

Syrian Arab Republic

Lattakia - Tishreen University

Department of Communication and  
electrical engineering

4<sup>th</sup>, Network: Homework No1



الجمهورية العربية السورية

اللاذقية - جامعة تشرين

كلية الهندسة الكهربائية والميكانيكية

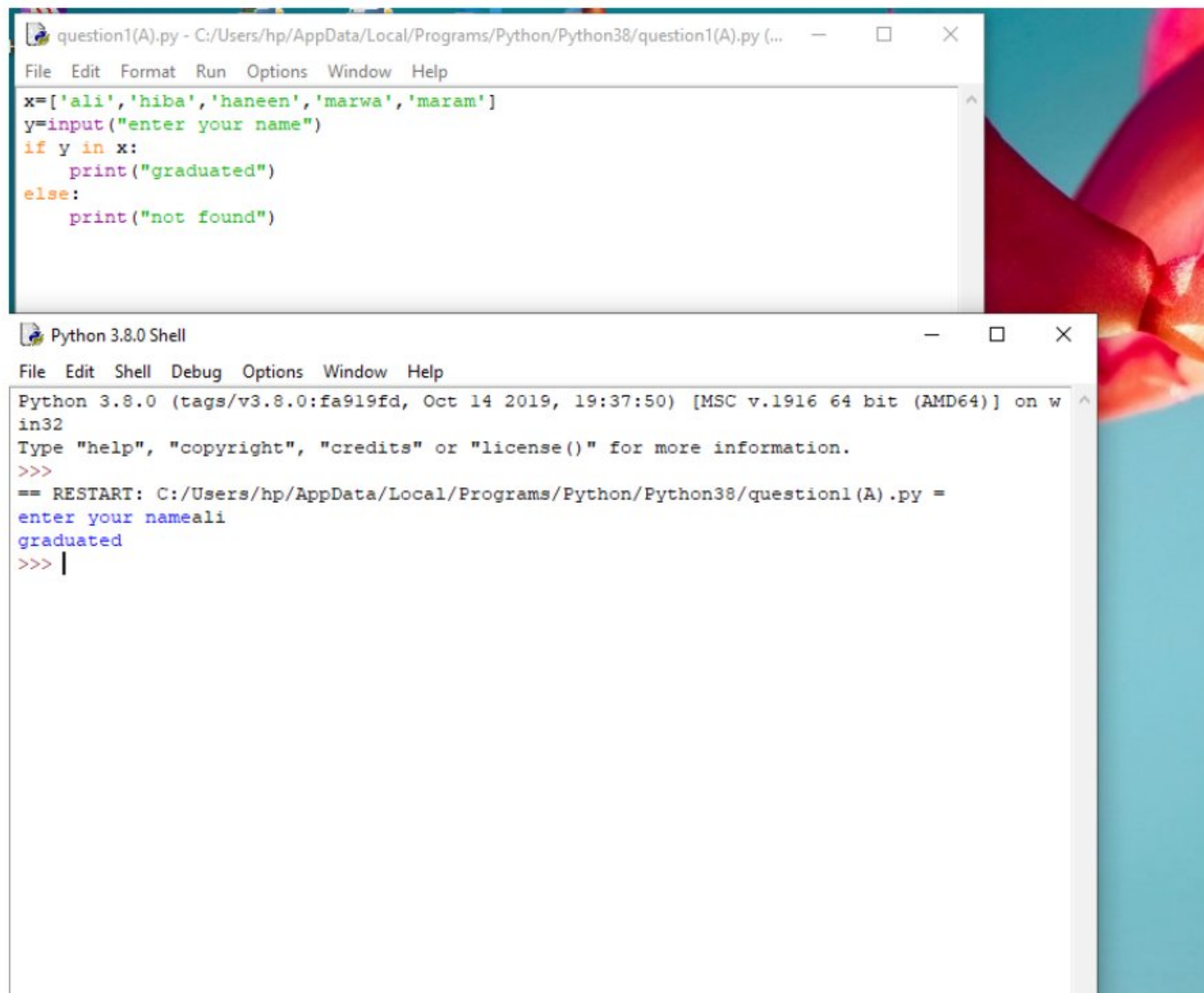
قسم هندسة الاتصالات والإلكترونيات

السنة الخامسة: وظيفة 1 شبكات

## First Network Programming Homework

### 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.



```
question1(A).py - C:/Users/hp/AppData/Local/Programs/Python/Python38/question1(A).py (...
