

الجمهورية العربية السورية  
جامعة تشرين  
كلية الهندسة الميكانيكية والكهربائية  
قسم هندسة الاتصالات والالكترونيات  
الوظيفة 1 برمجة شبكات



الاسم: حنين عمار زينة      الرقم الجامعي :  
2432

**\*\*The first 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.

**Solution: code**

```
homework1.py x imoji.py
2
3 graduate=['haneen','ali','ward','leen','noor']
4 name=input('Enter your name please : ')
5 for i in range(len(graduate)):
6     if name in graduate:
7         print('🎓 GRADUATED')
8         break
9     else:
10        print('😞 FAILED')
11        break
12
```

## :Results

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\Windows 10\Desktop\python1> python homework1.py
Enter your name please : haneen
🎓 GRADUATED
PS C:\Users\Windows 10\Desktop\python1> python homework1.py
Enter your name please : ahmad
😞 FAILED
PS C:\Users\Windows 10\Desktop\python1> █
```

**الشرح:** قمت بتعريف قائمة بأسماء الطلاب المتخرجين وطلبت من المستخدم أن يدخل اسمه ثم قمت بتعريف حلقة تمر على فهرس عناصر القائمة ثم قمت بوضع شرط ليختبر إذا كان الاسم المدخل ضمن قائمة

المتخرجين إذا كان ضمن القائمة يظهر له أنه  
متخرج وإذا لم يكن ضمن القائمة يظهر له أنه  
قد فشل !!!!

-----

----

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

**Tips:** "List Comprehension"

**Solution: code**

```
oodnumber=[num for num in range(1,1000,2)]  
print(oodnumber)
```

**:Results**

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\Windows 10\Desktop\python1> python homework1.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]
PS C:\Users\Windows 10\Desktop\python1>
```

## الشرح:

إنشاء قائمة بالأعداد الفردية من 1 إلى 1000  
استخدمت التابع range() يبدأ ب 1 وينتهي ب  
1000 بخطوة مقدارها 2 بالتالي سوف يمر  
فقط على الأعداد الفردية

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

## Solution: code

```
homework1.py x imoji.py
1
2 L=['Network','Math','Programming','Physics','Music']
3
4 for i in range(len(L)):
5     M=L[i]
6     if M[0]=='P':
7         print(M)
8
9     else:
10        pass
11
```

## :Results

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\Windows 10\Desktop\python1> python homework1.py
Programming
Physics
PS C:\Users\Windows 10\Desktop\python1> █
```

**الشرح:** المطلوب هو طباعة أسماء المواد التي تبدأ بحرف P : قمت بإنشاء حلقة تمشي

على فهارس ال list المعطاة ثم أنشأت متغير اسمه M وأسندت له القيمة بفهرس معين حسب دوران الحلقة .. نلاحظ أن عناصر ال List هي عبارة عن string و كما نعلم أن لكل محرف من محارف ال السلسلة النصية index خاص به وبالتالي وضعت شرط أنه إذا كانت أول قيمة محرف بكل عنصر من عناصر ال list تساوي تماما حرف P بالتالي يطبع على الخرج القيمة المحققة للشرط بالفهرس المحدد

---

**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}

**Solution: code**

```
1
2
3
4
5 square={x:x**2 for x in range(11) if x!=0}
6 print(square)
7
```

## :Results

---

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL
Windows PowerShell Copyright (C) Microsoft Corporation. All rights reserved.  PS C:\Users\Windows 10\Desktop\python1> python homework1.py {1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100} PS C:\Users\Windows 10\Desktop\python1> █			

**الشرح:** إن ال Dictionary comprehension تشبه تمامًا List comprehension ولكن نستعوض عن أقواس ال list بأقواس dictionary : فمن أجل كل قيمة من المجال من 0 إلى 11 (كتبنا 11 لأنه لا يأخذ آخر قيمة أي يأخذ المجال من 0 إلى 10) بشرط ألا يأخذ الصفر وذلك لتحقيق المطلوب , ومن

---

أجل كل قيمة من المجال سيقوم بتربيعها  
فينتج قاموس ال key له هو العدد بذاته وال  
value هي مربع هذا العدد.

---

## 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.

**:Solution and results**



```
{} student.json • {} score.json • decimal2binary.py x
4
5 num=int(input('enter the decimal number :'))
6 bin=[]
7 while num>=1:
8     bin.append(num%2)
9     num=num//2
10 bin.reverse()
11 for num in bin:
12     print(num)
13
14
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\Windows 10\Desktop\python1> python decimal2binary.py
enter the decimal number :8
1
0
0
0
PS C:\Users\Windows 10\Desktop\python1> 
```

**الشرح:** نحن نعلم للتحويل من عشري الى ثنائي نقسم العدد العشري على 2 قسمة صحيحة // والباقي يكون هو ناتج التحويل

ويكون عبارة عن أصفار ووحدات لكن نبدأ  
بأخذ العدد من الأخير للأول تبعاً للأعلى أهمية  
فقمتم بتعريف قائمة فارغة وأضفت إليها  
نتائج التحويل ثم عكستها والتحويل قمت به  
داخل حلقة اخذت فقط الأعداد الموجبة  
وأضفت الأعداد إلى القائمة باستخدام التابع  
`append()`

---

### 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

## Solution: code

