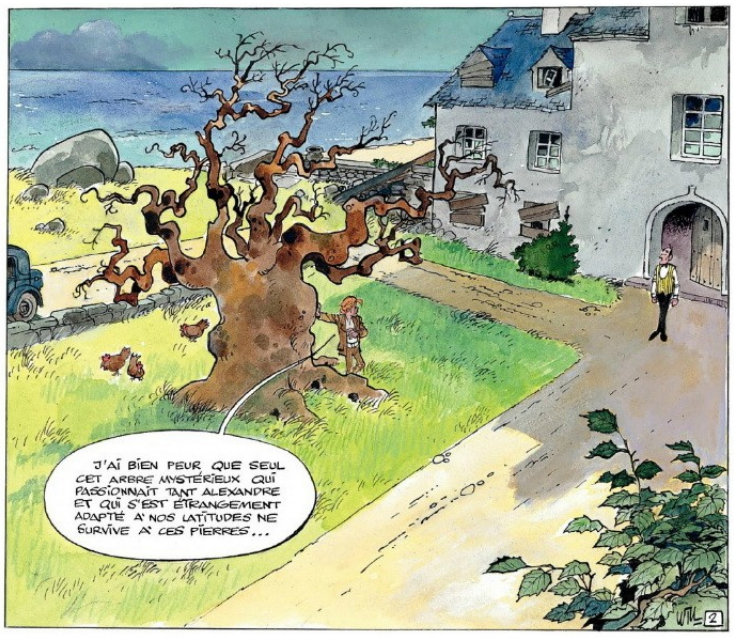
Algorithmics
Test #2 (C2)

Undergraduate 1^{st} year (S2)
EPITA

9 Mar. 2016 - 9:30 (D.S. 307186.87 BW)

Instructions (read it) :

- ☐ You must answer on **the answer sheets provided.**
  - No other sheet will be picked up. Keep your rough drafts.
  - Answer within the provided space. **Answers outside will not be marked**: Use your drafts!
  - Do not separate the sheets unless they can be re-stapled before handing in.
  - Penciled answers will not be marked.
- ☐ The presentation is negatively marked, which means that you are marked out of 20 points and the presentation points (maximum of 2) are taken off this grade.
- ☐ **Code:**
  - All code must be written in the language `Python` (no C, CAML, ALGO or anything else).
  - **Any `Python` code not indented will not be marked.**
  - All that you need (types, routines) is indicated in the **appendix** (last page)!
  - Your functions must follow the given examples of application.
- ☐ Duration : 2h



1 Expressions and trees

Reminder

An expression can be represented by a tree: internal nodes contain the operators and external ones contain the operands. Here we work with arithmetical expressions that use the binary operators +, -, * and /.

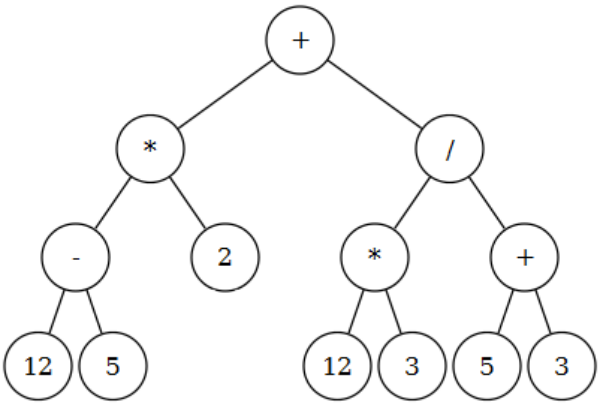


Figure 1: Tree for the expression $(12-5)*2+(12*3)/(5+3)$

Exercise 1.1 (Draw me – *5 points*)

Let B_1, B_2, B_3 and B_4 be binary trees representing expressions.

During the depth-first traversal:

- values of B_1 in preorder give the list: + - / 8 2 5 + 4 * 2 3
- values of B_2 in postorder give the list: 8 20 - 2 2 * 2 + /
- the traversals of B_3 and B_4 give the same inorder list: 8 + 7 / 5 - 4 * 2
- B_3 and B_4 are different but the two expressions they represent have the same result : -5.

Draw the trees B_1, B_2, B_3 and B_4. Give the values of the expressions from B_1 and B_2.

Exercise 1.2 (Count me – *3 points*)

Write the function `nodes` that count the number of operators and operands of an expression represented by a binary tree.

Application example with the tree in figure 1:

```
>>> nodes(B)
(6, 7)
```

Exercise 1.3 (Display me – *3 points*)

Write the function `exp2str` that returns a string of the fully parenthesized expression represented by a tree.

Application example with the tree in figure 1:

```
>>> exp2str(B)
'(((12-5)*2)+((12*3)/(5+3)))'
```

2 Some matrices

1	10	3	0	3	10	1
1	0	1	8	1	0	1
10	9	14	1	14	9	10
10	3	7	11	7	3	10
7	8	5	1	5	8	7

Figure 2: `Mat1`

1	10	3	3	10	1
1	0	1	1	0	1
10	9	4	4	9	10
10	3	7	7	3	10
7	8	15	15	8	7

Figure 3: `Mat2`

1	24	12	18	4
10	15	15	0	18
8	14	0	16	2
22	4	8	14	22
19	7	23	5	5

Figure 4: `Mat3`

The matrices are assumed to be non empty in the exercises below.

Exercise 2.1 (Minimax – *5 points*)

Write a function that searches for the minimum value of the maximums of each line in an integer matrice.

Two possible versions of this fonction:

- it returns the searched value: `minimax`
- it returns the position of the searched value: `posMinimax`

Application examples with the matrix in figure 4:

```
>>> minimax(Mat3)
16
>>> posMinimax(Mat3)
(2, 3)
```

Choose the version that suits you keeping in mind that it is obviously the second version that will bring the most points.[1]

Exercise 2.2 (Symmetry – *5 points*)

Write the function `symetric` that tests whether a matrix has a vertical axis of symmetry (horizontal symmetry).

Application examples on matrices in figures 2, 3 and 4:

```
>>> symetric(Mat1)
True
>>> symetric(Mat2)
True
>>> symetric(Mat3)
False
```

[1] Sometimes a few points are better than none.

Appendix

Binary Trees

The binary trees we work on are the same as the ones in tutorials.

- `None` is the empty tree.
- The non-empty tree has 3 attributes: `key`, `left`, `right`.

Authorised functions and methods

Functions you can use:

- `len` on lists.
- `range`.
- `str` on strings: it converts its parameter into a string.
- `min` and `max`, but only with two integer values!

```
1    >>> max(12, 5)
2    12
3
4    >>> min(12, 5)
5    5
```

Special value:

Suppose there exists a value `maxint` = the greatest representable integer.

Your functions

You can write your own functions as long as they are documented (we have to known what they do).
In any case, the last written function should be the one which answers the question.