**Python程式設計**

**期中考試題**

**銘傳大學電腦與通訊工程系**

|  |  |
| --- | --- |
| 班 級 | 電通四乙 |
| 姓 名 | 陳昱叡 |
| 學 號 | 04052474 |
| 成 績 |  |
| 授課教師 | 陳慶逸 |

1: 請以Python實現一個函式MaxProdNum(list)，該函式能計算所傳入之串列(list)中之元素兩兩相乘的乘積，並且回傳其中最小的值。

例如:

|  |  |
| --- | --- |
| **輸入** | **執行結果** |
| **MaxProdNum([27, 3, 4, 5, 7, 1])** | **7** |
| **MaxProdNum([-27, 3, -4, -40, 7, 12])** | -280 |
| **MaxProdNum([2, 3, 14, 5, 7, 41])** | 6 |

我的作答：

**請在下面欄位貼上程式碼：**

|  |
| --- |
| **def MaxProdNum(list):**  **first=list[0]\*list[1]**  **for i in range(0,len(list)):**  **if i+1<len(list):**  **nextone=list[i]\*list[i+1]**  **if first>nextone:**  **first=nextone**  **return first**  **print(MaxProdNum([27, 3, 4, 5, 7, 1]))**  **print(MaxProdNum([-20, -10, -40, 6, 27, 17]))**  **print(MaxProdNum([2, 13, 14, 55, 7, 41]))**  **print(MaxProdNum([60, 2, 400, 3, 8, 12]))**  **print(MaxProdNum([21, 1, 44, 55, 17, 280]))** |

**請依下面要求輸入參數，並將輸出結果複製起來貼上對應欄位:**

|  |  |
| --- | --- |
| **print(MaxProdNum([27, 3, 4, 5, 7, 1]))** | **7** |
| **print(MaxProdNum([-20, -10, -40, 6, 27, 17]))** | **-240** |
| **print(MaxProdNum([2, 13, 14, 55, 7, 41]))** | **26** |
| **print(MaxProdNum([60, 2, 400, 3, 8, 12]))** | **24** |
| **print(MaxProdNum([21, 1, 44, 55, 17, 280]))** | **21** |

**執行結果擷圖(說明: 請直接將下列程式碼複製貼上並且執行，再將包含指令與輸出結果的完整畫面擷取下來貼上)：**

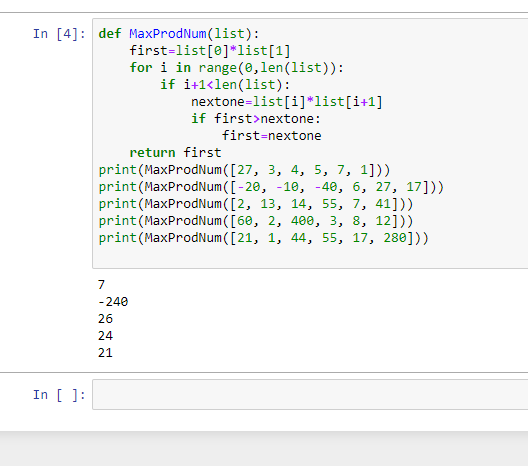
print(MaxProdNum([27, 3, 4, 5, 7, 1]))

print(MaxProdNum([-20, -10, -40, 6, 27, 17]))

print(MaxProdNum([2, 13, 14, 55, 7, 41]))

print(MaxProdNum([60, 2, 400, 3, 8, 12]))

print(MaxProdNum([21, 1, 44, 55, 17, 280]))



2: 請以Python實現一個函式max2(list)，該函式能回傳所傳入之串列(list)中，最大的兩個值。

例如:

|  |  |
| --- | --- |
| **輸入** | **執行結果** |
| **max2([3, 12, 5, 7, 9, 10])** | **(12, 10)** |
| **max2([233, 162, 25, 927, 119, 120])** | **(927, 233)** |

我的作答：

**請在下面欄位貼上程式碼：**

|  |
| --- |
| **def max2(x):**  **x=list(x)**  **x.sort()**  **m1=x[-1]**  **m2=x[-2]**  **return m1,m2**  **print(max2([5233, 2162, 125, 9227, 42, 129]))**  **print(max2([23, 21, 65, 97, 42, 129, -2]))**  **print(max2(range(2, 1024, 3)))**  **print(max2(range(1, 100)))**  **print(max2([223, 271, 62, 95, 422, 229, -26]))** |

