Binary Trees (Arbres binaires)
Display and load

ASCII art!

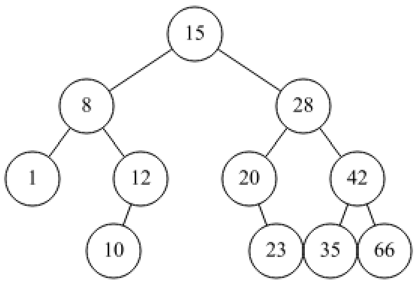
Here are two examples of the same tree displays:

```
>>> printTree(B)
 -  82
  | -  66
  |  | -  75
  |  |  | -  26
  |  |    -  46
  |    -  59
  |     | -  45
  |       -  96
    -  13
     | -  45
     |  | -  29
     |    -  15
       -  47
        | -  77
          -  62
```

```
>>> prettyPrintTree(B)
               82
              /  \
             /    \
            /      \
           /        \
          /          \
         /            \
        /              \
       /                \
      66                13
      / \               / \
     /   \             /   \
    /     \           /     \
   /       \         /       \
  75       59       45       47
  / \      / \      / \      / \
 /   \    /   \    /   \    /   \
26   46  45   96  29   15  77   62
```

The real pretty display

dot (graphviz)



"Simple" dot:

```
graph {
   15 -- 8
   15 -- 28
   8 -- 1
   8 -- 12
   28 -- 20
   28 -- 42
   12 -- 10
   20 -- 23
   42 -- 35
   42 -- 66
}
```

Problem: does not manage identical keys!

Dot with id:

```
graph G {
        N161384992[label="1"];
        N161385944 -- N161384992;
        N161385944[label="8"];
        N161383704 -- N161385944;
        N161384432[label="10"];
        N161384768 -- N161384432;
        N161384768[label="12"];
        N161385944 -- N161384768;
        N161383704[label="15"];
        N161510232[label="20"];
        N161384376 -- N161510232;
        N161510288[label="23"];
        N161510232 -- N161510288;
        N161384376[label="28"];
        N161383704 -- N161384376;
        N161510736[label="35"];
        N161510064 -- N161510736;
        N161510064[label="42"];
        N161384376 -- N161510064;
        N161513088[label="66"];
        N161510064 -- N161513088;
}
```

Real bonus!

Manage the simple nodes...

Load binary trees

The above tree was loaded from a textfile that contains:

```
(15(8(1()())(12(10()())()))(28(20()(23()()))(42(35()())(66()()))))
```

And what about this textfile?

```
(A
  (B
    (D
      (H
        ()
        ()
        )
      (I
        ()
        ()
        )
      )
    (E
      (J
        ()
        ()
        )
      (K
        ()
        ()
        )
      )
    )
  (C
    (F
      (L
        ()
        ()
        )
      (M
        ()
        ()
        )
      )
    (G
      (N
        ()
        ()
        )
      (O
        ()
        ()
        )
      )
    )
  )
```

Do not forget to test if the string in the file is "well-formed"!

Both files can be found in `https://algo-td.infoprepa.epita.fr//algo/S2/Python/files/`