1. B5 cor

In [ ]:

```
#exo1
def table_negation():
    for p in (True, False):
        print(p,not(p))
    return None
#table_negation()
#exo2
def non(a):
    return 1-a
```

print("non(0) renvoie :",non(0)) print("non(1) renvoie :",non(1))

In [ ]:

```
#for i in (0,1):
#    print(i,non(i))
#exo3
#2
def table_conjonction():
    for i in (False,True):
        for j in (False,True):
            print(i,j,i and j)
    return None
#print(table_conjonction())
#3
def conjonction(x,y):
    return x*y
#print("conjonction(0,0) renvoie :",conjonction(0,0))
#print("conjonction(0,1) renvoie :",conjonction(0,1))
#print("conjonction(1,0) renvoie :",conjonction(1,0))
#print("conjonction(1,1) renvoie :",conjonction(1,1))
#4
for i in (0,1):
    for j in (0,1):
        print(i,j,conjonction(i,j))
```

In [ ]:

```
#exo4
#2
def table_disjonction():
    for i in (False,True):
        for j in (False,True):
            print(i,j,i or j)
print(table_disjonction())
```

Loading [MathJax]/extensions/Safe.js

```
#3
def disjonction(x,y):
    return x+y-x*y
print("disjonction(0,0) renvoie :",disjonction(0,0))
print("disjonction(0,1) renvoie :",disjonction(0,1))
print("disjonction(1,0) renvoie :",disjonction(1,0))
print("disjonction(1,1) renvoie :",disjonction(1,1))
#4
for i in (0,1):
    for j in (0,1):
        print(i,j,disjonction(i,j))
```

In [ ]:

```
#exo6
#2
def table_xor():
    for i in (False,True):
        for j in (False,True):
            print(i,j,i ^ j)
print(table_xor())
#3
def xor(x,y):
    return abs(x-y)
print("xor(0,0) renvoie :",xor(0,0))
print("xor(0,1) renvoie :",xor(0,1))
print("xor(1,0) renvoie :",xor(1,0))
print("xor(1,1) renvoie :",xor(1,1))
#4
for i in (0,1):
    for j in (0,1):
        print(i,j,xor(i,j))
```

Loading [MathJax]/extensions/Safe.js