

Syrian Arab Republic

Lattakia - Tishreen University

Department of Communication and electrical  
engineering

5<sup>th</sup> , Network Programming : Homework No1



الجمهورية العربية السورية

اللاذقية- جامعة تشرين

كلية الهندسة الميكانيكية والكهربائية

قسم هندسة الاتصالات والالكترونيات

السنة الخامسة

الوظيفة الأولى برمجة شبكات

الاسم: يارا رفيق صقور

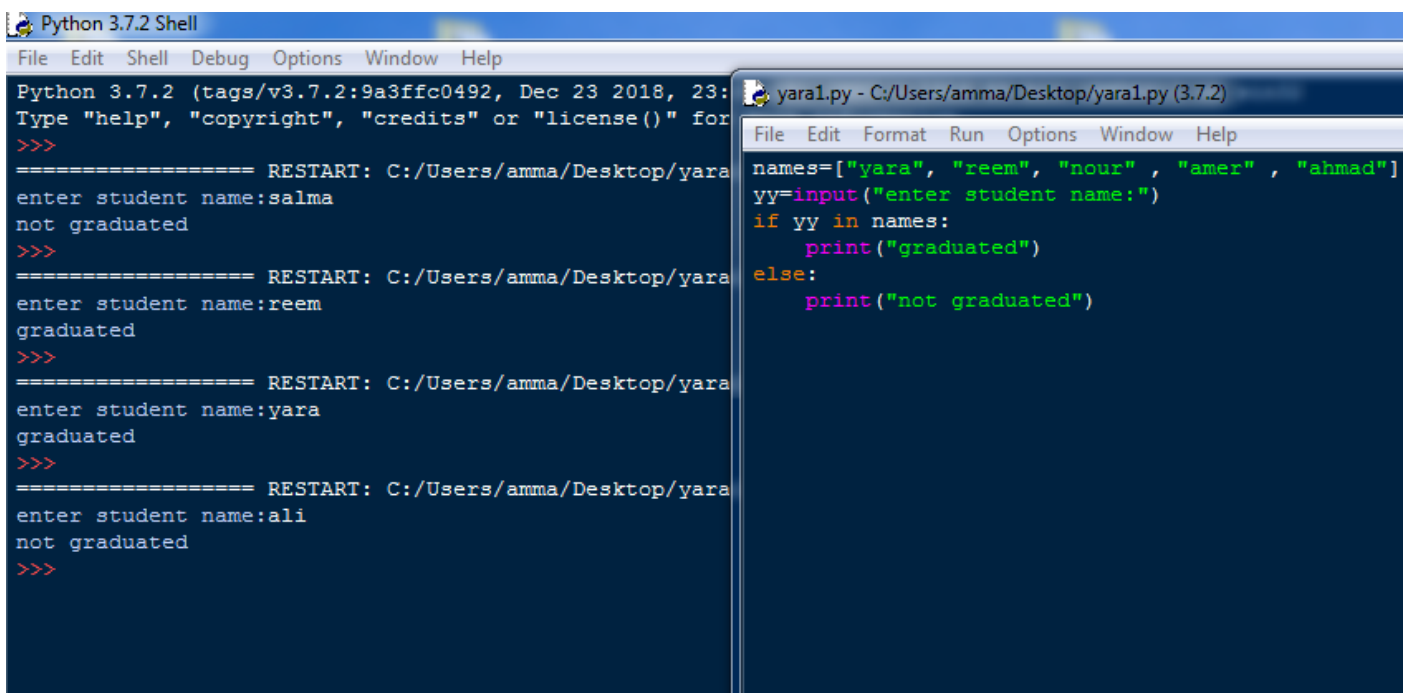
الرقم الجامعي: ٢٣١٨

# 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 .



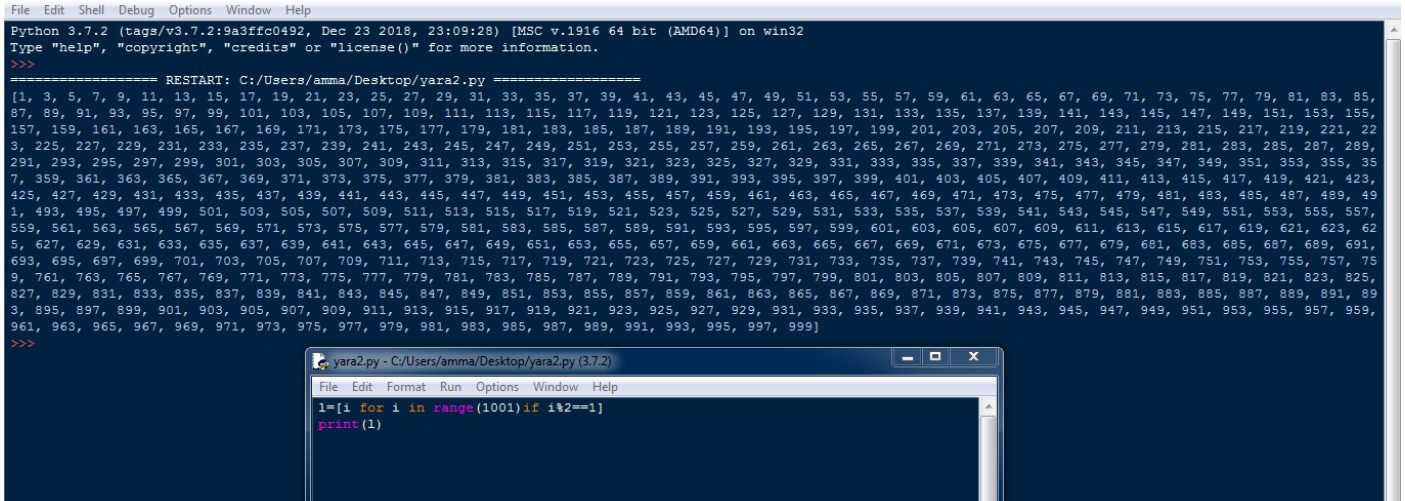
```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:
Type "help", "copyright", "credits" or "license()" for
>>>
===== RESTART: C:/Users/amma/Desktop/yara
enter student name:salma
not graduated
>>>
===== RESTART: C:/Users/amma/Desktop/yara
enter student name:reem
graduated
>>>
===== RESTART: C:/Users/amma/Desktop/yara
enter student name:yara
graduated
>>>
===== RESTART: C:/Users/amma/Desktop/yara
enter student name:ali
not graduated
>>>

