Rand<randomgrayRand<g Rand<GRand<g Rand<Grandomgray

German University in Cairo
Faculty of Media Engineering and Technology
Prof. Dr. Slim Abdennadher

Introduction to Computer Science, Winter Semester 2017
Practice Assignment 7

Discussion: 9.12.2017 - 14.12.2017

Exercise 7-1 in class
Sum

Given two lists A and B of the form A0,...,A(n-1) and B0,...,B(n-1) respectively. Write an algorithm that stores the sum of the corresponding elements of the lists A and B in a new list C.

Solution:

```
list_1 = eval(input())
list_2 = eval(input())
n = len(list_1)
list_Sum = [0]*n
i = 0
while i < n:
    list_Sum[i] = list_1[i] + list_2[i]
    i = i + 1
print(list_Sum)
```

Exercise 7-2 in class
Search

The simplest algorithm to search a list of Numbers N0,....,N(m-1) for a given key Key is to test successively each element.

```
N = eval(input("Enter the list of elements:"))
Key = eval(input("Enter a key:"))
m=len(N)
i = 0
FOUND = False
while i < m and FOUND == False:
    if Key == N[i]:
        FOUND = True
    else:
        i = i+1
if FOUND == False:
    print("Key found")
else:
    print("Key not found")
```

If a list is already stored in increasing order, a modified sequential search algorithm can be used that compares against each element in turn, stopping if a list element exceeds the target value. Write an algorithm for the modified sequential search.

Solution:

```
N = eval(input("Enter the list of elements:"))
Key = eval(input("Enter a key:"))
m=len(N)
i = 0
FOUND = False
```

Solution:

```
list_A = eval(input())
n = len(list_A)
list_B = [0] * n
i = 0
j = n-1
while(i < n):
    list_B[i] = list_A[j]
    i = i+1
    j = j-1
print(list_B)
```

Another solution:

```
list_A = eval(input())
n = len(list_A)
list_B = []
i = 0
j = n-1
while(i < n):
    list_B = list_B + list_A[j]
    i = i+1
    j = j-1
print(list_B)
```

Another solution:

```
list_A = eval(input())
n = len(list_A)
i = 0
j = n-1
while(i < n//2):
    temp = list_A[i]
    list_A[i] = list_A[j]
    list_A[j] = temp
    i = i+1
    j = j-1
print(list_A)
```

Exercise 7-4

Ordered

Given a list of integers, write an algorithm that checks whether a list is ordered in ascending order or not.

For example for the list consisting of

```
5 4 12 16 1
```

The algorithm should display

The list is not sorted

For the list

```
5 10 12 16
```

The algorithm should display

The list is sorted

Note: For the case where the list is not sorted, your algorithm should stop right away. For the example above, your algorithm should stop after comparing the 5 with the 4.

Solution:

```
A=eval(input())
n=len(A)-1
i = 1
isSorted = True
while(i < n and isSorted):
    if(A[i] > A[i+1]):
        isSorted = False
    i = i + 1
if(isSorted):
    print("The list is sorted")
else:
    print("The list is not sorted")
```