File Edit Format Run Options Window Help
x=['ali', 'hiba', 'haneen', 'marwa', 'maram']
y=input("enter your name")
if y in x:
    print("graduated")
else:
    print("not found")

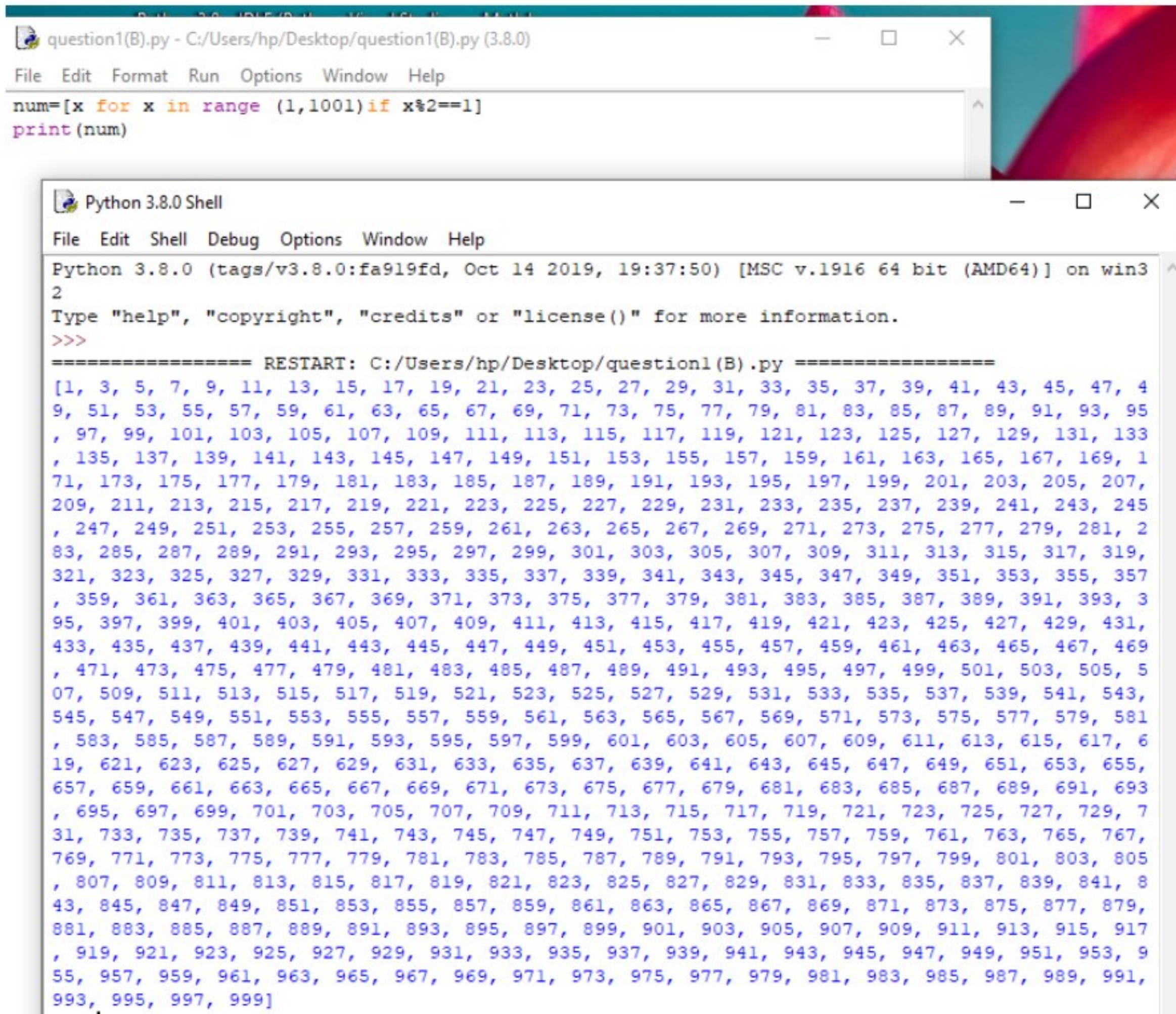
Python 3.8.0 Shell
File Edit Shell Debug Options Window Help
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:37:50) [MSC v.1916 64 bit (AMD64)] on w
in32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/hp/AppData/Local/Programs/Python/Python38/question1(A).py =
enter your nameali
graduated
>>> |
```

