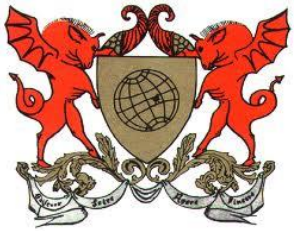
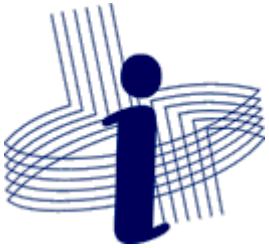


Universidade Federal de Viçosa
Departamento de Informática
Centro de Ciências Exatas e Tecnológicas



INF 100 – Introduction to Programming

Arrays

(Exercises)

Exercise 1

Simulate the execution of the program below:

```
A = [3,5,4,1,2]
size = len(A)

for i in range(0, size-1):
    print("position", i, end=" ")
    minPos = i
    for j in range(i+1, size):
        if A[j] < A[minPos]:
            minPos = j
    if minPos == i:
        print("is OK")
    else:
        print ("is exchanged with position", minPos)
        aux = A[i]
        A[i] = A[minPos]
        A[minPos] = aux

print(A)
```

Exercise 1

Simulate the execution of the program below:

```
position 0 is exchanged with position 3
position 1 is exchanged with position 4
position 2 is exchanged with position 3
position 3 is OK
[1, 2, 3, 4, 5]
```

```
A = [3,5,4,1,2]
size = len(A)

for i in range(0, size-1):
    print("position", i, end=" ")
    minPos = i
    for j in range(i+1, size):
        if A[j] < A[minPos]:
            minPos = j
    if minPos == i:
        print("is OK")
    else:
        print ("is exchanged with position", minPos)
        aux = A[i]
        A[i] = A[minPos]
        A[minPos] = aux

print(A)
```

Exercise 2

Write a Python program that reads two vectors and prints their *scalar product* (also called *dot product*).

The dot product of two vectors **A** = [A_1, A_2, ..., A_n] and **B** = [B_1, B_2, ..., B_n] is defined as:

$$\mathbf{A} \cdot \mathbf{B} = \sum_{i=1}^{n} A_i B_i = A_1 B_1 + A_2 B_2 + \cdots + A_n B_n$$

Exercise 2

Write a Python program that reads two vectors and prints their *scalar product* (also called *dot product*).

The dot product of two vectors **A** = [A_1, A_2, ..., A_n] and **B** = [B_1, B_2, ..., B_n] is defined as:

$$\mathbf{A} \cdot \mathbf{B} = \sum_{i=1}^{n} A_i B_i = A_1 B_1 + A_2 B_2 + \cdots + A_n B_n$$

```
What is the size of the sequences? 3
Type 3 numbers for the first sequence:
2
4
-5
Type 3 numbers for the second sequence:
3
5
2
Scalar product= 16.0
```

```
import numpy as np

size = int(input("What is the size of the sequences? "))
A = np.empty(size)
B = np.empty(size)

print("Type", size, "numbers for the first sequence:")
for i in range(0, size):
    A[i] = int(input(""))
print("Type", size, "numbers for the second sequence:")
for i in range(0, size):
    B[i] = int(input(""))
```


```
import numpy as np

size = int(input("What is the size of the sequences? "))
A = np.empty(size)
B = np.empty(size)

print("Type", size, "numbers for the first sequence:")
for i in range(0, size):
    A[i] = int(input(""))
print("Type", size, "numbers for the second sequence:")
for i in range(0, size):
    B[i] = int(input(""))

sum = 0
for i in range(0, size):
    sum += A[i] * B[i]

print("Scalar product=", sum)
```

Exercise 3

Write a Python program that reads two sequences of integer values and indicates whether they are equal or different.

