, 4, 5, 6]
[20, 2, 3, 4, 14, 7, 45, 12]
[12, 3, 4, 444, 4444, 442, 21, 44, 1, 2]
nuvemaula@cloudshell:~$
```


1 2 3 4 5 6
20 2 3 4 14 7 45 12
12 3 4 444 4444 442 21 44 1 2
Ln 3, Col 41 LF UTF-8 Spaces: 4
cloudshell
nuvemaula@cloudshell:~$ python3 exe8006.py < dados.in
[1, 2, 3, 4, 5, 6]
[20, 2, 3, 4, 14, 7, 45, 12]
[12, 3, 4, 444, 4444, 442, 21, 44, 1, 2]
nuvemaula@cloudshell:~$

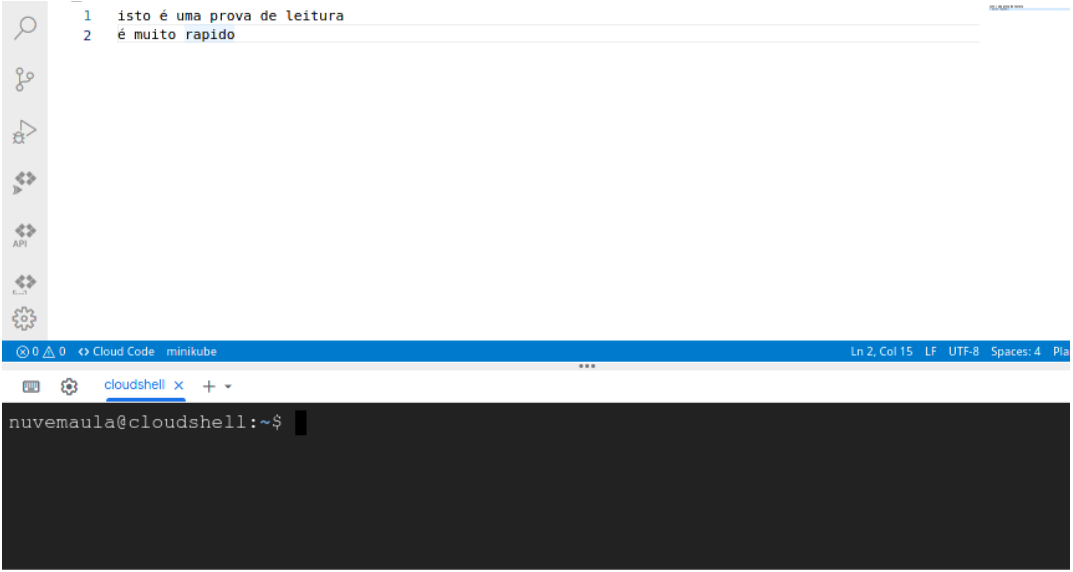
```
1  M = [range(6) for i in range(6)]
2  for x in range(6):
3    A = input()
4    L = []
5    i=j=0
6    while j<=len(A):
7      if i<len(A) and A[i]==" ":
8          i = j = i+1
9      elif j<len(A) and A[j]!=" ":
10         j = j+1
11     elif i<len(A) and A[i]!=" ":
12         L = L + [ int(A[i:j+1]) ]
13         i = j = j+1
14     else:
15         i = j = j+1
16   M[x] = L
17 print(M)
```

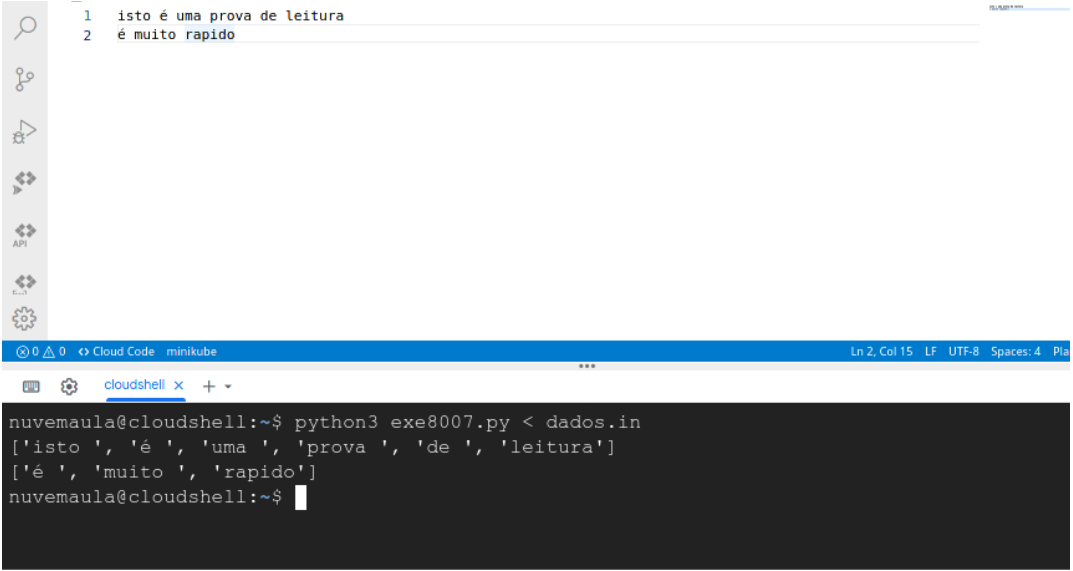
```
nuvemaula@cloudshell:~$ python3 exe8006.py < dados.in
[[1, 2, 3, 4, 5, 6], [5, 4, 5, 2, 23, 1], [3, 4, 1, 3, 4, 4], [3, 4, 4, 4, 4, 4], [2, 2, 4, 2], [2, 2, 4, 4, 4, 2]]
nuvemaula@cloudshell:~$
```


1 2 3 4 5 6
5 4 5 2 23 1
3 4 1 3 4 4
3 4 4 4 4 4
2 2 4 4 4 2
2 2 4 4 4 2
API
0 0 Cloud Code minikube
Ln 6, Col 13 LF UTF-8 Spaces: 4
cloudshell
nuvemaula@cloudshell:~$ python3 exe8006.py < dados.in
[[1, 2, 3, 4, 5, 6], [5, 4, 5, 2, 23, 1], [3, 4, 1, 3, 4, 4], [3, 4, 4, 4, 4, 4], [2, 2, 4, 2], [2, 2, 4, 4, 4, 2]]
nuvemaula@cloudshell:~$

# Exercício 5

# Leitura de strings

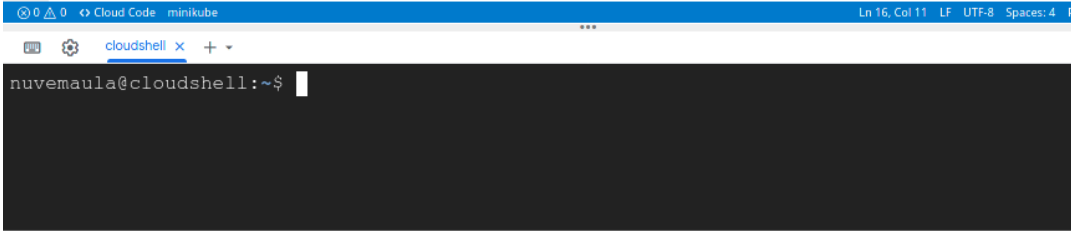

1 isto é uma prova de leitura
2 é muito rapido
API
0 0 Cloud Code minikube
Ln 2, Col 15 LF UTF-8 Spaces: 4
cloudshell
nuvemaula@cloudshell:~$


1 isto é uma prova de leitura
2 é muito rapido
0 0 Cloud Code minikube
Ln 2, Col 15 LF UTF-8 Spaces: 4
cloudshell
nuvemaula@cloudshell:~$ python3 exe8007.py < dados.in
['isto ', 'é ', 'uma ', 'prova ', 'de ', 'leitura']
['é ', 'muito ', 'rapido']
nuvemaula@cloudshell:~$

