



جامعة تشرين

كلية الهندسة الميكانيكية والكهربائية

قسم الاتصالات والالكترونيات

الوظيفة الأولى

إيهاب محمد صوفان 1370

السؤال الأول الطلب الأول:

```
x = ["Ahmad", "Basel", "Cema", "Dall", "Eman"]
y = input("enter the name: ")
if y in x:
    print("student graduated")
else:
    print("student not graduated")
```

يتم فحص الاسم المدخل إذا كان موجود يطبع أن الطالب متخرج وإلا يطبع أنه غير متخرج.

```
enter the name: Ahmad
student graduated
```

السؤال الأول الطلب الثاني:

```
odd = [i for i in range(0, 1000) if (i % 2 != 0)]
print(odd)
```

تتكرر الحلقة وتبني *list* تحوي أعداد لا تقبل القسمة على 2 أي فردية.

```
[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]
```

السؤال الأول الطلب الثالث:

```
L=["Network","Math","Programming","Physics","Music"]
sp=[]
for i in L:
    if i[0] == 'P':
        sp.append(i)
y=len(sp)
print("there is " + str(y) + " start with p: " + str(sp))
```

يتم البحث في عناصر ال *list* عن عنصر أول حرف فيه هو P ويتم وضعه في *list* أخرى وعند الانتهاء يتم طباعة ال *list* التي تحوي العناصر المطلوبة.

```
there is 2 start with p: ['Programming', 'Physics']
```

السؤال الأول الطلب الرابع:

```
x={}
for i in range(1,11):
    x[i]= i * i
print(x)
```

عرفنا قاموس فارغ ومن ثم عملنا حلقة لملء القاموس بقيم المفاتيح ومقابل كل مفتاح مربع المفتاح

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

السؤال الثاني:

```
dec_num=int(input("enter a decimal number: "))
bin_num=[]
x=""
while dec_num!=0:
    y= dec_num % 2
    bin_num.append(y)
    dec_num= dec_num // 2
bin_num.reverse()
for i in bin_num:
    x=x+str(i)
print("the binary number is: "+x)
```

تم حساب ناتج قسمة العدد على 2 وفي كل مرة نخزن القيمة في قائمة
وبالنهاية نعكسها ونطبع النتيجة:

```
enter a decimal number: 8
the binary number is: 1000
```

السؤال الثالث:

```
import json
s={}
s_name=input("enter your name: ")
with open("A.json","r") as A:
    fA = json.load(A)
with open("Q.json","r") as Q:
    fQ = json.load(Q)
stanswer=[]
for key,value in fQ.items():
    print(key)
    print(value)
    answer=input("enter answer")
    stanswer.append(answer)
x=0
for i in range(0,20):
    if stanswer[i]==fA[i]:
        x+=1
s[s_name]=x
print("the reselt is: " + str(x))
with open("save.json","w") as save:
    fm = json.dumps(s)
    save.write(fm)
```

الأسئلة و الاجابات موجودة ضمن ملف json يتم استدعائه ضمن الكود حيث يجب الطالب على كل سؤال بالترتيب و يتم تخزين اجاباته ضمن مصفوفة يتم مقارنتها بالإجابات الصحيحة و إعطاء العلامة المناسبة باستخدام عداد يتزايد مع كل تطابق و يتم تخزين إجابة الطالب في ملف json مع اسمه

```
enter your name: x
```

```
1.Questions
```

```
['A', 'B', 'C']
```

```
enter answerA
```

```
2.Questions
```

```
['A', 'B', 'C']
```

```
enter answerC
```

```
3.Questions
```

```
['A', 'B', 'C']
```

```
enter answerB
```

```
4.Questions
```

```
['A', 'B', 'C']
```

```
enter answerA
```

```
5.Questions
```

```
['A', 'B', 'C']
```

```
enter answerC
```

```
6.Questions
```

```
['A', 'B', 'C']
```

```
enter answerB
```

```
7.Questions
```

```
['A', 'B', 'C']
```

```
enter answerA
```

```
8.Questions
```

```
['A', 'B', 'C']
```

```
enter answerC
```

```
9.Questions
```

```
['A', 'B', 'C']
```

```
enter answerB
```

```
10.Questions
```

```
['A', 'B', 'C']
```

```
enter answerA
```

```
19.Questions
```

```
['A', 'B', 'C']
```

```
enter answerB
```

```
20.Questions
```

```
['A', 'B', 'C']
```

```
enter answerA
```

```
the reselt is: 12
```

تم حفظ النتيجة كالتالي

```
{"x": 12}
```