Exercise 3

Write a Python program that reads two sequences of integer values and indicates whether they are equal or different.

```
What is the size of the sequences? 3
Type 3 numbers for the first sequence:
1
5
3
Type 3 numbers for the second sequence:
1
5
3
The sequences are equal
```

Exercise 3

Write a Python program that reads two sequences of integer values and indicates whether they are equal or different.

```
What is the size of the sequences? 3
Type 3 numbers for
1
5
3
Type 3 numbers for
1
5
3
The sequences are
```

```
What is the size of the sequences? 5
Type 5 numbers for the first sequence:
1
2
3
4
5
Type 5 numbers for the second sequence:
1
3
3
4
5
The sequences are different
```

```
import numpy as np

size = int(input("What is the size of the sequences? "))
# OBS: parameter dtype indicates type of the cells
A = np.empty(size, dtype=int)
B = np.empty(size, dtype=int)

print("Type", size, "numbers for the first sequence:")
for i in range(0, size):
    A[i] = int(input(""))
print("Type", size, "numbers for the second sequence:")
for i in range(0, size):
    B[i] = int(input(""))
```

```
import numpy as np

size = int(input("What is the size of the sequences? "))
# OBS: parameter dtype indicates type of the cells
A = np.empty(size, dtype=int)
B = np.empty(size, dtype=int)

print("Type", size, "numbers for the first sequence:")
for i in range(0, size):
    A[i] = int(input(""))
print("Type", size, "numbers for the second sequence:")
for i in range(0, size):
    B[i] = int(input(""))

isEqual = True
for i in range(0, size):
    if (A[i] != B[i]):
        isEqual = False
        break
if isEqual:
    print("The sequences are equal")
else:
    print("The sequences are different")
```

Exercise 4

We will call two sequences of numerical values ***compatible*** if they have the same number of elements and:

- whenever the values decrease in a sequence, the other sequence does not increase in the same position;
- whenever the values increase in a sequence, the other sequence does not decrease in the same position.

Write a Python program that reads two sequences of integer values and indicates whether they are compatible.

```
What is the size of the sequences? 5
Type 5 numbers for the first sequence:
1
5
8
8
3
Type 5 numbers for the second sequence:
3
3
6
5
4
The sequences are compatible
```

```
What is the size of the sequences? 5
Type 5 numbers for the first sequence:
1
5
8
8
3
Type
3
3
6
5
4
The s
```

```
What is the size of the sequences? 4
Type 4 numbers for the first sequence:
2
4
6
8
Type 4 numbers for the second sequence:
3
3
1
7
The sequences are not compatible
```

```
import numpy as np

size = int(input("What is the size of the sequences? "))
A = np.empty(size)
B = np.empty(size)

print("Type", size, "numbers for the first sequence:")
for i in range(0, size):
    A[i] = int(input(""))
print("Type", size, "numbers for the second sequence:")
for i in range(0, size):
    B[i] = int(input(""))
```

```
import numpy as np

size = int(input("What is the size of the sequences? "))
A = np.empty(size)
B = np.empty(size)

print("Type", size, "numbers for the first sequence:")
for i in range(0, size):
    A[i] = int(input(""))
print("Type", size, "numbers for the second sequence:")
for i in range(0, size):
    B[i] = int(input(""))

isCompatible = True
for i in range(0, size-1):
    if (A[i] < A[i+1] and B[i] > B[i+1] or
            A[i] > A[i+1] and B[i] < B[i+1]):
        isCompatible = False
        break
if isCompatible:
    print("The sequences are compatible")
else:
    print("The sequences are not compatible")
```

Exercise 5

Write a Python program that reads a word and prints it in reversal order.

```
Type a word: programming
gnimmargorp
```

Exercise 5

Write a Python program that reads a word and prints it in reversal order.

```
Type a word: programming
gnimmargorp
```

```
w = input("Type a word: ")

size = len(w)
for i in range(0, size):
    print(w[size-i-1], end='')
```

Exercise 6

Write a Python program that reads a word and:

- **indicates whether the word is a *palindrome*;**
- **prints only the first letter, then the first and second letters, and so on, until printing the whole word.**

```
Type a word: abcba
The word is a palindrome
a
ab
abc
abcb
abcba
```

```
w = input("Type a word: ")

size = len(w)
```

```
w = input("Type a word: ")

size = len(w)

isPalindrome = True
s = size//2
for i in range(0, s):
    if w[i] != w[size-i-1]:
        isPalindrome = False
        break
if isPalindrome:
    print("The word is a palindrome")
else:
    print("The word is not a palindrome")
```

```
w = input("Type a word: ")

size = len(w)

isPalindrome = True
s = size//2
for i in range(0, s):
    if w[i] != w[size-i-1]:
        isPalindrome = False
        break
if isPalindrome:
    print("The word is a palindrome")
else:
    print("The word is not a palindrome")

for i in range(0, size):
    for j in range(0, i+1):
        print(w[j], end='')
    print("")
```