## التفسير:

سوف نقوم بإنشاء قائمة تحوي أسماء الطلاب المتخرجين  
ثم نطلب من المستخدم إدخال أسمة ونقارن إذا كان  
الطالب قد تخرج يطبع أنه متخرج      وغيرذلك يطبع غير  
موجود



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



```
question1(B).py - C:/Users/hp/Desktop/question1(B).py (3.8.0)
File Edit Format Run Options Window Help
num=[x for x in range (1,1001) if x%2==1]
print(num)
```

```
Python 3.8.0 Shell
File Edit Shell Debug Options Window Help
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:37:50) [MSC v.1916 64 bit (AMD64)] on win3
2
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/hp/Desktop/question1(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]
```

List Comprehension

التفسير:

في هذا البرنامج سوف نقوم بتوليد الأعداد الفردية من الصفر وحتى الألف نستخدم



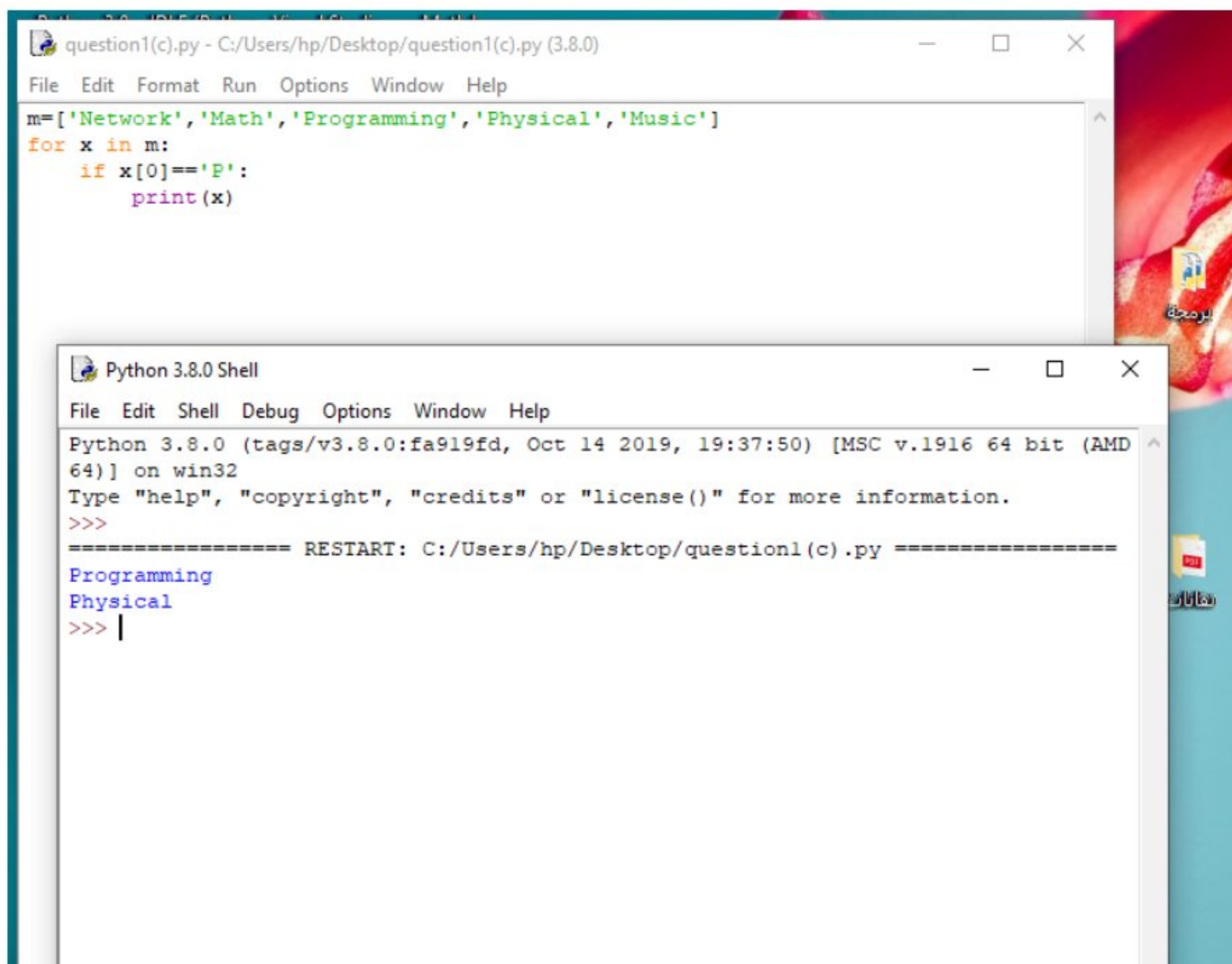
: List comprehension

# نستخدم حلقة لتوليد هذه الأعداد ضمن مجال محدد

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



The screenshot shows a Python IDE with two windows. The top window, titled 'question1(c).py - C:/Users/hp/Desktop/question1(c).py (3.8.0)', contains the following code:

```
m=['Network', 'Math', 'Programming', 'Physical', 'Music']
for x in m:
    if x[0]=='P':
        print(x)
```

The bottom window, titled 'Python 3.8.0 Shell', shows the output of the script after execution:

```
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:37:50) [MSC v.1916 64 bit (AMD 64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/hp/Desktop/question1(c).py =====
Programming
Physical
>>> |
```

