

In the name of god

By: Hosein Ghadimi

1. What possible values can a Boolean expression have?

=>True and False.

2. Where does the term Boolean originate?

=>It comes from the name of the british mathematician George Boole.

3. What is an integer equivalent to True in Python?

=>any positive or negative integer is True, only 0 is False.

4. What is the integer equivalent to False in Python?

=>0

5. Is the value -16 interpreted as True or False?

=>True.

6. Given the following definitions:

x, y, z = 3, 5, 7 evaluate the following Boolean expressions:

(a) $x == 3$	\Rightarrow	True.
(b) $x < y$	\Rightarrow	True.
(c) $x >= y$	\Rightarrow	False.
(d) $x <= y$	\Rightarrow	True.
(e) $x != y - 2$	\Rightarrow	False.
(f) $x < 10$	\Rightarrow	True.
(g) $x >= 0$ and $x < 10$	\Rightarrow	True and True = True.
(h) $x < 0$ and $x < 10$	\Rightarrow	False and True = False.
(i) $x >= 0$ and $x < 2$	\Rightarrow	True and False = False.
(j) $x < 0$ or $x < 10$	\Rightarrow	False or True = True.
(k) $x > 0$ or $x < 10$	\Rightarrow	True or True = True.
(l) $x < 0$ or $x > 10$	\Rightarrow	False or False = False.

7. Given the following definitions:

$x, y = 3, 5$

$b1, b2, b3, b4 = \text{True}, \text{False}, x == 3, y < 3$

$\Rightarrow b1 == \text{True}.$

$b2 == \text{False}.$

$b3 == \text{True}.$

$b4 == \text{False}.$

evaluate the following Boolean expressions:

- (a) $b_3 \Rightarrow \text{True.}$
- (b) $b_4 \Rightarrow \text{False.}$
- (c) $\text{not } b_1 \Rightarrow \text{False.}$
- (d) $\text{not } b_2 \Rightarrow \text{True.}$
- (e) $\text{not } b_3 \Rightarrow \text{False.}$
- (f) $\text{not } b_4 \Rightarrow \text{True.}$
- (g) $b_1 \text{ and } b_2 \Rightarrow \text{True and False = False.}$
- (h) $b_1 \text{ or } b_2 \Rightarrow \text{True or False = True.}$
- (i) $b_1 \text{ and } b_3 \Rightarrow \text{True and True = True.}$
- (j) $b_1 \text{ or } b_3 \Rightarrow \text{True or True = True.}$
- (k) $b_1 \text{ and } b_4 \Rightarrow \text{True and False = False.}$
- (l) $b_1 \text{ or } b_4 \Rightarrow \text{True or False = True.}$
- (m) $b_2 \text{ and } b_3 \Rightarrow \text{False and True = False.}$
- (n) $b_2 \text{ or } b_3 \Rightarrow \text{False or True = True.}$
- (o) $b_1 \text{ and } b_2 \text{ or } b_3 \Rightarrow \text{True and False or True = True.}$

(p) $b_1 \text{ or } b_2 \text{ and } b_3 \Rightarrow$

$\text{True or False and True} = \text{True}.$

(q) $b_1 \text{ and } b_2 \text{ and } b_3$

$\Rightarrow \text{True and False and True} = \text{False}.$

(r) $b_1 \text{ or } b_2 \text{ or } b_3 \Rightarrow$

$\text{True or False or True} = \text{True}.$

(s) $\text{not } b_1 \text{ and } b_2 \text{ and } b_3 \Rightarrow$

$\text{False and False and True} = \text{False}.$

(t) $\text{not } b_1 \text{ or } b_2 \text{ or } b_3 \Rightarrow$

$\text{False or True or True} = \text{True}.$

(u) $\text{not } (b_1 \text{ and } b_2 \text{ and } b_3) \Rightarrow$

$\text{not}(\text{True and False and True}) = \text{True}.$

(v) $\text{not } (b_1 \text{ or } b_2 \text{ or } b_3) \Rightarrow$

$\text{not}(\text{True or False or True}) = \text{False}.$

(w) $\text{not } b_1 \text{ and not } b_2 \text{ and not } b_3 \Rightarrow$

$\text{False and True and False} = \text{False}.$

(x) $\text{not } b_1 \text{ or not } b_2 \text{ or not } b_3 \Rightarrow$

$\text{False or True or False} = \text{True}.$

(y) not (not b1 and not b2 and not b3) =>
not(False and True and False) = True.

(z) not (not b1 or not b2 or not b3) =>
not(False or True or False) = False.

8. Express the following Boolean expressions in simpler form; that is, use fewer operators or fewer symbols. x is an integer.

(a) not (x == 2) => x != 2

(b) x < 2 or x == 2 => x <= 2

(c) not (x < y) => x >= y

(d) not (x <= y) => x > y

(e) x < 10 and x > 20 => False.

(f) x > 10 or x < 20 => True.

(g) x != 0 => True.

(h) x == 0 => False.

9. Express the following Boolean expressions in an equivalent form without the not operator. x and y are integers.

(a) not (x == y) => x != y

(b) not (x > y)	=>	x <= y
(c) not (x < y)	=>	x >= y
(d) not (x >= y)	=>	x < y
(e) not (x <= y)	=>	x > y
(f) not (x != y)	=>	x == y
(g) not (x != y)	=>	x == y
(h) not (x == y and x < 2)	=>	x != y or x >= 2
(i) not (x == y or x < 2)	=>	x != y and x >= 2
(j) not (not (x == y))	=>	x == y

10. What is the simplest tautology? => True.

11. What is the simplest contradiction? => False.

12. Write a Python program that requests an integer value from

the user. If the value is between 1 and 100 inclusive, print "OK;"

otherwise, do not print anything.

