Fundamentals of Programming Python
chapter 4, section 14 Exercises with Answers
Second Homework 2023/02/20

1. What possible values can a Boolean expression have?
answer: A variable of the primitive data type boolean can have two values: true and false
(Boolean literals). or off. Boolean expressions use relational and logical operators.
The result of a Boolean expression is either true or false.

2. Where does the term Boolean originate?
Answer: George Boole, an English mathematician in the 19th century, developed "Boolean Logic" in order
to combine certain concepts and exclude certain concepts when searching databases.

3. What is an integer equivalent to True in Python?
Answer:In Python, the integer 0 is always False , while every other number, including negative
numbers, are True.

4. What is an integer equivalent to False in Python?
Answer: integer 0

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

6. 6. Given the following definitions:
x, y, z = 3, 5, 7
evaluate the following Boolean expressions:

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

7. Given the following definitions:
x, y = 3, 5
b1, b2, b3, b4 = True, False, x == 3, y < 3
evaluate the following Boolean expressions:

```
(a) b3                                => True
(b) b4                                => False
(c) not b1                            => False
(d) not b2                            => True
(e) not b3                            => False
(f) not b4                            => True
(g) b1 and b2                         => False
(h) b1 or b2                          => True
(i) b1 and b3                         => True
(j) b1 or b3                          => True
```

```
(k) b1 and b4                          => False
(l) b1 or b4                           => True
(m) b2 and b3                          => False
(n) b2 or b3                           => True
(o) b1 and b2 or b3                    => True
(p) b1 or b2 and b3                    => True
(q) b1 and b2 and b3                   => False
(r) b1 or b2 or b3                     => True
(s) not b1 and b2 and b3               => False
(t) not b1 or b2 or b3                 => True
(u) not (b1 and b2 and b3)             => True
(v) not (b1 or b2 or b3)               => False
(w) not b1 and not b2 and not b3       => False
(x) not b1 or not b2 or not b3         => True
(y) not (not b1 and not b2 and not b3) => True
(z) not (not b1 or not b2 or not b3)   => 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?
Answer: True

11. What is the simplest contradiction?
Answer: 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.

```
number = None;
while not number:
    number = input("Please enter an integer number: ");
number = int(number);
```

```
if number <= 100 and number >= 1:
    print("Ok");
```

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 an integer number: ");
number = int(number);
if (number <= 100 and number >= 1):
    print ("Ok");
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.

```
day = None;
while not day:
    day = input("Please Enter an English day of the week: ");
if day == "monday":
    print("Spanish : lunes");
elif day == "tuesday":
    print("Spanish : martes");
elif day == "wednesday":
    print("Spanish : miércoles");
elif day == "thursday":
    print("Spanish : jueves");
elif day == "friday":
    print("Spanish : viernes");
elif day == "saturday":
    print("Spanish : sábado");
elif day == "sunday":
    print("Spanish : domingo");
else: print("You didn't an English day of the week!");
```

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 => it will print: i = 5 j = 5 k = 7
(b) i is 3, j is 7, and k is 5 => it will print: i = 3 j = 5 k = 5
(c) i is 5, j is 3, and k is 7 => it will print: i = 7 j = 3 k = 7
(d) i is 5, j is 7, and k is 3 => it will print: i = 5 j = 3 k = 3
(e) i is 7, j is 3, and k is 5 => it will print: i = 5 j = 3 k = 5

(f) i is 7, j is 5, and k is 3 => it will print: 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:

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

```
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?
Answer:
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 that 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 = 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");
```