19BCE245 2CS404 4 20/02/2021

1 [illegible]

Correct
~~right~~ ~~code~~ :

Q-1 A.)
```
numbers = [ 11, 33, 55, 37, 55, 79, 38 ]
for num in numbers:
    if num %2 ==0 :
        print("This list contains an even number")
else:
    print(" this list doesn not contain even number")
```

The program code given in question will give error as there is single quote before the closing bracket in last line. changing it to double quote (written code as above) will give output :

The list contains an even number.

B.) Correct ~~right~~ code :
```
print(" Absolute value : ", math.fabs(-65))
print ("Absolute value : ", abs(-65))
```

here we have to write abs without math as there is no methode /module in math named as 'abs'

Output :
```
65.0
65
```

C.) The given code in the question is correct (no change)
```
vehicles = set (['Bike', 'Bus', 'Car', 'Scooter'])
vehicles.add ('Train')
vehicles.update ( ['Truck', 'Rickshaw', "Train"])
print(vehicles)
```

VISION

19BCE245 2CS404 4 20/02/2021 [illegible]

2

output :
set (['Bus', 'Scooter', 'Bike', 'Train', 'Truck', 'Car', 'Rickshow'])

4.) The given code in the question is correct till we give inputs in the range of [0,100]
→ but it will give error if we ~~imp~~ give input numbers otherthan it's range.
→ If we give input in range [0, 100] then nothing will be on output screen.
→ But if we give input from out of it's range like 111, -1 then it will give AssertionError and it will also print the message which we gave i.e. "Number must be in the range of 0 to 100" along with that error name.

Q - 2.) Python code to generate armstrong numbers in given range with input from command line arg.

```
→ import sys

lower = sys.argv[1]   # getting lower limit from cmd line arg.
upper = sys.argv[2]   # getting upper limit from cmd line arg.

for num in range(lower, upper + 1):
    sum = 0
    temp = num
    while temp>0:
        digit = temp %10
        sum += digit ** 3
        temp // = 10
    if num == sum:
        ~~print print (sum)~~
        print(num)
```

Q-3. python program to reverse the string enatered by user using recursive function.

→
```
def reverse_string(string):
    if len(string) == 0:
        return string
    else:
        return reverse_string(string[1:]) + string[0]

original_string = str(input(" Enter the string to be reversed: "))
print(reverse_string(original_string))
```

Q-4) (1.)

~~impo~~

```
import random

data_dict = {}          # dictionary to store main data.

item_name = []          # list of item names.
item_value = []         # list of item values.

#assigning item names.

~~for item~~
```

19BCE245 2CS404 4 20/02/2021 [illegible]

4

```
#assigning item names.

for i in range(0,30):
    item_name.append("item" + str(i+1))

#now item_name = ["item1", item2", ... , "item 30"].

#assigning random values to item_value.

for i in range(0,30):
    item_value.append(round(random.uniform((10.0,500.0)),1))

#now item_value will be filled with random values
#from 10.0 to 500.0 (with 1 decimal point.)
```

Que 4-2
```
order_dict = {}

order_dict["order1"] = {"item5": ~~data_dic~~ data_dict["item5"]*2,
                        "item17": data_dict["item17"]*5,
                        "item28": data_dict["item28"]*10,
                        "item 2": data_dict["item2"]*1}

order_dict["order 2"] = {"item25": data_dict["item25"]*1,
                         "item9": data_dict["item9"]* 2,
                         "item 1": data_dict["item1"]*1 }

order_dict["order1"]["total"]= order_dict["order1"]["item 5"]
                             + order_dict["order1"]["item17"]
                             + order_dict["order1"]["item 28"]
                             + order_dict["order1"]["item 2"]
```

VISION

19BCE245 2CS404 4 20/02/2021

5 [illegible]

```
print("Total for order1 is:",order_dict["order1"]["total"])

order_dict["order2"]["total"] = order_dict["order2"]["item25"]
                              +order_dict["order2"]["item9"]
                              +order_dict["order2"]["item1"]

print("Total for order2 is :",order_dict["order2"]["total"])
```

Q-4-3
```
order_id = "0" + str( random.randint(0, 999999)).zfill(6,"0")

~~for i in range~~
~~print(" order_dict["order1"]["item~~
  for i in range(4):
      print("~~order_dict["oreder1"]~~ "item"+i,
            order_dict["order1"]["item" +i] x Quantity,
            Order_dict["order1"]["item"] ["total"]);

  for i in range(3):
      print(" order_dict ["order 2"]["item"+i]",
            order_dict ["order2"]["item"+i] x Quantity
            Order_dict ["order2"]["total"] )
```