

Student name: diana Sameer mkanna  
University ID: 2113

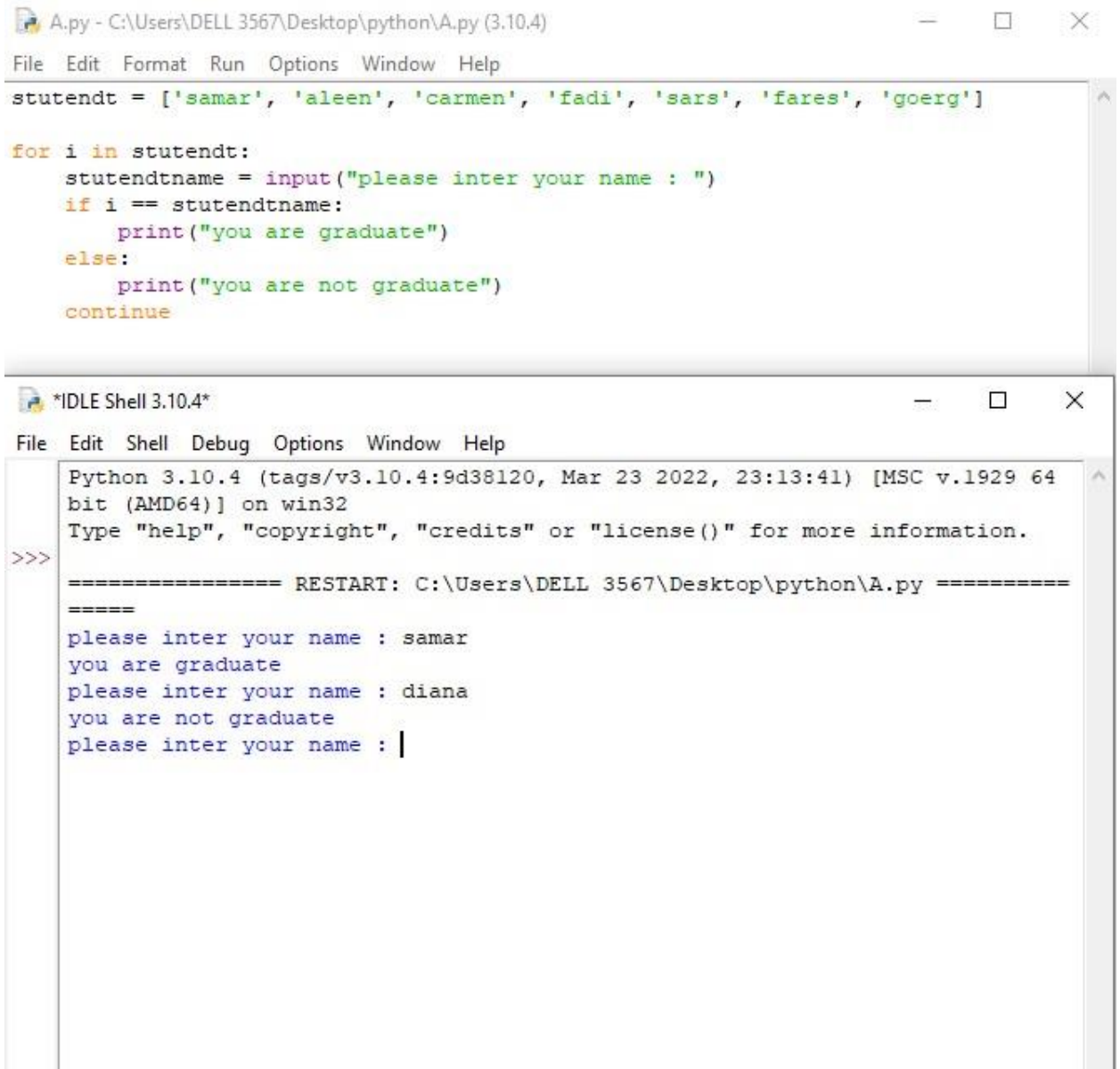
Question 1: **Python Basics?**

A-Define a list that contain the names of graduated students" 5 students at least":  
Create a program that accept student name and prints if the user is graduated or not.