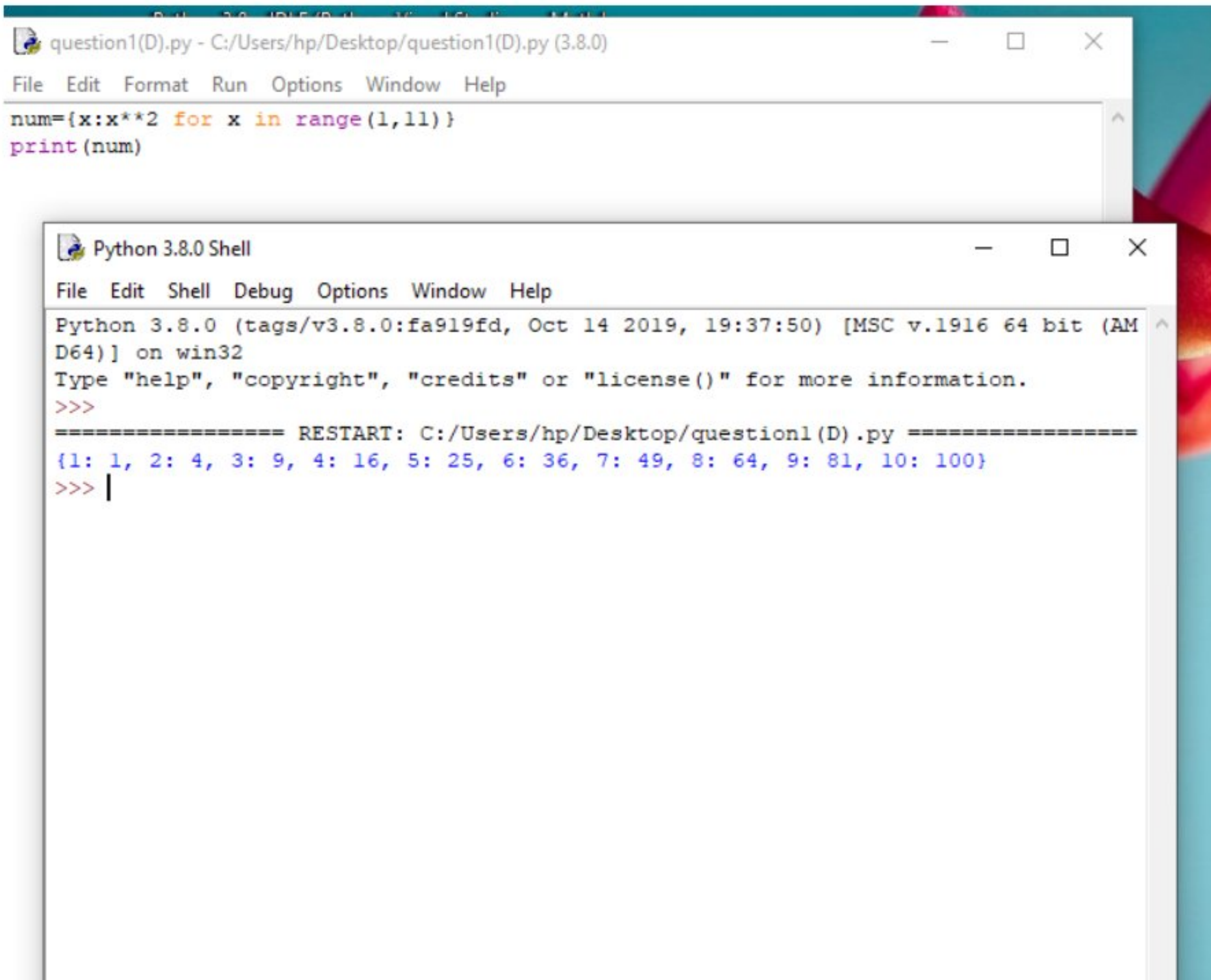
:التفسير

في هذا البرنامج نقوم بإنشاء قائمة تحوي  
مجموعة من المواد ثم نمشي ع كل  
العناصر ونطبع الأسماء الذي تبدأ

p

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





The image shows a screenshot of a Python IDE window titled 'question1(D).py - C:/Users/hp/Desktop/question1(D).py (3.8.0)'. The code in the editor is:

```
num={x:x**2 for x in range(1,11)}  
print(num)
```