=>Note: im infering that the user wont enter a floating point number. if

this was the case then we could use a try: block.

```
number = None;
```

```
while not number = input("Please enter a  
number: ");
```

```
number = int(number);
```

```
if number <= 100 and number >= 1:
```

```
    print("Okay");
```

13. Write a Python program that requests an integer value from

the user. If the value is between 1 and 100 inclusive, print "OK;"

otherwise, print "Out of range."

```
=>number = None;
```

```
while not number:
```

```
    number = input("Please enter a number: ");
```

```
number = int(number);
```

```
if number <= 100 and number >= 1:
```

```
    print("Okay");
```

else:

```
print("Out of Range!");
```

14. Write a Python program that allows a user to type in an

English day of the week (Sunday, Monday, etc.). The program

should print the Spanish equivalent, if possible.

=>Please note that im infering the question asks for elifs, because we

can use match, case: as well.

```
day = None;
```

```
while not day:
```

```
    day = input("Please Enter a week day: ");
```

```
if day == "monday":
```

```
    print("Monday is lunes in spanish!");
```

```
elif day == "tuesday":
```

```
    print("Tuesday is martes in spanish!");
```

```
elif day == "wednesday":
```



```
print("Wednesday is miércoles in spanish!");  
elif day == "thursday":  
    print("Thursday is jueves in spanish!");  
elif day == "friday":  
    print("Friday is viernes in spanish!");  
elif day == "saturday":  
    print("Saturday is sábado in spanish!");  
elif day == "sunday":  
    print("Sunday is domingo in spanish!");  
else:  
    print("You did not enter a week day!");
```

15. Consider the following Python code fragment:

```
# i, j, and k are numbers
```

```
if i < j:
```

```
    if j < k:
```

```
        i = j
```

```
    else: j = k
```

else:

if j > k:

j = i

else:

i = k

print("i =", i, " j =", j, " k =", k)

What will the code print if the variables i, j, and k have the following values?

(a) i is 3, j is 5, and k is 7 => prints: i = 5 j = 5 k = 7

(b) i is 3, j is 7, and k is 5 => prints: i = 3 j = 5 k = 5

(c) i is 5, j is 3, and k is 7 => prints: i = 7 j = 3 k = 7

(d) i is 5, j is 7, and k is 3 => prints: i = 5 j = 3 k = 3

(e) i is 7, j is 3, and k is 5 => prints: i = 5 j = 3 k = 5

(f) i is 7, j is 5, and k is 3 => prints: i = 7 j = 7 k = 3

16. Consider the following Python program that prints one line

of text:

```
val = int(input())
```

```
if val < 10:
    if val != 5:
        print("wow ", end="")
    else:
        val += 1
else:
    if val == 17:
        val += 10
    else:
        print("whoa ", end="")
print(val)
```

What will the program print if the user provides the following

input?

- | | | |
|--------|----|---------------------------|
| (a) 3 | => | wow. |
| (b) 21 | => | whoa. |
| (c) 5 | => | prints nothing, val = 6. |
| (d) 17 | => | prints nothing, val = 27. |

(e) -5

=>

WOW.

17. Consider the following two Python programs that appear very similar:

A:

```
n = int(input())
if n < 1000:
    print('*', end='')
if n < 100:
    print('*', end='')
if n < 10:
    print('*', end='')
if n < 1:
    print('*', end='')
print()
```

B:

```
n = int(input())
if n < 1000:
```

```
print('*', end='')  
elif n < 100:  
    print('*', end='')  
elif n < 10:  
    print('*', end='')  
elif n < 1:  
    print('*', end='')  
print()
```

How do the two programs react when the user provides the

following inputs?

- | | | |
|----------|----|-----------------------|
| (a) 0 | => | A:****, B:* |
| (b) 1 | => | A:***, B:* |
| (c) 5 | => | A:***, B:* |
| (d) 50 | => | A:**, B:* |
| (e) 500 | => | A:*, B:* |
| (f) 5000 | => | A:nothing, B:nothing. |

Why do the two programs behave as they do?

=>because A checks for each if statement, and if more than one of them

is true it will print astrix more than one time.

but B has a $n < 1000$ as the first if and the other statements as elif so

even tho numbers are small, but as long as they are smaller than 1000

the first if activates and other elifs wont act.

18. Write a Python program that requests five integer values

from the user. It then prints the maximum and minimum values

entered. If the user enters the values 3, 2, 5, 0, and 1, the

program would indicate that 5 is the maximum and 0 is the

minimum. Your program should handle ties properly; for

example, if the user enters 2, 4, 2, 3, and 3, the program should

report 2 as the minimum and 4 as maximum.

=>Please note that im infering we have no knowledge of min() and

max() existing...

```
max = None;
```

```
min = None;
```

```
for i in range(5):
```

```
    number = int(input('Please enter a number: '));
```

```
    if i == 0:
```

```
        max = number;
```

```
        min = number;
```

```
        elif(number < min):
```

```
            min = number;
```

```
        elif(number > max):
```

```
            max = number;
```

```
print("Max is:", max, "\nMin is:", min);
```

19. Write a Python program that requests five integer values from the user. It then prints one of two things: if any of the values entered are duplicates, it prints "DUPLICATES"; otherwise, it prints "ALL UNIQUE".

=>

```
repeat = 0;
temp = None;
for i in range(5):
    number = int(input("Please Enter a integer: "));
    if number == temp:
        repeat = 1;
    else:
        temp = number;
if repeat:
    print("DUPLICATES");
```


else:

print("ALL UNIQUE");