Answer:

```
stutendt = ['samar', 'aleen', 'carmen', 'fadi', 'sars', 'fares', 'goerg']

for i in stutendt:
    stutendtname = input("please inter your name : ")
    if i == stutendtname:
        print("you are graduate")
    else:
        print("you are not graduate")
    continue
```



The image shows two windows from the Python IDLE environment. The top window, titled 'A.py - C:\Users\DELL 3567\Desktop\python\A.py (3.10.4)', contains a Python script. The script defines a list 'stutendt' with names 'samar', 'aleen', 'carmen', 'fadi', 'sars', 'fares', and 'goerg'. It then uses a 'for' loop to iterate over this list. Inside the loop, it prompts the user to 'please inter your name : ' and checks if the input matches the current name in the list. If it matches, it prints 'you are graduate'; otherwise, it prints 'you are not graduate' and continues the loop.

```
A.py - C:\Users\DELL 3567\Desktop\python\A.py (3.10.4)
File Edit Format Run Options Window Help
stutendt = ['samar', 'aleen', 'carmen', 'fadi', 'sars', 'fares', 'goerg']

for i in stutendt:
    stutendtname = input("please inter your name : ")
    if i == stutendtname:
        print("you are graduate")
    else:
        print("you are not graduate")
    continue
```

The bottom window, titled '\*IDLE Shell 3.10.4\*', shows the execution of the script. It displays the Python version and architecture information, followed by a restart message for the file 'A.py'. The execution then shows the prompts and outputs for the first two names: 'samar' (graduate) and 'diana' (not graduate). The prompt for the third name is shown with a cursor.

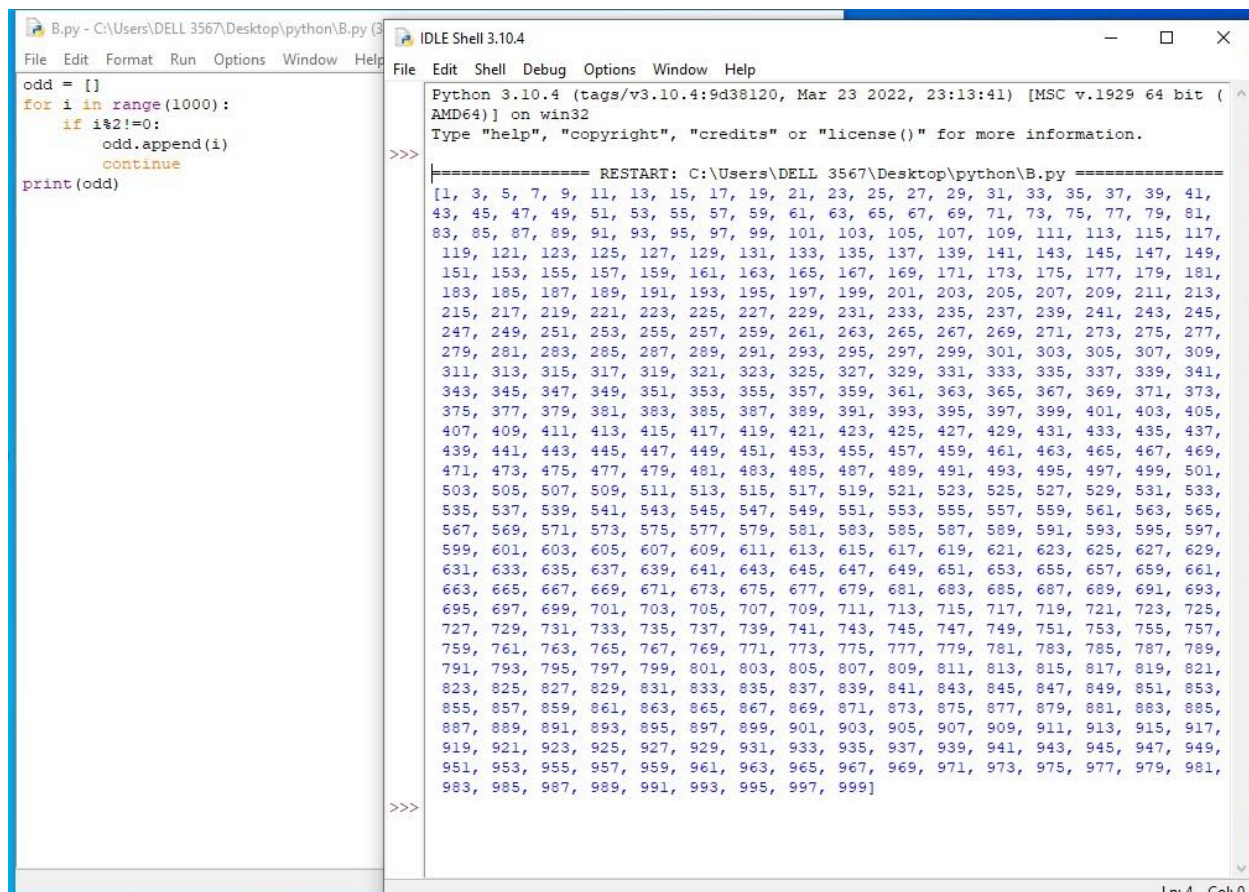
```
*IDLE Shell 3.10.4*
File Edit Shell Debug Options Window Help
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64
bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\DELL 3567\Desktop\python\A.py =====
=====
please inter your name : samar
you are graduate
please inter your name : diana
you are not graduate
please inter your name : |
```