```python

for x in range(2):
    A = input()
    L = []
    i=j=0
    while j<=len(A):
      if i<len(A) and A[i]==" ":
          i = j = i+1
      elif j<len(A) and A[j]!=" ":
          j = j+1
      elif i<len(A) and A[i]!=" ":
          L = L + [ (A[i:j+1]) ]
          i = j = j+1
      else:
          i = j = j+1
    print(L)
```

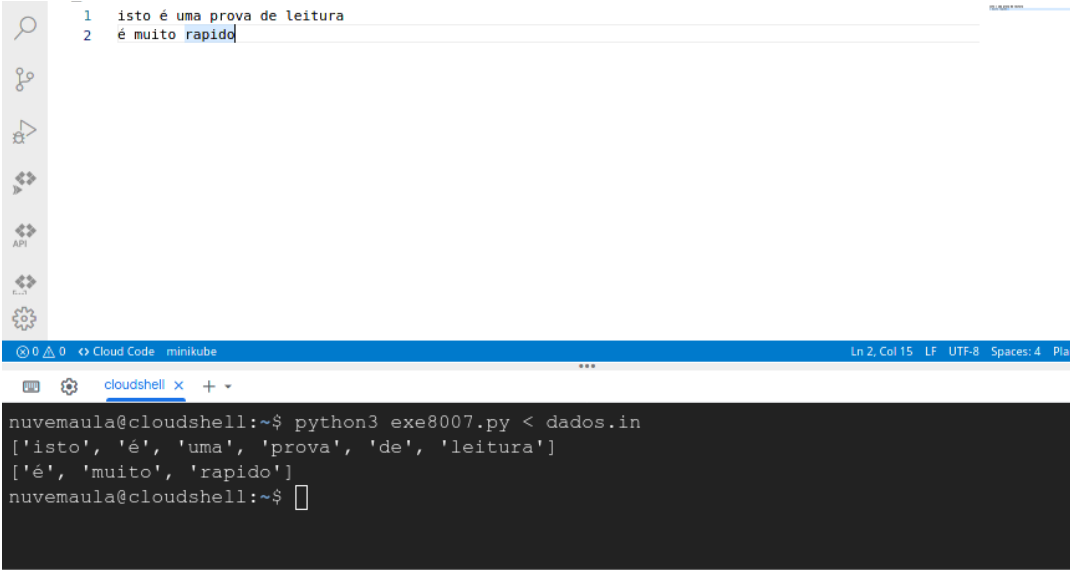
```
nuvemaula@cloudshell:~$ python3 exe8007.py < dados.in
['isto ', 'é ', 'uma ', 'prova ', 'de ', 'leitura']
['é ', 'muito ', 'rapido']
nuvemaula@cloudshell:~$
```