yara1.py - C:/Users/amma/Desktop/yara1.py (3.7.2)
File Edit Format Run Options Window Help
names=["yara", "reem", "nour", "amer", "ahmad"]
yy=input("enter student name:")
if yy in names:
    print("graduated")
else:
    print("not graduated")
```

تعريف مصفوفة نضع فيها أسماء الطلاب المتخرجين في حال كان الاسم المدخل ضمن مصفوفة الطلاب  
يطبع متخرج والا يطبع غير متخرج عن طريق استخدام if/else الشرطية.

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

Tips: "List Comprehension"



```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/amma/Desktop/yara2.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]
>>>
```

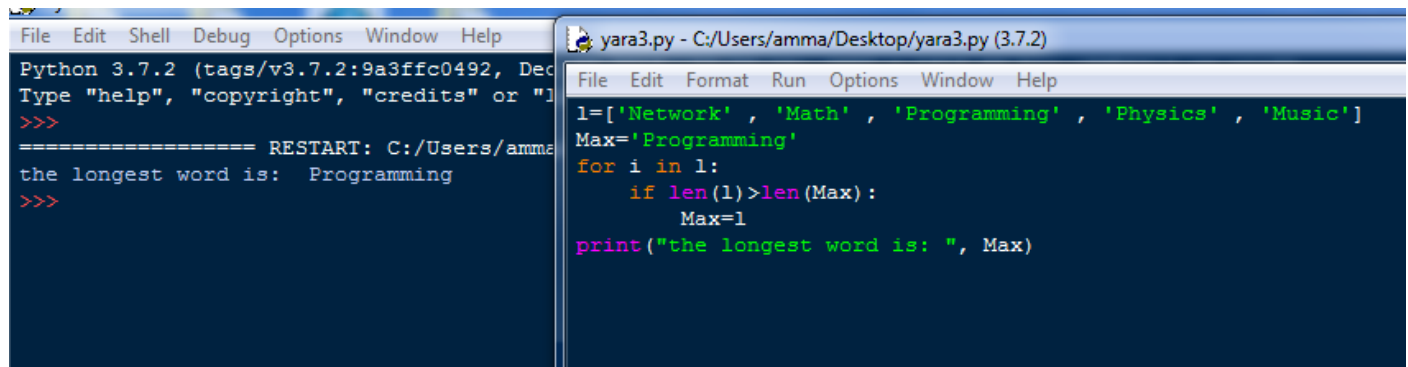
```
yara2.py - C:/Users/amma/Desktop/yara2.py (3.7.2)
File Edit Format Run Options Window Help
l=[i for i in range(1001) if i%2==1]
print(l)
```

البرنامج يقوم بطباعة الأعداد الفردية باستخدام حلقة for بخطوة افتراضية بمقدار واحد والمجال من 0 إلى 1000 وذلك في حال كان باقي قسمة العدد على 2 يساوي 1 .