B- Generate and print a list of odd numbers from 1 to 1000.

**Tips:** "List Comprehension"

Answer:

```
odd = []
for i in range(1000):
    if i%2!=0:
        odd.append(i)
        continue
print(odd)
```



The screenshot displays a Python IDE with two windows. The left window, titled 'B.py - C:\Users\DELL 3567\Desktop\python\B.py (3', contains the following code:

```
odd = []
for i in range(1000):
    if i%2!=0:
        odd.append(i)
        continue
print(odd)
```

The right window, titled 'IDLE Shell 3.10.4', shows the output of the script. It begins with the Python version and architecture information: 'Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)] on win32'. It then prompts the user to type 'help', 'copyright', 'credits' or 'license()' for more information. After pressing Enter, it displays the output of the script: a list of odd numbers from 1 to 999, with the last element being 999. The output is wrapped in a restart message: '===== RESTART: C:\Users\DELL 3567\Desktop\python\B.py ====='.

```
>>>
===== RESTART: C:\Users\DELL 3567\Desktop\python\B.py =====
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41,
43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 69, 71, 73, 75, 77, 79, 81,
83, 85, 87, 89, 91, 93, 95, 97, 99, 101, 103, 105, 107, 109, 111, 113, 115, 117,
119, 121, 123, 125, 127, 129, 131, 133, 135, 137, 139, 141, 143, 145, 147, 149,
151, 153, 155, 157, 159, 161, 163, 165, 167, 169, 171, 173, 175, 177, 179, 181,
183, 185, 187, 189, 191, 193, 195, 197, 199, 201, 203, 205, 207, 209, 211, 213,
215, 217, 219, 221, 223, 225, 227, 229, 231, 233, 235, 237, 239, 241, 243, 245,
247, 249, 251, 253, 255, 257, 259, 261, 263, 265, 267, 269, 271, 273, 275, 277,
279, 281, 283, 285, 287, 289, 291, 293, 295, 297, 299, 301, 303, 305, 307, 309,
311, 313, 315, 317, 319, 321, 323, 325, 327, 329, 331, 333, 335, 337, 339, 341,
343, 345, 347, 349, 351, 353, 355, 357, 359, 361, 363, 365, 367, 369, 371, 373,
375, 377, 379, 381, 383, 385, 387, 389, 391, 393, 395, 397, 399, 401, 403, 405,
407, 409, 411, 413, 415, 417, 419, 421, 423, 425, 427, 429, 431, 433, 435, 437,
439, 441, 443, 445, 447, 449, 451, 453, 455, 457, 459, 461, 463, 465, 467, 469,
471, 473, 475, 477, 479, 481, 483, 485, 487, 489, 491, 493, 495, 497, 499, 501,
503, 505, 507, 509, 511, 513, 515, 517, 519, 521, 523, 525, 527, 529, 531, 533,
535, 537, 539, 541, 543, 545, 547, 549, 551, 553, 555, 557, 559, 561, 563, 565,
567, 569, 571, 573, 575, 577, 579, 581, 583, 585, 587, 589, 591, 593, 595, 597,
599, 601, 603, 605, 607, 609, 611, 613, 615, 617, 619, 621, 623, 625, 627, 629,
631, 633, 635, 637, 639, 641, 643, 645, 647, 649, 651, 653, 655, 657, 659, 661,
663, 665, 667, 669, 671, 673, 675, 677, 679, 681, 683, 685, 687, 689, 691, 693,
695, 697, 699, 701, 703, 705, 707, 709, 711, 713, 715, 717, 719, 721, 723, 725,
727, 729, 731, 733, 735, 737, 739, 741, 743, 745, 747, 749, 751, 753, 755, 757,
759, 761, 763, 765, 767, 769, 771, 773, 775, 777, 779, 781, 783, 785, 787, 789,
791, 793, 795, 797, 799, 801, 803, 805, 807, 809, 811, 813, 815, 817, 819, 821,
823, 825, 827, 829, 831, 833, 835, 837, 839, 841, 843, 845, 847, 849, 851, 853,
855, 857, 859, 861, 863, 865, 867, 869, 871, 873, 875, 877, 879, 881, 883, 885,
887, 889, 891, 893, 895, 897, 899, 901, 903, 905, 907, 909, 911, 913, 915, 917,
919, 921, 923, 925, 927, 929, 931, 933, 935, 937, 939, 941, 943, 945, 947, 949,
951, 953, 955, 957, 959, 961, 963, 965, 967, 969, 971, 973, 975, 977, 979, 981,
983, 985, 987, 989, 991, 993, 995, 997, 999]
>>>
```