```
for x in range(2):
  A = input()
  L = []
  i=j=0
  while j<=len(A):
    if i<len(A) and A[i]==" ":
        i = j = i+1
    elif j<len(A) and A[j]!=" ":
        j = j+1
    elif i<len(A) and A[i]!=" ":
        L = L + [ (A[i:j+1]).strip() ]
        i = j = j+1
    else:
        i = j = j+1
  print(L)
```



```
1
2  for x in range(2):
3    A = input()
4    L = []
5    i=j=0
6    while j<=len(A):
7      if i<len(A) and A[i]==" ":
8          i = j = i+1
9      elif j<len(A) and A[j]!=" ":
10         j = j+1
11     elif i<len(A) and A[i]!=" ":
12         L = L + [ (A[i:j+1]).strip() ]
13         i = j = j+1
14     else:
15         i = j = j+1
16   print(L)
```

```
nuvemaula@cloudshell:~$ python3 exe8007.py < dados.in
['isto', 'é', 'uma', 'prova', 'de', 'leitura']
['é', 'muito', 'rapido']
nuvemaula@cloudshell:~$
```


1 isto é uma prova de leitura
2 é muito rapido
API
0 0 Cloud Code minikube
Ln 2, Col 15 LF UTF-8 Spaces: 4
cloudshell
nuvemaula@cloudshell:~$ python3 exe8007.py < dados.in
['isto', 'é', 'uma', 'prova', 'de', 'leitura']
['é', 'muito', 'rapido']
nuvemaula@cloudshell:~$

# Perguntas ....

# Referências


- Zanoni Dias, MC102, Algoritmos e Programação de Computadores, IC/UNICAMP, 2021. https://ic.unicamp.br/~mc102/

  Aula Introdutória [ slides ] [ vídeo ]
  Primeira Aula de Laboratório [ slides ] [ vídeo ]
  Python Básico: Tipos, Variáveis, Operadores, Entrada e Saída [ slides ] [ vídeo ]
  Comandos Condicionais [ slides ] [ vídeo ]
  Comandos de Repetição [ slides ] [ vídeo ]
  Listas e Tuplas [ slides ] [ vídeo ]
  Strings [ slides ] [ vídeo ]
  Dicionários [ slides ] [ vídeo ]
  Funções [ slides ] [ vídeo ]
  Objetos Multidimensionais [ slides ] [ vídeo ]
  Algoritmos de Ordenação [ slides ] [ vídeo ]
  Algoritmos de Busca [ slides ] [ vídeo ]
  Recursão [ slides ] [ vídeo ]
  Algoritmos de Ordenação Recursivos [ slides ] [ vídeo ]
  Arquivos [ slides ] [ vídeo ]
  Expressões Regulares [ slides ] [ vídeo ]
  Execução de Testes no Google Cloud Shell [ slides ] [ vídeo ]
  Numpy [ slides ] [ vídeo ]
  Pandas [ slides ] [ vídeo ]

- Panda - Cursos de Computação em Python (IME -USP) https://panda.ime.usp.br/

  Como Pensar Como um Cientista da Computação https://panda.ime.usp.br/pensepy/static/pensepy/
  Aulas de Introdução à Computação em Python https://panda.ime.usp.br/aulasPython/static/aulasPython/

- Fabio Kon, Introdução à Ciência da Computação com Python http://bit.ly/FabioKon/
- Socratica, Python Programming Tutorials http://bit.ly/SocraticaPython/
- Google - online editor for cloud-native applications (Python programming) https://shell.cloud.google.com/
- w3schools - Python Tutorial https://www.w3schools.com/python/
- Outros, citados nos Slides.