**請依下面要求輸入參數，並將輸出結果複製起來貼上對應欄位::**

|  |  |
| --- | --- |
| **print(max2([5233, 2162, 125, 9227, 42, 129]))** | (9227, 5233) |
| **print(max2([23, 21, 65, 97, 42, 129, -2]))** | (129, 97) |
| **print(max2(range(2, 1024, 3)))** | (1022, 1019) |
| **print(max2(range(1, 100)))** | (99, 98) |
| **print(max2([223, 271, 62, 95, 422, 229, -26]))** | (422, 271) |

**執行結果擷圖(說明: 請直接將下列程式碼複製貼上並且執行，再將包含指令與輸出結果的完整畫面擷取下來貼上)：**

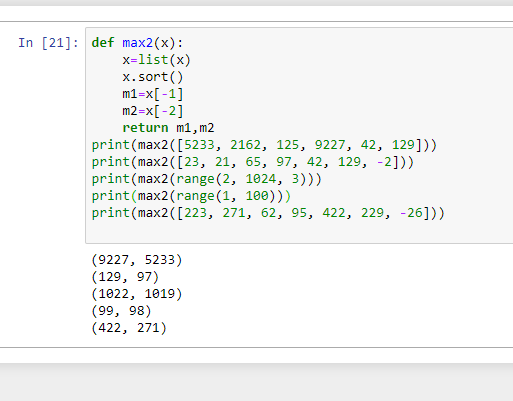
print(max2([5233, 2162, 125, 9227, 42, 129]))

print(max2([23, 21, 65, 97, 42, 129, -2]))

print(max2(range(2, 1024, 3)))

print(max2(range(1, 100)))

print(max2([223, 271, 62, 95, 422, 229, -26]))



3: 試撰寫一個可畫空心正方形的函式，命名為drawsquare(rows)，當輸入的邊長參數分別為7和9時，其執行結果的擷圖如果應如下：

|  |  |
| --- | --- |
|  |  |

我的作答：

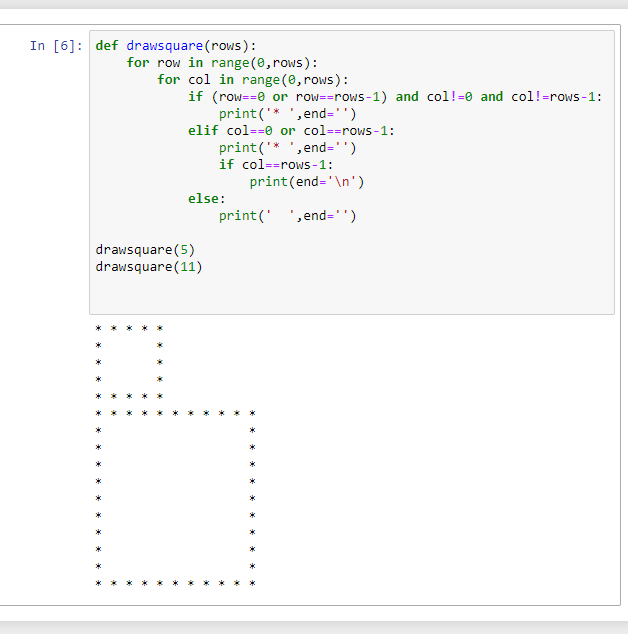
**請在下面欄位貼上程式碼：**

|  |
| --- |
| **def drawsquare(rows):**  **for row in range(0,rows):**  **for col in range(0,rows):**  **if (row==0 or row==rows-1) and col!=0 and col!=rows-1:**  **print('\* ',end='')**  **elif col==0 or col==rows-1:**  **print('\* ',end='')**  **if col==rows-1:**  **print(end='\n')**  **else:**  **print(' ',end='')**  **drawsquare(5)**  **drawsquare(11)** |

