HW_WEEK 2ND_ESFAND_#HOOMAN_ILEAYI#

1. What possible values can a Boolean expression have?

.A Boolean expression can have two possible values: true or false

2. Where does the term Boolean originate?

The term Boolean originated from the 19th century mathematician George Boole, who developed a system of logic that is now known as Boolean algebra

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

1_ONE

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

0_ZERO

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

FALSE

6. Given the following definitions:

x, y, z = 3, 5, 7

evaluate the following Boolean expressions:

(a) x == 3

(b) x < y

(c) x >= y

(d) x <= y

(e) x != y - 2

(f) x < 10

(g) x >= 0 and x < 10

(h) x < 0 and x < 10

(i) x >= 0 and x < 2

(j) x < 0 or x < 10

(k) x > 0 or x < 10

(l) x < 0 or x > 10

(a) True

(b) True

(c) False

(d) True

(e) True

(f) True

(g) True

(h) False

(i) False

(j) True

(k) True

(l) 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

(b) b4

(c) not b1

(d) not b2

(e) not b3

(f) not b4

(g) b1 and b2

(h) b1 or b2

(i) b1 and b3

(j) b1 or b3

(k) b1 and b4

(l) b1 or b4

(m) b2 and b3

(n) b2 or b3

(o) b1 and b2 or b3

(p) b1 or b2 and b3

(q) b1 and b2 and b3

(r) b1 or b2 or b3

(s) not b1 and b2 and b3

(t) not b1 or b2 or b3

(u) not (b1 and b2 and b3)

(v) not (b1 or b2 or b3)

(w) not b1 and not b2 and not b3

(x) not b1 or not b2 or not b3

(y) not (not b1 and not b2 and not b3)

(z) not (not b1 or not b2 or not b3)

(a) True

(b) False

(c) False

(d) True

(e) False

(f) True

(g) False

(h) True

(i) True

(j) True

(k) False

(l) True

(m) False

(n) True

(o) True

(p) False

(q) False

(r) True

(s) False

(t) True

(u) False

(v) True

(w) False

(x) True

(y) False

(z) True

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)

(b) x < 2 or x == 2

(c) not (x < y)

(d) not (x <= y)

(e) x < 10 and x > 20

(f) x > 10 or x < 20

(g) x != 0

(h) x == 0

(a) $x \neq 2$

(b) $x \leq 2$

(c) $x \geq y$

(d) x > y

(e) $x \leq 10$ or $x \geq 20$

(f) 10 < x < 20

(g) $x \neq 0$

(h) x = 0

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

are integers.

(a) not (x == y)

(b) not (x > y)

(c) not (x < y)

(d) not (x >= y)

(e) not (x <= y)

(f) not (x != y)

(g) not (x != y)

(h) not (x == y and x < 2)

(i) not (x == y or x < 2)

(j) not (not (x == y))

(a) x ≠ y

(b) x ≤ y

(c) x ≥ y

(d) x < y

(e) x > y

(f) x == y

(g) x == y

(h) x != y or x ≥ 2

(i) x != y and x ≥ 2

(j) x == y

10. What is the simplest tautology?

".The simplest tautology is "A statement is true if and only if it is true

11. What is the simplest contradiction?

The simplest contradiction is "This statement is 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.

Get user input #

num = int (input ("Please enter an integer between 1 and 100: "))

Check if number is between 1 and 100 inclusive #

:if num >= 1 and num <= 100

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.

Request an integer value from the user #

num = int (input ("Please enter an integer value: "))

Check if the value is between 1 and 100 inclusive #

:if num >= 1 and num <= 100

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.

Program to print Spanish equivalent of an English day of the week #

Dictionary containing English and Spanish days of the week #

,'days_dict = {'Monday': 'Lunes', 'Tuesday': 'Martes', 'Wednesday': 'Miércoles

,'Thursday': 'Jueves', 'Friday': 'Viernes'

{'Saturday': 'Sábado', 'Sunday':'Domingo'

Get input from user #

day = input ("Please enter an English day of the week: ")

Check if day is present in dictionary #

:if day in days_dict

Print the Spanish equivalent of the day #

print ("The Spanish equivalent of", day,"is", days_dict[day])

:else

If not present, print appropriate message #

print ("Sorry! No Spanish equivalent found for", 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

(b) i is 3, j is 7, and k is 5

(c) i is 5, j is 3, and k is 7

(d) i is 5, j is 7, and k is 3

(e) i is 7, j is 3, and k is 5

(f) i is 7, j is 5, and k is 3

(a) i = 5 j = 7 k = 3

(b) i = 5 j = 3 k = 7

(c) i = 3 j = 5 k = 7

(d) i = 3 j = 7 k = 5

(e) i = 5 j = 3 k = 7

(f) i = 7 j = 5 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

(b) 21

(c) 5

(d) 17

(e) -5

(a) wow 3

(b) whoa 21

(c) 5

(d) whoa 27

(e) wow -5

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

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

How do the two programs react when the user provides the following inputs?

(a) 0

(b) 1

(c) 5

(d) 50

(e) 500

(f) 5000

* :W(a) 0: Program 1: *** Program 2

* :(b) 1: Program 1: ** Program 2

* :(c) 5: Program 1: ** Program 2

* :(d) 50: Program 1: ** Program 2

* :(e) 500: Program 1: * Program 2

:(f) 5000: Program 1: * Program 2

The two programs behave differently because the first program uses an if statement, which will only execute the code within its block if the condition is true. The second program uses an elif statement, which will only execute the code within its block if all of the previous conditions are false and the current condition is true. This means that in the first program, all of the conditions will be checked even if one of them is true, while in the second program, only one condition will be checked.

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.

Program to find the maximum and minimum values entered by the user #

Initializing variables #

max_value = 0

min_value = 0

Taking input from the user #

print ("Enter five integer values:")

num1 = int (input ())

num2 = int (input ())

num3 = int (input ())

num4 = int (input ())

num5 = int (input ())

.Comparing the values and storing them in max_value and min_value variables #

:if (num1 >= num2) and (num1 >= num3) and (num1 >= num4) and (num1 >= num5)

max_value = num1

:elif (num2 >= num1) and (num2 >= num3) and (num2 >= num4) and (num2 >= num5)

max_value = num2

:elif (num3 >= num1) and (num3 >= num2) and (num3 >= num4) and (num3 >= num5)

max_value = num3 # Comparing maximum value. # Comparing minimum .value

) if (num1 <= num2) and (num1 <= num3) and (num1 <= num4) and

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".

[] = values

:(٥) for i in range

value = int (input ("Please enter an integer value: "))

values. append(value)

:if len(values) != len(set(values))

print("DUPLICATES")

:else

```
print ("ALL UNIQUE")
```