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

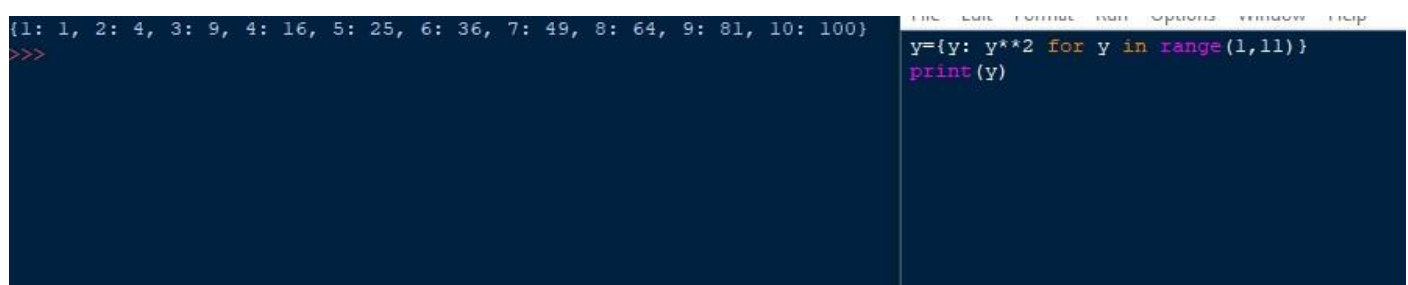


```
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 11 2019)
Type "help", "copyright", "credits" or "license()"
>>>
===== RESTART: C:/Users/amma/Desktop/yara3.py (3.7.2)
the longest word is: Programming
>>>
```

```
File Edit Format Run Options Window Help
yara3.py - C:/Users/amma/Desktop/yara3.py (3.7.2)
l=['Network' , 'Math' , 'Programming' , 'Physics' , 'Music']
Max='Programming'
for i in l:
    if len(l)>len(Max):
        Max=l
print("the longest word is: ", Max)
```

من القائمة المعطاة نفرض أن كلمة Programming هي الكلمة الأطول حيث يتم تخزينها في المتحول Max ونستخدم الحلقة for للمرور على عناصر القائمة ومقارنتها مع الكلمة المفروضة ، ويتم طباعة الكلمة الأطول.

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



```
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100}
>>>
```

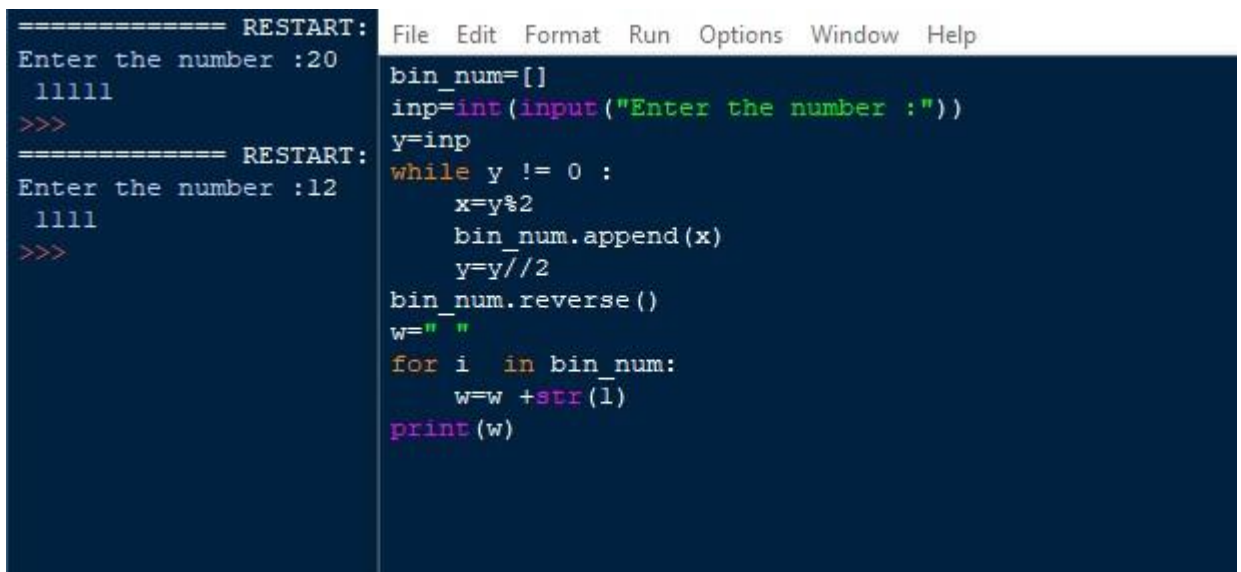
```
File Edit Format Run Options Window Help
y={y: y**2 for y in range(1,11)}
print(y)
```

