**Questions en vrac sur la logique booléenne : tirées E3C**

**Question A.1**

1. À quelle affectation sont équivalentes les instructions suivantes, où a, b sont des variables entières et c une variable booléenne ?

if a==b:

c = True

elif a > b+10:

c = True

else:

c = False

***Réponses***

A c = (a==b) or (a > b+10)

B c = (a==b) and (a > b+10)

C c = not(a==b)

D c = not(a > b+10)

**Question A.2**

On considère l'extrait de code suivant :

while (a < 20) or (b > 50):

......

......

Quelles conditions permettent de mettre fin à cette boucle ?

Réponses

A la boucle prend fin lorsque a < 20 ou b > 50

B la boucle prend fin lorsque a < 20 et b > 50

C la boucle prend fin lorsque a >= 20 ou b <= 50

D la boucle prend fin lorsque a >= 20 et b <= 50

**Question A.3**

Si A et B sont des variables booléennes, laquelle de ces expressions booléennes est équivalente   
à (not A) or B ?

***Réponses***

A (A and B) or (not A and B)

B (A and B) or (not A and B) or (not A and not B)

C (not A and B) or (not A and not B)

D (A and B) or (not A and not B)

**Question A.4**

Parmi les quatre expressions suivantes, laquelle s'évalue en True ?

Réponses

A False and (True and False)

B False or (True and False)

C True and (True and False)

D True or (True and False)

**Question A.5**

Quelle table de vérité correspond à l'expression (non(A) ou B) ?

Remarque : dans les tables proposées, la première colonne donne les valeurs de A, la première ligne les valeurs de B.

Réponses

|  |  |  |
| --- | --- | --- |
| A\B | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |

A C

|  |  |  |
| --- | --- | --- |
| A\B | 0 | 1 |
| 0 | 0 | 1 |
| 1 | 1 | 1 |

B D

|  |  |  |
| --- | --- | --- |
| A\B | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |

|  |  |  |
| --- | --- | --- |
| A\B | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 1 | 0 |

**Question A.6**

À quelle affectation sont équivalentes les instructions suivantes, où a, b sont des variables entières et c une variable booléenne ?

if a==b:

c = True

elif a > b+10:

c = True

else:

c = False

***Réponses***

A c = (a==b) or (a > b+10)

B c = (a==b) and (a > b+10)

C c = not(a==b)

D c = not(a > b+10)