Below the editor is a 'Python 3.8.0 Shell' window. It displays the following output:

```
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:37:50) [MSC v.1916 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
===== RESTART: C:/Users/hp/Desktop/question1(D).py =====  
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100}  
>>> |
```

التفسير:

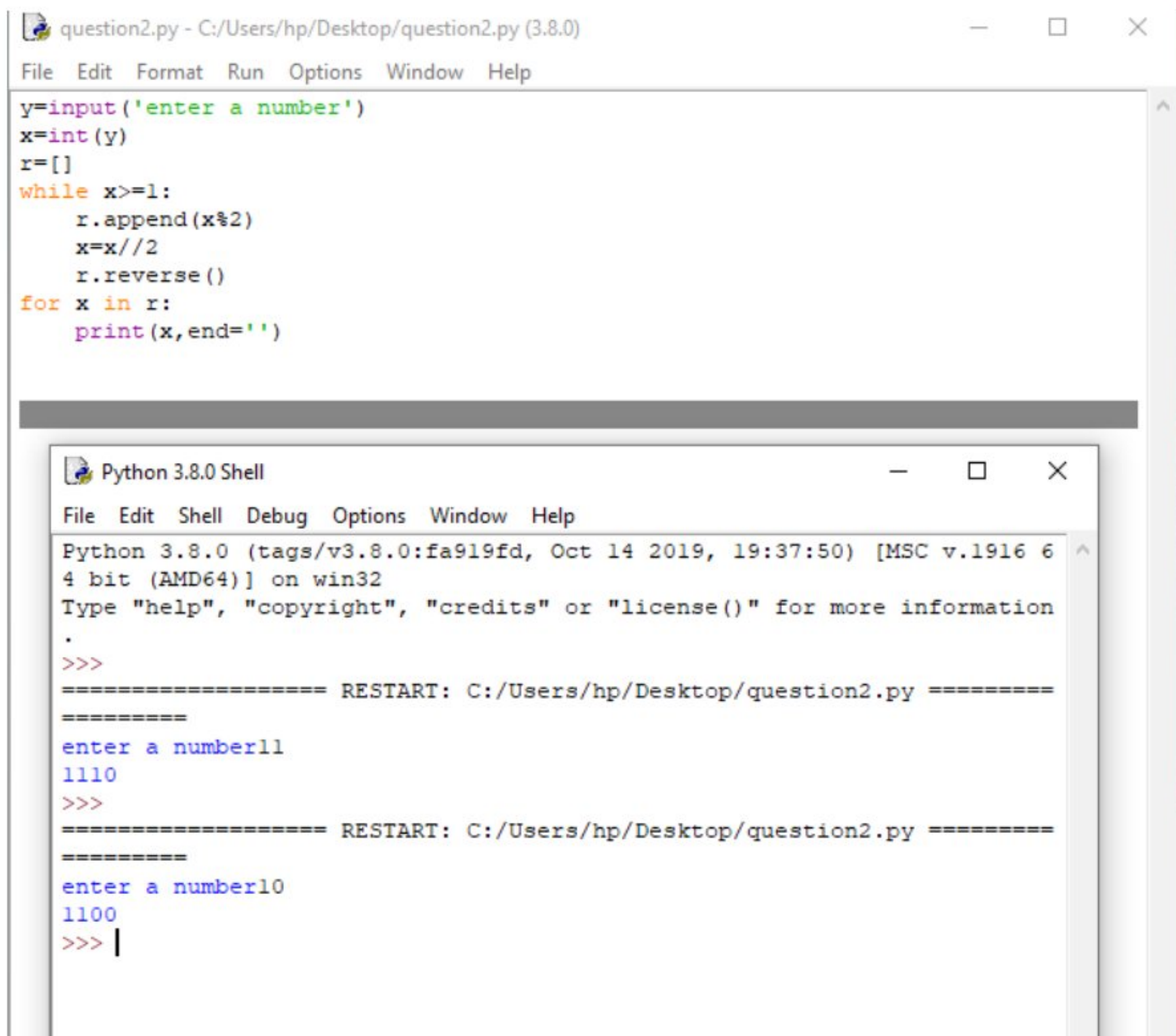
نفوم بتوليد الأعداد الفردية ومربعها من العدد واحد وحتى الإحدى عشر