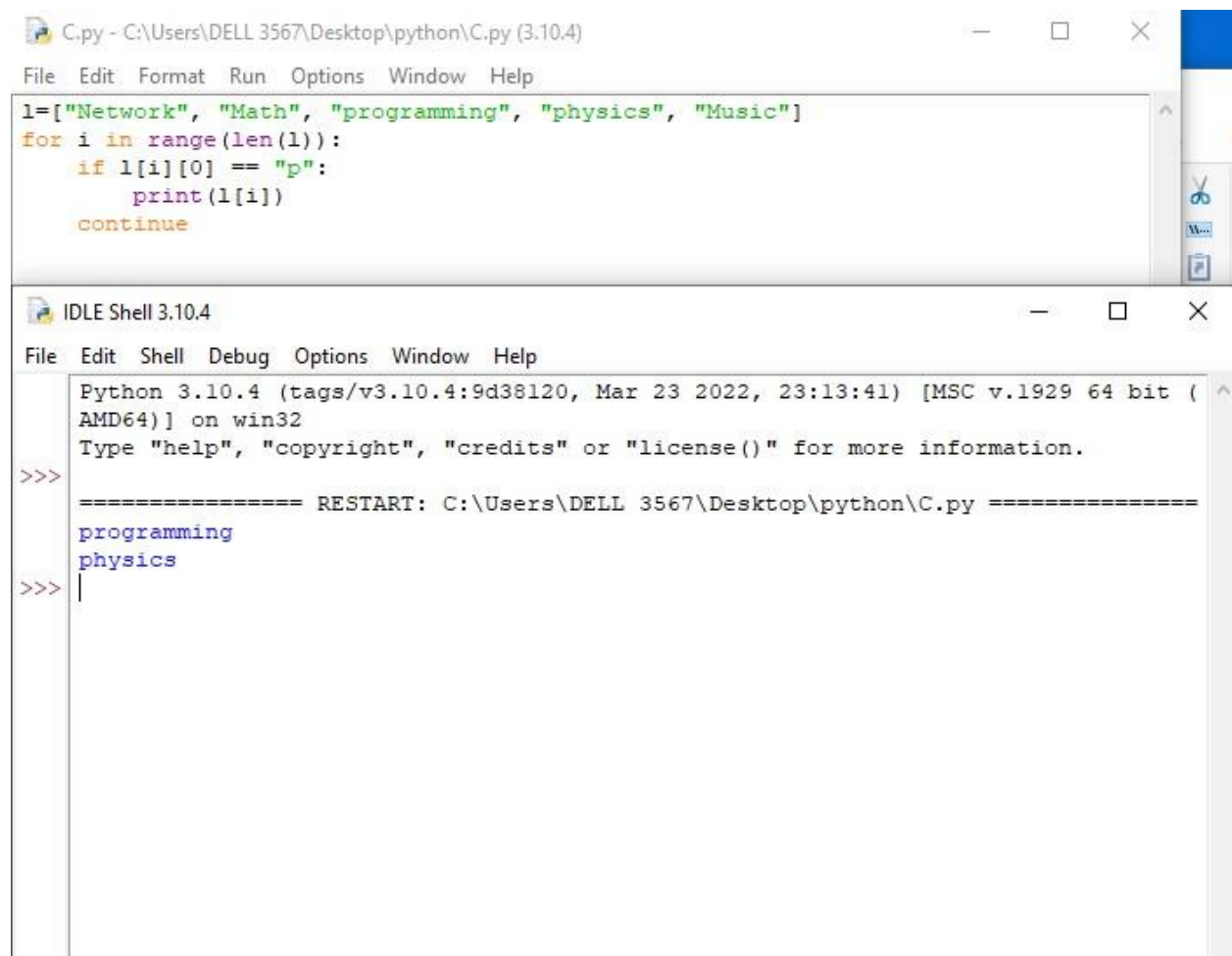
C- L:['Network', 'Math', 'Programming', 'Physics', 'Music']

