Week2-Hw2-40113841054010

1.What possible values can a Boolean expression have?

Arithmetic expressions evaluate to numeric values; a Boolean expression, sometimes called a predicate,

may have only one of two possible values: false or true.

2. Where does the term Boolean originate?

The term Boolean comes from the name of the British mathematician George Boole.

A branch of discrete mathematics called Boolean algebra is dedicated to the study of the properties and the manipulation of logical expressions.

3. What is an integer equivalent to True in Python? 1

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 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 x > 20

(f) x > 10 or x < 20 10 < x < 20

(g) x != 0 not (x == 0)

(h) x == 0 not (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) 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.

number=int(input("please enter a number bertween 1 and 100 ="))

if 1<number<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.”

number=int(input("please enter a number bertween 1 and 100"))

if 1<number<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.

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 i=5 , j=5 , k=7

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

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

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

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

(f) i is 7, j is 5, and k is 3 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 6

(d) 17 27

(e) -5 wow

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

|  |  |
| --- | --- |
| 1  n = int(input())  if n < 1000:  print('\*', end='')  if n < 100:  print('\*', end='')  if n < 10:  print('\*', end='')  if n < 1:  print('\*', end='')  print() | 2  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 (1) \*\*\*\* (2) \*

(b) 1 (1) \*\*\* (2) \*

(c) 5 (1) \*\*\* (2) \*

(d) 50 (1) \*\* (2) \*

(e) 500 (1) \* (2) \*

(f) 5000 (1) (2) \*

Why do the two programs behave as they do?

because 1 checks for each if statement, and if more than one of them

is true it will print astrix more than one time.

but 2 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.

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".

n1=int(input("num1="))

n2=int(input("num2="))

n3=int(input("num3="))

n4=int(input("num4="))

n5=int(input("num5="))

dublicated=0

if n1==n2:

dublicated+=1

if n1==n3:

dublicated+=1

if n1==n4:

dublicated+=1

if n1==n5:

dublicated+=1

if n2==n3:

dublicated+=1

if n2==n4:

dublicated+=1

if n2==n5:

dublicated+=1

if n3==n4:

dublicated+=1

if n3==n5:

dublicated+=1

if n4==n5:

dublicated+=1

if dublicated !=0:

print("DUPLICATES ")

else:

print("ALL UNIQUE ")