## 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



```
===== RESTART:
Enter the number :20
11111
>>>
===== RESTART:
Enter the number :12
1111
>>>
```

```
File Edit Format Run Options Window Help
bin_num=[]
inp=int(input("Enter the number :"))
y=inp
while y != 0 :
    x=y%2
    bin_num.append(x)
    y=y//2
bin_num.reverse()
w=""
for i in bin_num:
    w=w+str(i)
print(w)
```

نعرف مصفوفة عبارة عن الرقم الثنائي الذي سوف يظهر لدينا عند تنفيذ البرنامج ونفرض أنه طالما العدد العشري المدخل لا يساوي الصفر ونأخذ باقي قسمته على 2 ونضيفه إلى مصفوفة العدد الثنائي وهكذا إلى أن يكون باقي قسمة العدد العشري المدخل يساوي الصفر ونسندده إلى مصفوفة العدد الثنائي وتخرجه المصفوفة على شكل string.

### 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.

```
>>> q1
'x+y=? :\na.9\nb.10'
>>> q10
'z*z*x? :\na.144\nb.10'
>>> q14
'x*z/y? :\na.1\nb.3.34'
>>>
```

```
File Edit Format Run Options
import json
q1="""x+y=? :
a.9
b.10"""
q2="""x-y=? :
a.1
b.-1"""
q3="""x+z=? :
a.9
b.10"""
q4="""x+y+z=? :
a.14
b.10"""
q5="""z+y-x=? :
a.9
b.7"""
q6="""x*y=? :
a.9
b.20"""
q7="""y//x=? :
a.1
b.10"""
q8="""x*x*y? :
a.80
b.10"""
q9="""y*y*x? :
a.1
b.100"""
q10="""z*z*x? :
a.144
b.10"""
q11="""z-y-x? :
a.-3
b.10"""
q12="""z-x? :
a.2
b.10"""
q13="""z*y/x ? :
a.7.5
b.10"""
q14="""x*z/y? :
a.1
b.3.34"""
```

```

File Edit Format Run Options Window Help
>>> q1
'x+y=? :\na.9\nb.10'
>>> q10
'z*z*x? :\na.144\nb.10'
>>> q14
'x*z/y? :\na.1\nb.3.34'
>>>
b.10""
q9=""y*y*x? :
a.1
b.100""
q10=""z*z*x? :
a.144
b.10""
q11=""z-y-x? :
a.-3
b.10""
q12=""z-x? :
a.2
b.10""
q13=""z*y/x ? :
a.7.5
b.10""
q14=""x*z/y? :
a.1
b.3.34""
q15=""y*y-x*x? :
a.9
b.10""
q16=""x*x+z*y ? :
a.46
b.10""
q17=""y*z-2x? :
a.22
b.10""
q18=""2x+2z-y? :
a.1
b.10""
q19=""3x-z*y? :
a.30
b.10""
q20=""z-4y+6x? :
a.1
b.10""
dic = {q1:"a",q2:"b",q3:"b",q4:"a",q5:"b",q6:"b",q7:"a",q8:"a",q9:"b",q10:"a",
q11:"a",q12:"a",q13:"a",q14:"b",q15:"a",q16:"a",q17:"a",q18:"b",q19:"a",q20:"b"}
q=json.dumps(dic)
with open("q.json","w") as f:
    f.write(q)

```