**請依下面要求輸入參數，並將執行結果擷圖貼上對應欄位:**

|  |  |
| --- | --- |
| **drawsquare(5)** | \* \* \* \* \*  \* \*  \* \*  \* \*  \* \* \* \* \* |
| **drawsquare(11)** | \* \* \* \* \* \* \* \* \* \* \*  \* \*  \* \*  \* \*  \* \*  \* \*  \* \*  \* \*  \* \*  \* \*  \* \* \* \* \* \* \* \* \* \* \* |

**執行結果擷圖：**



4: 鳶尾花資料集是非常著名的生物資訊資料集之一，取自美國加州大學歐文分校的機器學習資料庫<http://archive.ics.uci.edu/ml/datasets/Iris>，資料的筆數為150筆，共有五個欄位：1. 花萼長度。2. 花萼寬度、3. 花瓣長度、4. 花瓣寬度、5. 類別(Class)：可分為Setosa，Versicolor和Virginica三個品種。

我們可以利用sklearn的IRIS dataset得到上述五個欄位(程式碼中**X**為前面四維特徵向量；Y為第五維資料，也就是類別(Class)):

|  |
| --- |
| **from sklearn import datasets**  **iris = datasets.load\_iris()**  **X = iris.data[:, :5]**  **Y = iris.target** |

試寫一個函式ClosestDist(lst)，當我們輸入一個有4個數值的串列(list)，函式可輸出與該筆輸入資料最相似的資料及其對應的類別。如:

**The closest feature vector is : x x x x**

**The class of the closest feature vector is: x**

例如:

|  |  |
| --- | --- |
| **輸入** | **執行結果** |
| **ClosestDist([6.25, 3.42, 5.42, 2.31])** | **The closest feature vector is : 6.2 3.4 5.4 2.3**  **The class of the closest feature vector is: 2** |
| **ClosestDist([5.1 , 2.09 , 3.55, 1.07 ])** | **The closest feature vector is : 5.0 2.0 3.5 1.0**  **The class of the closest feature vector is: 1** |
| **ClosestDist([4.95 , 3.49, 1.28, 0.31])** | **The closest feature vector is : 5.0 3.5 1.3 0.3**  **The class of the closest feature vector is: 0** |

我的作答：

**請在下面欄位貼上程式碼：**

|  |
| --- |
| **from sklearn import datasets**  **iris = datasets.load\_iris()**  **X = iris.data[:, :4]**  **Y = iris.target**  **def ClosestDist(lst):**    **ans=[]**  **com=lst**  **for i in X:**  **a=(com[0]-i[0])\*\*2**  **b=(com[1]-i[1])\*\*2**  **c=(com[2]-i[2])\*\*2**  **d=(com[3]-i[3])\*\*2**  **ans.append((a+b+c+d)\*\*0.5)**  **small=ans[0]**  **index=0**  **for i in range(0,len(ans)):**  **if small>ans[i]:**  **small=ans[i]**  **index=i**  **return iris.data[index, :4],iris.target[index]**  **print('The closest feature vector is :',ClosestDist([4.91, 3.2, 1.58, 0.19])[0])**  **print('The class of the closest feature vector is:',ClosestDist([4.91, 3.2, 1.58, 0.19])[1])**  **print('The closest feature vector is :',ClosestDist([5.85, 3.12, 4.78, 1.6])[0])**  **print('The class of the closest feature vector is:',ClosestDist([5.85, 3.12, 4.78, 1.6])[1])**  **print('The closest feature vector is :',ClosestDist([7.6, 2.56, 6.89, 2.2 ])[0])**  **print('The class of the closest feature vector is:',ClosestDist([7.6, 2.56, 6.89, 2.2 ])[1])** |