In this exercise, you will implement a Python program that reads the items of the previous list and identifies the **items that starts with 'P' letter**, then print it on screen.

**Tips:** using loop, list 'len ()' method

Answer:

```
l=["Network", "Math", "programming", "physics", "Music"]
for i in range(len(l)):
    if l[i][0] == "p":
        print(l[i])
    continue
```



The screenshot displays a Python IDE with two windows. The top window, titled 'C.py - C:\Users\DELL 3567\Desktop\python\C.py (3.10.4)', contains the following code:

```
l=["Network", "Math", "programming", "physics", "Music"]
for i in range(len(l)):
    if l[i][0] == "p":
        print(l[i])
    continue
```

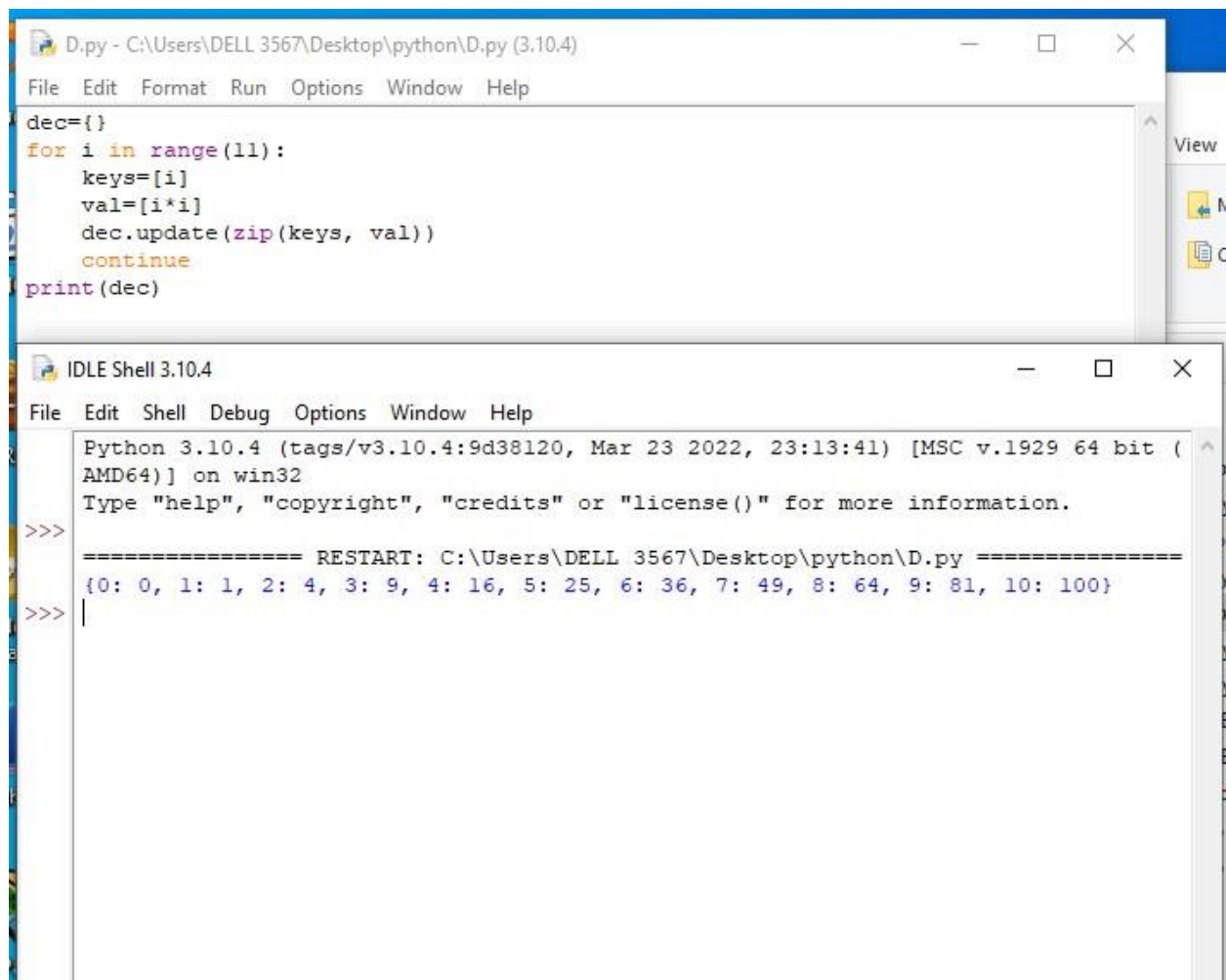
The bottom window, titled 'IDLE Shell 3.10.4', shows the execution output. It starts with the Python version and architecture information, followed by a restart message and the output of the script:

```
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\DELL 3567\Desktop\python\C.py =====
programming
physics
>>> |
```

