Algorithmics
Correction Final Exam #1 (P1)

UNDERGRADUATE 1^{st} YEAR S1 – EPITA

7 January 2020 - 13h-15h

Solution 1 **(Abstract Types: Iterative lists (delete) – *3 points*)**

OPERATIONS

delete : list × integer → list

where *delete(λ, k)* deletes the k^{th} element of the list λ (k begins at 1).

1. We can put the following precondition that will "shorten" the axioms. *Do not specify it here does not matter if the predicate exists at axioms.*

   PRECONDITIONS

   delete(λ, k) **is-defined-iaoi** $\lambda \neq$ *emptylist* & $1 \leqslant i \leqslant$ *length(λ)*

   The first condition $\lambda \neq$ *emptylist* can be omited, as it is "included" in the second!

2. The axioms are as follows:

   AXIOMS

   $length(delete(\lambda, k)) = length(\lambda) - 1$
   $1 \leq i < k \Rightarrow nth(delete(\lambda, k), i) = nth(\lambda, i)$
   $k \leq i \leq length(\lambda) - 1 \Rightarrow nth(delete(\lambda, k), i) = nth(\lambda, i + 1)$

   WITH

   λ : List
   k, i : Integer

Solution 2 **(Binary Search: search "path" – *2 points*)**

	NO	YES
46 - 65 - 81 - 73 - 70 - 66		√
31 - 62 - 90 - 72 - 61 - 66	√	
36 - 70 - 53 - 40 - 42 - 66	√	
35 - 51 - 55 - 58 - 61 - 66		√

Solution 3 **(Test -)**

`test(`x`,` L`)` tests whether x is in the list L.

Solution 4 **(Integers $\leftrightarrow$ list – *6 points*)**

1. The function `int_to_list(`n`,` p`)` returns the list of the p digits of n:

```
def int_to_list(n, p):
    L = []
    while n != 0:
        L.append(n % 10)
        n = n // 10
    for i in range(p-len(L)):
        L.append(0)
    return L

#-------------------------------------------------------------------------

def int_to_list2(n, p):
    L = []
    while p > 0:
        L.append(n % 10)
        n = n // 10
        p -= 1
    return L
```

2. The function `list_to_ints(`$[d_1, d_2, \cdots, d_p]$`)` returns the pair of integers $(d_1 d_2 \cdots d_p,\ d_p \cdots d_2 d_1)$:

```
def list_to_ints(L):
    left = 0
    right = 0
    n = len(L)
    for i in range(n):
        left = left * 10 + L[i]
        right = right * 10 + L[n-i-1]
    return (left, right)

#-------------------------------------------------------------------------

def list_to_ints2(L):
    left = 0
    right = 0
    p = 1
    for i in range(len(L)):
        left = left * 10 + L[i]
        right = right + L[i]*p
        p = p * 10
    return (left, right)
```

Solution 5 **(Histogram and sort – *4 points*)**

1. The function `hist(`L`)` returns the list that represents the histogram of the values in L (L contains only digits):

```
1
2       def hist(L):
3           H = []
4           for i in range(10):
5               H.append(0)
6           # ou H = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
7           for e in L:
8               H[e] += 1
9           return H
```

2. The function `sort(`L`)` returns the list L sorted in increasing order (L contains only digits):

```
1
2       def sort(L):
3           H = hist(L)
4           L = []
5           for i in range(10):
6               for nb in range(H[i]):
7                   L.append(i)
8           return L
```

Solution 6 **(Kaprekar – *5 points*)**

The function `Kaprekar(`n`, `p`)` performs the Kaprekar routine to the positive p-digit integer n, till it reaches the same value twice. It displays the computed values.

```
1       def Kaprekar(n, p):
2           L = []
3           while not test(n, L):
4               L.append(n)
5               digits = int_to_list(n,p)
6               digits = sort(digits)
7               (low, high) = list_to_ints(digits)
8               n = high - low
9           L.append(n)
10          return L
```