**請依下面要求輸入參數，並將輸出結果複製起來貼上對應欄位::**

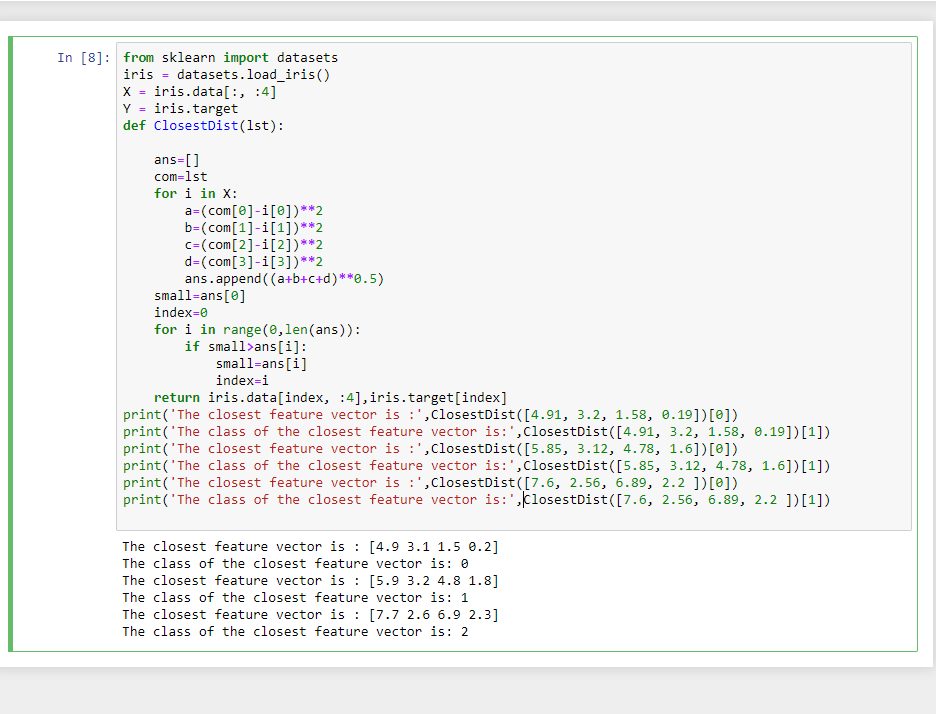
|  |  |
| --- | --- |
| **print(ClosestDist([4.91, 3.2, 1.58, 0.19]))** | **The closest feature vector is : [4.9 3.1 1.5 0.2]**  **The class of the closest feature vector is: 0** |
| **print(ClosestDist([5.85, 3.12, 4.78, 1.6]))** | **The closest feature vector is : [5.9 3.2 4.8 1.8]**  **The class of the closest feature vector is: 1** |
| **print(ClosestDist([7.6, 2.56, 6.89, 2.2 ]))** | **The closest feature vector is : [7.7 2.6 6.9 2.3]**  **The class of the closest feature vector is: 2** |

**執行結果擷圖(說明: 請直接將下列程式碼複製貼上並且執行，再將包含指令與輸出結果的完整畫面擷取下來貼上)：**

print(ClosestDist([4.91, 3.2, 1.58, 0.19]))

print(ClosestDist([5.85, 3.12, 4.78, 1.6]))

print(ClosestDist([7.6, 2.56, 6.89, 2.2 ]))



5: 請以Python的for-loop等功能來實現一個函式changeWord(str)，該函式可將輸入字串str轉成list (串列)回傳(提示: 可利用str.split()的功能)，但若是字串中有出現'good'，則將之變更為'excellent'；當字串中有出現'bad'則直接略過(提示: 直接跳過迴圈剩餘的敍述，繼續執行下一次的迴圈運作)。

例如:

|  |  |
| --- | --- |
| **輸入** | **執行結果** |
| **changeWord('This is a beautiful dog')** | **['This', 'is', 'a', 'beautiful', 'dog']** |
| **changeWord('This is a good dog')** | **['This', 'is', 'a', 'excellent', 'dog']** |
| **changeWord('This is a bad dog')** | **['This', 'is', 'a', 'dog']** |

我的作答：

**請在下面欄位貼上程式碼：**

|  |
| --- |
| **def changeWord(str):**  **lst=[]**  **str2=str.split(' ')**  **for i in str2:**  **if i=="good":**  **i="excellent"**  **if i=="bad":**  **continue**  **lst.append(i)**  **return lst**  **print(changeWord('What a beautiful day!'))**  **print(changeWord('He is also a good man in a tough business.'))**  **print(changeWord('President Donald Trump said he will not accept a bad deal.'))** |

