# Quiz 1D: Boolean Expressions in Python

**Question 1:** Given the truth table (with no-argument functions f(), g(), h() returning Boolean values) find the expression to compute the truth table.   
(BTW, it expresses the following implication: **if** f() **then** g() **and** h(). But there is no implication in Python, so you have to rewrite.)  
Select 2 answers.

|  |  |  |  |
| --- | --- | --- | --- |
| f() | g() | h() | B(f(),g(),h()) |
| False | False | False | True |
| False | False | True | True |
| False | True | False | True |
| False | True | True | True |
| True | False | False | False |
| True | False | True | False |
| True | True | False | False |
| True | True | True | True |

(**not** f() **or** g() **or** h()) **and**  (**not** f() **or** g() **or** **not** h()) **and** (**not** f() **or not** g() **or** h())

**not** f() **and** g() **and** h() **or**  **not** f() **and** g() **and** **not** h() **or** **not** f() **and not** g() **and** h()

**not** f() **or** f() **and** g() **and** h()

**Question 2:** Drop the unnecessary parentheses in a given Boolean expression   
(Provide the precedence table first).

¬, ∧, ∨, ⊕, 🡪 ↔

|  |  |  |
| --- | --- | --- |
| Operation | Precedence | Associativity |
| **not** | 1 | - (unary) |
| **and** | 2 | left |
| **or** | 3 | left |
| ^ (XOR, ⊕) | 4 | left |
| *(implication)* | 5 | False |
| == | 6 | True |

<https://coq.inria.fr/library/Coq.Init.Logic.html>

OR: Given a syntax tree, find which (minimally parenthesized) Boolean expression represents that tree?

**Question 3:** When evaluating row ... in this expression (a CNF?) what is the order how the expressions are evaluated?   
Assume that short circuit execution happens (True or f() 🡪 True right away).

**Question 4:** (Short-circuit in practice?)  
uu = (a - b)<0 or sqrt(a - b)>0.