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.



```
question2.py - C:/Users/hp/Desktop/question2.py (3.8.0)
File Edit Format Run Options Window Help
y=input('enter a number')
x=int(y)
r=[]
while x>=1:
    r.append(x%2)
    x=x//2
    r.reverse()
for x in r:
    print(x,end='')

Python 3.8.0 Shell
File Edit Shell Debug Options Window Help
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:37:50) [MSC v.1916 6
4 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information
.>>>
===== RESTART: C:/Users/hp/Desktop/question2.py =====
enter a number11
1110
>>>
===== RESTART: C:/Users/hp/Desktop/question2.py =====
enter a number10
1100
>>> |
```

## التفسير:

نقوم بطلب من المستخدم بأدخال رقم ثم نحسب ونقوم بتعريف قائمة فارغة ثم نضيف لها باقي قسمة العدد



المدخل على اثنان شرط أن يكون ناتج باقي القسمة أكبر من الصفر ونحسب ناتج القسمة الصحيحة ثم نأخذ معكوس ونطبع الناتج باستخدام حلقة

### 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)).It asks the questions and finally computes and prints user results and store user name and result in separate file.



```
1 import json
2 num=input("enter your name")
3 s=0
4 ob={}
5 with open("question.json","r") as f:
6     ob=json.load(f)
7
8 for i in ob.keys():
9     print(i)
10    oa=input("enter your answer")
11
12    if ob[i]==oa:
13        print("excellent")
14        s=s+10
15    else:
16        print("false")
17 with open("result.json","w") as f:
18     f.write(num,s)
19
```



```
{} question.json X
C: > Users > hp > Desktop > {} question.json > ...
1 [{"1+1":"2","2+2":"4","2+5":"7"}]
```



:التفسير

في هذا البرنامج ننشأ ملف بامتداد json يحتوي قائمة تتضمن مجموعة من الأسئلة والأجوبة

ننشأ برنامج بامتداد بايثون نضمن ضمنية json عن طريق تعليمة import

نطلب من المستخدم إدخال أسمة ثم ننشأ dict فارغة وعدد نعطية قيمة 0

نفتح ملف JSON ونقوم بتحميل الملف عن طريق تعليمة Load ثم نمر على مفاتيح ال dict أي على الأسئلة ونطبعها

نطلب من المستخدم إدخال الجواب ونقارن مع الأجوبة الموجودة إذا كانت نفسها نطبع excellent ونزيد العدد S أي العلامة بمقدار 10 أو نطبع false

ثم نطبع النتيجة في ملف النتيجة الذي إمتداده json