D:Using Dictionary comprehension, Generate this dictionary  
d1:1,2,4,3:9,4: 16,5:25,6:36,7:42,8:64,9:81,10:100}

Answer:

```
dec={}
for i in range(11):
    keys=[i]
    val=[i*i]
    dec.update(zip(keys, val))
    continue
print(dec)
```



The screenshot displays the Python IDLE 3.10.4 environment. The top window, titled 'D.py - C:\Users\DELL 3567\Desktop\python\D.py (3.10.4)', contains the following Python code:

```
dec={}
for i in range(11):
    keys=[i]
    val=[i*i]
    dec.update(zip(keys, val))
    continue
print(dec)
```

The bottom window, titled 'IDLE Shell 3.10.4', shows the execution output. It includes the standard Python startup message and a restart notice. The output of the script is:

```
>>>
===== RESTART: C:\Users\DELL 3567\Desktop\python\D.py =====
{0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100}
>>> |
```

## Question 2: Convert from decimal to binary

Write a Python program that converts a decimal number into its equivalent binary number.

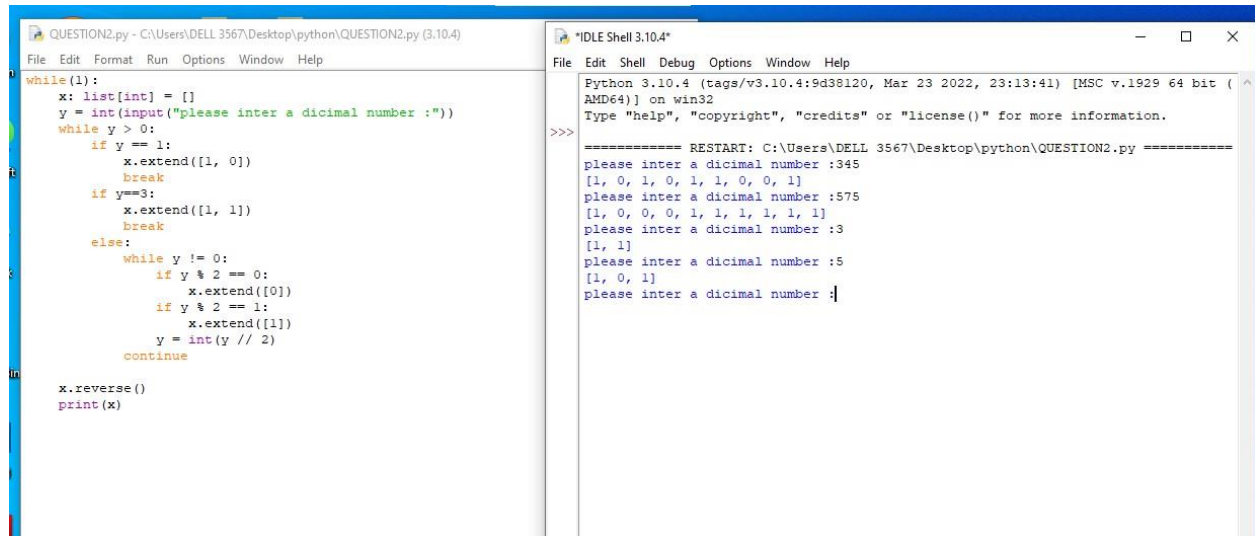
The program should start reading the decimal number from the user. Then the binary equivalent number must be calculated. Finally, the program must display the equivalent binary number on the screen.

