

## الوظيفة الأولى – برمجة شبكات – نور عساف – 2083

### 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.

```
graduated = ["Nour", "Ahmad", "Safi", "Maram", "Toni", "Marah", "Tamer", "Maher", "Tasneem"]
new_student_name = input('Student name: ')
if new_student_name in graduated:
    print(new_student_name+' Is graduated.')
else:
    print(new_student_name+' Is not graduated.')
```

في هذا البرنامج نقوم بتعريف لائحة من الطلاب المتخرجين ثم نطالب المستخدم بإدخال اسم طالب للاستعلام عنه. ثم نقوم بالبحث عن اسم الطالب المطلوب ضمن لائحة الطلاب المتخرجين باستخدام الكلمة in. ثم نطبع العبارة المناسبة.

```
Student name: Nour
Nour Is graduated.
```

```
Student name: Ali
Ali Is not graduated.
```

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

```
numbers = [ x for x in range(1000) if x % 2 > 0 ]
print(numbers)
```

في هذا البرنامج نقوم ببناء اللائحة المطلوبة عن طريق اختبار الأعداد التي تقع ضمن المجال 0 – 999 في حال كانت فردية نقوم بإضافتها لللائحة.

```
[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.

```
L = ['Network', 'Math', 'Programming', 'Physics', 'Music']
for i in range(len(L)):
    if L[i].startswith('P'):
        print(L[i])
```

في هذا البرنامج نقوم باستخدام حلقة for بالمرور على عناصر اللائحة وفي كل مرة نختبر في حال كان الحرف الأول هو P باستخدام التابع .startswith وفي حال التحقق نقوم بطباعة الكلمة.

## Programming Physics

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}

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

في هذا البرنامج نقوم ببناء الـ dictionary المطلوبة عن طريق المرور على الأعداد التي تقع ضمن المجال 0 – 10. وفي كل مرة نضيف العدد كـ Key ومربعه كـ Value.

```
{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.

```
num = int(input('Number: '))  
binary[] =  
while num > 0:  
    binary.append(num%2)  
    num=num//2  
for i in range(len(binary)):  
    print(binary[-i-1], end='')
```

في هذا البرنامج نقوم بالقسمة المتكررة للعدد المدخل على 2 (باستخدام حلقة while) وفي كل مرة نخزن باقي القسمة في اللائحة binary والاحتفاظ بباقي القسمة بدلاً من العدد نفسه. وتكرار العملية للوصول للقيمة صفر. عند الانتهاء نقوم بطباعة اللائحة binary من النهاية حتى البداية.

```
Number: 200  
11001000
```

```
Number: 20  
10100
```

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.

البرنامج المطلوب:

```
import json  
  
f=open('questions.json')  
questions = json.load(f)  
f.close()  
  
student = input('Enter Your name:')  
mark = 0  
  
for element in questions:  
    q = element["q"]  
    a = element["a"]  
    ans = input(q+'=')
```

```

if ans == a:
    mark = mark + 1

print('your mark is: {}'.format(mark))

f=open('students.txt','a')
f.write(student+" = "+str(mark)+"\n")
f.close()

```

ملف الأسئلة: (ملف json)

```

[
  {"q": "10^1", "a": "10"},
  {"q": "10^2", "a": "100"},
  {"q": "10^3", "a": "1000"},
  {"q": "10^4", "a": "10000"},
  {"q": "10^5", "a": "100000"},
  {"q": "10^6", "a": "1000000"},
  {"q": "(10*7)/5", "a": "14"},
  {"q": "5*3", "a": "15"},
  {"q": "1+1", "a": "2"},
  {"q": "2+6/3", "a": "4"},
  {"q": "(2+4)/3", "a": "2"},
  {"q": "cos(90)", "a": "0"},
  {"q": "sin(0)", "a": "0"},
  {"q": "sin(90)", "a": "1"},
  {"q": "PI", "a": "3.14"},
  {"q": "18/6", "a": "3"},
  {"q": "x+5 is 0 so x", "a": "-5"},
  {"q": "3*7", "a": "21"},
  {"q": "7*7", "a": "49"},
  {"q": "7*10", "a": "70"}
]

```

في هذا البرنامج نقوم بتحميل محتوى الملف questions.json وتحويل محتواه إلى كائن باستخدام التابع json.load. ثم نقوم بإدخال اسم الطالب ونفرض علامته 0 في البداية.

ثم باستخدام حلقة تمر على جميع الأسئلة وفي كل مرة نعرض السؤال للطالب ونطلبه بإدخال الإجابة، ثم نقارن إجابته مع الإجابة الصحيحة وفي حال التطابق يكسب الطالب علامة.

بعد الانتهاء من الاختبار نعرض للطالب علامته ونخزن اسمه وعلامته في ملف نصي students.txt.

مثال على تنفيذ الاختبار:

```
Enter Your name:Nour
10^1=10
10^2=100
10^3=1000
10^4=10000
10^5=100000
10^6=1000000
(10*7)/5=0
5*3=0
1+1=0
2+6/3=0
(2+4)/3=0
cos(90)=0
sin(0)=0
sin(90)=0
PI=0
18/6=0
x+5 is 0 so x=0
3*7=0
7*7=0
7*10=0
your mark is: 8
```

النتيجة يتم تخزينها في ملف نصي.  
مثال عن الملف بعد إجراء الاختبار مرة ثانية:

```
students.txt
1  Nour = 8
2  Sami = 4
3  |
```