السؤال الاول:

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.
B- Generate and print a list of odd numbers from 1 to 1000.
    Tips: "List Comprehension"
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
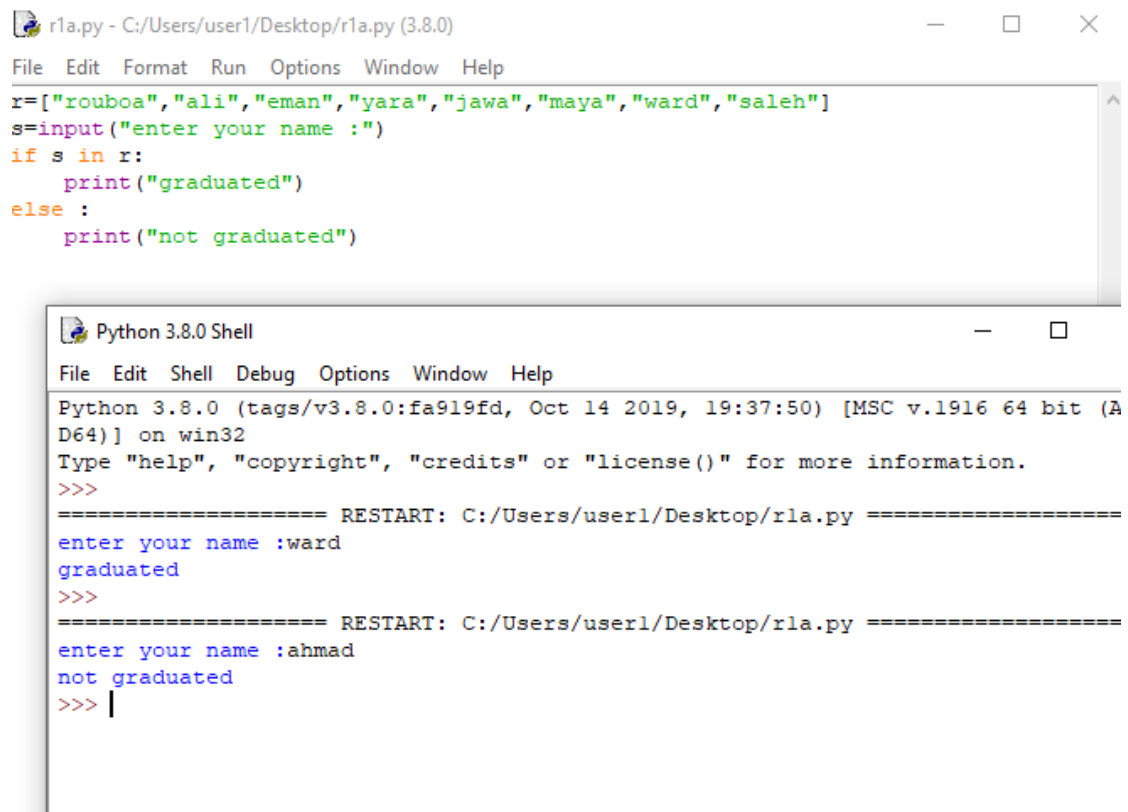   the items that starts with 'P' letter, then print it on screen.
    Tips: using loop, list 'len ()' method
D: Using Dictionary comprehension, Generate this dictionary d={1:1,2:4,3:9,4:16,5:25,6:36,7:42,8:64,9:81,10:100}

الطلب A:

قائمه بأسماء الطلاب المتخرجين

الكود هنا يأخذ اسم الطالب ويعطي في حال هذا الطالب متخرج ام لا

```
r1a.py - C:/Users/user1/Desktop/r1a.py (3.8.0)                    —    □    ✕

File  Edit  Format  Run  Options  Window  Help
r=["rouboa","ali","eman","yara","jawa","maya","ward","saleh"]
s=input("enter your name :")
if s in r:
    print("graduated")
else :
    print("not graduated")
```

```
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 (A
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=================== RESTART: C:/Users/user1/Desktop/r1a.py ===================
enter your name :ward
graduated
>>>
=================== RESTART: C:/Users/user1/Desktop/r1a.py ===================
enter your name :ahmad
not graduated
>>>
```

الطلب B:

نطبع ضمن قائمه الاعداد الفرديه من 1 ال 1000

```
e=[i for i in range (1,1001)if i%2==1]
print(e)
```

```
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:37:50) [MSC v.1916 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/user1/AppData/Local/Programs/Python/Python38/1b.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]
>>>
```

الطلب C:

قائمه باسماء المواد نريد طباعة فقط المواد التي تبدأ ب P

```
ooo.py - C:/Users/user1/Desktop/ooo.py (3.8.0)

File  Edit  Format  Run  Options  Window  Help

LIST=["Network","Math","Programming","Pysics",'Music']
for n in LIST :
    if n[0]=="P":
        print(n)
```

```
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
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more informatio
>>>
=================== RESTART: C:/Users/user1/Desktop/ooo.py ==========
Programming
Pysics
>>>
```

الطلب D:
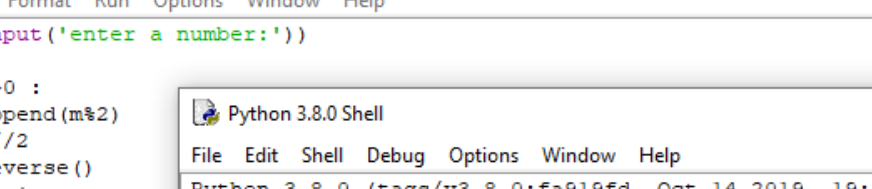
نبني dictionary ونطبع منها مربعات الاعداد من 1 ل 10

```
File  Edit  Format  Run  Options  Window  Help

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

```
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
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more informatio
>>>
==================== RESTART: C:/Users/user1/Desktop/tt.py ==========
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100}
>>>
```

السؤال الثاني :

في هذا البرنامج ندخل عدد عشري ونحوله الى ثنائي ونريد طباعة العدد الثنائي بحيث نقسم العدد العشري على 2 ونخزن الناتج في العدد نفسه ثم نقسم الناتج على 2 ايضا ونحتفظ بباقي القسمه ونكرر هذه العمليه حتى ناتج يساوي 0 ثم نطبع باقي القسمه في كل عمليه ويكون هو العدد الثنائي المطلوب

السؤال الثالث:

نضع مجموعه من الاسئله ضمن ملف json وايضا نضع الحل الصحيح ثم نكتب برنامج بايثون نجعل الطالب يدخل اسمه ونختبره بالاسئله السابقه ونقارن اجوته مع الاجابه الصحيحه بحيث يحصل على علامه عند كل اجابه صحيحه ولا يأخذ أي علامة عند الاجابه الغلط ونطبع اسم الطالب ونتيجته وعلامته المستحقه

```python
import json
1="what is the sky color?"
2="what is the snow color?"
3="what is the sky color?"
4="what is the snow color?"
5="what is the sky color?"
true={1:"blue",2:"white",3:"blue",4:"white",5:"blue"}
r=json.dumbs(true)
with open ("r.json","w") as t:
    t.write(r)
```

```python
import json
name=input("enter your name:")
mark=0
p=[]
with open ("r.json","r") as t:
    r=json.loads(t.read())
for i in r :
    ans=input("enter the answer:")
    p.append(ans)
    if ans==r[i]:
        print("yes")
        maark=mark+1
    else :
        print("no")
w={name,p}
print(w)
print("the mark is : ",mark)
```