**Tips:** use empty list to hold binary number, use loop, use % operator, use // operator, use list append method, reverse the list.

Answer:

```
while(1):
    x: list[int] = []
    y = int(input("please enter a decimal number :"))
    while y > 0:
        if y == 1:
            x.extend([1, 0])
            break
        if y==3:
            x.extend([1, 1])
            break
        else:
            while y != 0:
                if y % 2 == 0:
                    x.extend([0])
                if y % 2 == 1:
                    x.extend([1])
                y = int(y // 2)
            continue

    x.reverse()
    print(x)
```



The image shows a screenshot of a Python IDE with two windows. The left window, titled 'QUESTION2.py', contains a Python program that converts a binary number (entered as a string) into a decimal number. The program uses a list to store the binary digits, processes them from right to left, and calculates the decimal value using powers of 2. The right window, titled 'IDLE Shell 3.10.4', shows the execution of the program. It displays the prompt 'please enter a decimal number :', followed by the user input '345', and then the output of the program: '[1, 0, 1, 0, 1, 1, 0, 0, 1]'. The program then prompts for the decimal number, and the user enters '575'. The program then prompts for the decimal number, and the user enters '3'. The program then prompts for the decimal number, and the user enters '5'. The program then prompts for the decimal number, and the user enters '1'. The program then prompts for the decimal number, and the user enters '1'.

```
QUESTION2.py - C:\Users\DELL 3567\Desktop\python\QUESTION2.py (3.10.4)
File Edit Format Run Options Window Help

while(1):
    x: list[int] = []
    y = int(input("please enter a decimal number :"))
    while y > 0:
        if y == 1:
            x.extend([1, 0])
            break
        if y == 3:
            x.extend([1, 1])
            break
        else:
            while y != 0:
                if y % 2 == 0:
                    x.extend([0])
                if y % 2 == 1:
                    x.extend([1])
                y = int(y // 2)
            continue
    x.reverse()
    print(x)

IDLE Shell 3.10.4
File Edit Shell Debug Options Window Help

Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>>
===== RESTART: C:\Users\DELL 3567\Desktop\python\QUESTION2.py =====
please enter a decimal number :345
[1, 0, 1, 0, 1, 1, 0, 0, 1]
please enter a decimal number :575
[1, 0, 0, 0, 1, 1, 1, 1, 1, 1]
please enter a decimal number :3
[1, 1]
please enter a decimal number :5
[1, 0, 1]
please enter a decimal number :1
```

### Question 3: "Working with Files" Quiz Program"

Type python quiz program that takes a text or json or csv file as input for (20 (Questions, Answers)). I tasks the questions and finally computes and prints user results and store user name and result in separate file.

