

Name : \_\_\_\_\_

Code permanent : \_\_\_\_\_

Place number: \_\_\_\_\_

Directives :

- Write your name, first name, code permanent and place number.
- Read carefully all questions and **answer directly on the questionnaire.**
- You can only use a pen or pencil, **no documentation or other object is allowed.**
- This exam contains 7 questions for 110 points.
- Be careful with time as approximately 1 point = 1 minute.
- This exam contains 10 pages, including 2 draft pages at the end.
- **Write visibly and detail your answers when necessary.**
- You have 110 minutes to complete this exam.

GOOD LUCK !

1	/ 15
2	/ 5
3	/ 20
4	/ 15
5	/ 15
6	/ 25
7	/ 15
Total	/ 110

1. (15) Write a Python function (or pseudo-code), `renverse`, which reverse the elements of a sequence (i.e. Python list), so that the elements are returned in a new sequence in the reverse order to the initial sequence, e.g.  
`renverse( [ 1,2,3,4 ] )` returns `[ 4,3,2,1 ]`. NB. you cannot use the Python reverse function in your implementation, and neither a stack.

2. (5) Consider the following Python instructions and give the expected result of the `print` instructions.

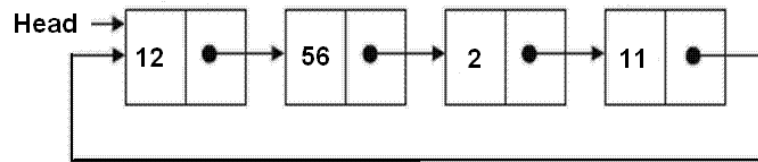
```
x = [1,2,3]
y = x
x.append( 4 )
y.append( 5 )
print( x )
print( y )
```

3. (20) Write a Python function (or pseudo-code), `sum_odds`, which computes the sum of the odd elements of a sequence `S` passed as an argument.
  - a) (5) Give the exact mathematical expressions that give the number of elementary instructions executed in the best case, on average, and in the worst case.
  - b) (5) Give the order (big-O) to estimate the growing execution time in the best case, on average, and in the worst case, in terms of `n`, that is the number of elements in `S`.

4. (15) Write a recursive Python function (or pseudo-code), `recursivemult`, to multiply two numbers, `n` and `m`, only by using the addition and subtraction operations.

5. (15) Write a Python function (or pseudo-code), `renverse`, which reverse the elements of a sequence (i.e. Python list), so that the elements are returned in a new sequence in the reverse order to the initial sequence, e.g.  
`renverse( [ 1, 2, 3, 4 ] )` returns `[ 4, 3, 2, 1 ]`. This time using a stack!

6. (25) A circular singly linked list (CSLL) is a singly linked list in which the **next** element of the last element points to the **head** of the list.



- a) (10) Write a Python function (or pseudo-code) to compute the number of elements in a CSLL.
- b) (15) Suppose two references, **x** et **y**, to nodes in two CSLL which are not necessarily the same list. Write a Python function (or pseudo-code) to determine if **x** and **y** points to nodes in the same CSLL.

7. (15) Fill the array below by showing in each line the state changes of a corresponding array when we sort it using heapsort. Start with the initial heap. Draw on the next page the initial heap and subsequent alterations. The last line will have the elements sorted in increasing order. (*NB. the number of lines in the array below does not necessarily represent the exact number needed to sort the values of its initial state*).

[illegible]

Draw your heaps here:



**Draft 1**

**Draft 2**