```
3.py • {} student.json • {} score.json •
1 import json
2 question={}
3 score=0
4 numofques=1
5 student={'The number of bones in the human body is 206':'T',
6 '5 * 5 equals 16':'F','The largest country in the world is Russia':'T'
7 , 'Geology is the science of the layers of the earth':'T',
8 'The most powerful type of stone is diamond':'T',
9 'The unit of measurement for sound intensity is the volt':'F','2 * 2 equals 4':'T','10 * 10 equals 11':'F',
10 '3*3=9':'T','8*8=16':'F','1*1=2':'F','The root of 4 is 2 ':'T','5**2=25':'T','6*6=36':'T','9*4=7':'F',
11 'Damascus, the capital of Syria':'T','The capital of Lebanon is Beirut':'T','1+4=5':'T','8+9=10':'F','3+3=6':'T'}
12 with open('student.json','w') as f:
13     json.dump(student,f)
14 with open('student.json') as v:
15     question=json.load(v)
16
17 name=input('please enter your name:')
18 print('welcom',name,'good luck in the test')
19
20 for q in question.keys():
21     print('Question-',numofques,q)
22     res=input('the answer is')
23     if res.upper()==question[q].upper():
24         score=score+1
25         print('well done')
26         numofques=numofques+1
27     else:
28         print('wrong answer')
29         numofques=numofques+1
30     result={name:score}
31     with open('score.json','a') as v:
32         result=json.dump(result,v)
33
```

# :Results

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

1

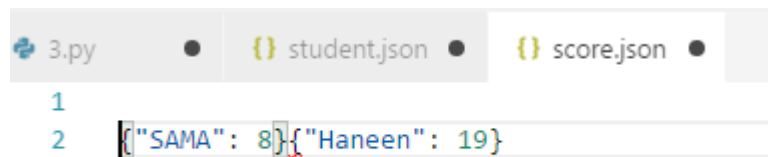
```
please enter your name:Haneen
welcom Haneen good luck in the test
Question- 1 The number of bones in the human body is 206
the answer isT
well done
Question- 2 5 * 5 equals 16
the answer isF
well done
Question- 3 The largest country in the world is Russia
the answer isT
well done
Question- 4 Geology is the science of the layers of the earth
the answer isT
well done
Question- 5 The most powerful type of stone is diamond
the answer isT
well done
Question- 6 The unit of measurement for sound intensity is the volt
the answer isF
well done
Question- 7 2 * 2 equals 4
the answer isT
well done
Question- 8 10 * 10 equals 11
the answer isF
well done
Question- 9 3*3=9
the answer isT
well done
Question- 10 8*8=16
the answer isF
well done
Question- 11 1*1=2
the answer isT
wrong answer
Question- 12 The root of 4 is 2
```

In 1 Co

**الشرح:** قمت بتعريف قاموس فارغ لإضافة الأسئلة عليه ثم قمت بتعريف متغير يعبر عن العلامة ومتغير يعبر عن عدد الأسئلة ثم قمت بتعريف قاموس يحوي ال 20 سؤال المطلوبة ك مفتاح و الأجوبة ك قيمة وقمت بتحويله ورفعة إلى ملف json اسمه student.json ثم مرة أخرى قمت بتحويل ملف ال json الى object نوعه قاموس لسهولة التعامل معه ثم طلبت من الطالب ان يدخل اسمه ثم قمت بتعريف حلقة تمشي على مفاتيح القاموس والتي تعبر عن الأسئلة ثم قمت بطباعة رقم السؤال مع السؤال للمستخدم وكانت الأسئلة عبارة عن صح و خطأ ثم قارنت جواب المستخدم مع قيمة المفتاح أي مع الجواب واستخدمت upper() لكي أضمن ان يكون الحرف المدخل كبير فإذا كان الجواب صحيح اطبع احسنت و أضيف له

علامة عند كل جواب صحيح وإلا يطبع له  
إجابة خاطئة ولا يزيد علامات ثم ينتقل  
للسؤال الذي يأتي بعده وهكذا حتى تنتهي  
الأسئلة وقمت بتخزين اسم الطالب وعلامته  
في قاموس ثم فتحت ملف json عبارة عن  
النتائج وحولت القاموس الذي يعبر عن اسم  
الطالب مع علامته الى ملف json اسمه  
score.json

وكانت النتيجة كالتالي :



```
3.py • student.json • score.json •  
1  
2 [{"SAMA": 8}, {"Haneen": 19}]
```