Answer:

```
import json

game = {"one": [{"q1": "what is your name name ? ", "res1": "name"}],
        "tow": [{"q2": "what is 5*8 ? ", "res2": "40"}],
        "three": [{"q3": "what is 32/2 ? ", "res3": "16"}],
        "four": [{"q4": "what is 125+25 ? ", "res4": "150"}],
        "five": [{"q5": "what is 14+6 ? ", "res5": "20"}],
        "six": [{"q6": "what is 4+6 ? ", "res6": "10"}],
        "seven": [{"q7": "what is 25+15 ? ", "res7": "40"}],
        "eight": [{"q8": "what is 5*(8-2) ? ", "res8": "30"}],
        "nine": [{"q9": "what is 30+40 ? ", "res9": "70"}],
        "ten": [{"q10": "what is 15+15 ? ", "res10": "30"}]
    }

# putting the questions in json data file

with open('qr.json', "w") as write_file:
    json.dump(game, write_file)

with open("qr.json", "r") as data_file:
```

```

    game = json.load(data_file)
n = 0
d = {}
for i in range(9):
    q11 = input(print(game["one"][0]["q1"]))
    n = n + 1
    q22 = input(game["two"][0]["q2"])
    if q22 == game["two"][0]["res2"]:
        n = n + 1
    q33 = input(game["three"][0]["q3"])
    if q33 == game["three"][0]["res3"]:
        n = n + 1
    q44 = input(game["four"][0]["q4"])
    if q44 == game["four"][0]["res4"]:
        n = n + 1
    q55 = input(game["five"][0]["q5"])
    if q55 == game["five"][0]["res5"]:
        n = n + 1
    q66 = input(game["six"][0]["q6"])
    if q66 == game["six"][0]["res6"]:
        n = n + 1
    q77 = input(game["seven"][0]["q7"])
    if q77 == game["seven"][0]["res7"]:
        n = n + 1
    q88 = input(game["eight"][0]["q8"])
    if q88 == game["eight"][0]["res8"]:
        n = n + 1
    q99 = input(game["nine"][0]["q9"])
    if q99 == game["nine"][0]["res9"]:
        n = n + 1
    q1010 = input(game["ten"][0]["q10"])
    if q1010 == game["ten"][0]["res10"]:
        n = n + 1
    break
print("Your Degree Is : ", n, "/10")

game_responses={ "response": [q11, q22, q33, q44, q55, q66, q77, q88, q99,
q1010, n] }
with open('RES.json', "w") as write_file:
    json.dump(game_responses, write_file)

print(game_responses)

```



```
QUESTION3.py - C:\Users\DELL 3567\Desktop\python\QUESTION3.py (3.10.4)
File Edit Format Run Options Window Help

import json

game = {"one": [{"q1": "what is your name name ? ", "res1": "name"}],
        "two": [{"q2": "what is 5*8 ? ", "res2": "40"}],
        "three": [{"q3": "what is 32/2 ? ", "res3": "16"}],
        "four": [{"q4": "what is 125+25 ? ", "res4": "150"}],
        "five": [{"q5": "what is 14+6 ? ", "res5": "20"}],
        "six": [{"q6": "what is 4+6 ? ", "res6": "10"}],
        "seven": [{"q7": "what is 25+15 ? ", "res7": "40"}],
        "eight": [{"q8": "what is 5*(8-2) ? ", "res8": "30"}],
        "nine": [{"q9": "what is 30+40 ? ", "res9": "70"}],
        "ten": [{"q10": "what is 15+15 ? ", "res10": "30"}]
        }

# putting the questions in json data file

with open('qr.json', 'w') as write_file:
    json.dump(game, write_file)

with open("qr.json", "r") as data_file:
    game = json.load(data_file)

n = 0
d = {}
for i in range(9):
    q11 = input(print(game["one"][0]["q1"]))
    n = n + 1
    q22 = input(game["two"][0]["q2"])
    if q22 == game["two"][0]["res2"]:
        n = n + 1
    q33 = input(game["three"][0]["q3"])
    if q33 == game["three"][0]["res3"]:
        n = n + 1
    q44 = input(game["four"][0]["q4"])
    if q44 == game["four"][0]["res4"]:
        n = n + 1
    q55 = input(game["five"][0]["q5"])
    if q55 == game["five"][0]["res5"]:
        n = n + 1
    q66 = input(game["six"][0]["q6"])
    if q66 == game["six"][0]["res6"]:
        n = n + 1

IDLE Shell 3.10.4
File Edit Shell Debug Options Window Help

Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>>
===== RESTART: C:\Users\DELL 3567\Desktop\python\QUESTION3.py =====
what is your name name ?
Nonediana
what is 5*8 ? 40
what is 32/2 ? 16
what is 125+25 ? 200
what is 14+6 ? 20
what is 4+6 ? 10
what is 25+15 ? 40
what is 5*(8-2) ? 30
what is 30+40 ? 70
what is 15+15 ? 40
Your Degree Is : 8 /10
{'response': ['diana', '40', '16', '200', '20', '10', '40', '30', '70', '40', 8]}
>>>
```

