FLIP ROBO INTERNSHIP WORK

NAME – SUBHAM

Batch – DS2401

Answer 1) C) 15

The initial call is func(30,70). Since a is not equal to zero else statement follows and 75 modulus 30 is 30 and b(70) becomes a(30) ->(75%30,30) which becomes (15,30). Now it again checks if a is equal to zero or not until a equals zero. Now a is not zero so again else statement follows and 30 modulus 15 is 0 and b(30) becomes a(15) ->(30%15,15) becomes (0,15). It checks if a=0 ,since a=0 it will return b value which is 15 now.

Answer 2) b) filter

numbers = (4, 7, 19, 2, 89, 45, 72, 22) - Defines a tuple of numbers.

sorted_numbers = sorted(numbers) - Creates a new list containing the sorted numbers.

even = lambda a: a % 2 == 0 - Defines a lambda function even that returns True for even numbers and False for odd numbers.

even_numbers = filter(even, sorted_numbers) - Uses the filter function to filter out the even numbers from sorted_numbers.

print(type(even_numbers)) - Prints the type of the object resulting from the filter operation. Therefore, the output of the code will be: <class 'filter'>

Answer 3) a) Tuple

When *args is used in function definition , it collects any additional positional arguments into a tuple.

Answer 4) d) error

This will result in a typeError because + operator is not supported for set. Sets are unordered collection of data. They are not indexed , immutable , non duplicatable)

Answer 5) a) raise

Answer 6) c) datetime

Answer 7) c) 208

4**3 means 4^3 -> 64. 7+5 -> 12 and 1+1 -> 2 so 12**2 means 12^2 -> 144 and 64+144 = 208

Answer 8) b) strftime

Answer 9) b) immutable

Immutable meaning its elements cannot be changed after the creation of tuple.

Answer 10) a) range()

Answer 11) c) Lambda function

Answer 12) c) Both A and B

Answer 13) b) dump() method

Answer 14) a) load() method

Answer 15) d) all of the above

Answer 16) d) both a and b

Answer 17) d) captains = { }

Answer 18) b) Captains["Enterprise"] = "Picard"

Captains["Voyager"] = "Janeway"

Captains["Defiant"] = "Sisko"

Answer 19) b) for ship , captain in captains.items():

Print(f" The {ship} is captained by {captain}. ")

Output would be:

The Enterprise is captained by Picard.

The Voyager is captained by Janeway.

The Defiant is captained by Sisko.

The Discovery is captained by unkown.

Answer 20) c) del captains["Discovery"]