**請依下面要求輸入參數，並將輸出結果複製起來貼上對應欄位::**

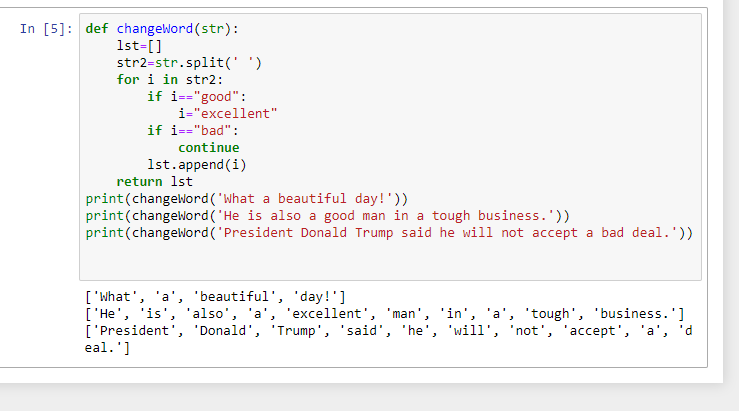
|  |  |
| --- | --- |
| **print(changeWord('What a beautiful day!'))** | **['What', 'a', 'beautiful', 'day!']** |
| **print(changeWord('He is also a good man in a tough business.'))** | **['He', 'is', 'also', 'a', 'excellent', 'man', 'in', 'a', 'tough', 'business.']** |
| **print(changeWord('President Donald Trump said he will not accept a bad deal.'))** | **['President', 'Donald', 'Trump', 'said', 'he', 'will', 'not', 'accept', 'a', 'deal.']** |

**執行結果擷圖：**

print(changeWord('What a beautiful day!'))

print(changeWord('He is also a good man in a tough business.'))

print(changeWord('President Donald Trump said he will not accept a bad deal.'))



6: 試實現一個python函式，當使用者輸入一個字串，該程式能在這個字串最末端加上'ing'；但是當字串的最末三個字元已經是'ing'時，則在'ing'後面加上'ly'。

例如:

|  |  |
| --- | --- |
| **輸入** | **執行結果** |
| **print(add\_string('circuit'))** | **circuiting** |
| **print(add\_string('string'))** | **stringly** |
| **print(add\_string('ab'))** | **abing** |

我的作答：

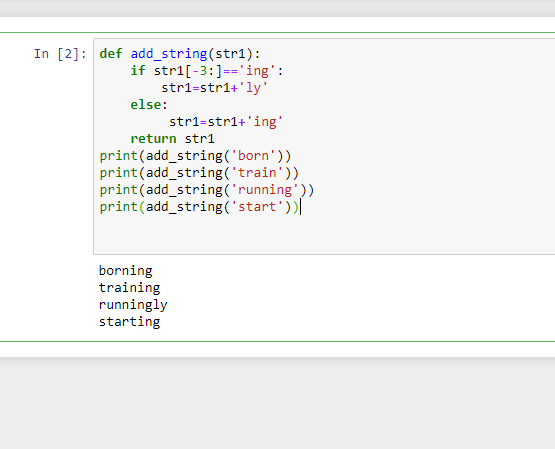
**請在下面欄位貼上程式碼：**

|  |
| --- |
| **def add\_string(str1):**  **if str1[-3:]=='ing':**  **str1=str1+'ly'**  **else:**  **str1=str1+'ing'**  **return str1**  **print(add\_string('born'))**  **print(add\_string('train'))**  **print(add\_string('running'))**  **print(add\_string('start'))** |

**請依下面要求輸入參數，並將輸出結果複製起來貼上對應欄位:**

|  |  |
| --- | --- |
| **print(add\_string('born'))** | **borning** |
| **print(add\_string('train'))** | **training** |
| **print(add\_string('running'))** | **runningly** |
| **print(add\_string('start'))** | **starting** |

**執行結果擷圖：**



7: 請以Python來實現一個函式make\_ends(lst)，該函式可將輸入一個由整數所構成，且長度至少為4的串列(list)之前二個元素與最末兩個元素取出，組成一個新的串列；接下來，我們針對這個新串列中之元素進行奇偶數的判斷，當元素為偶數的話，則在一個對應的串列中增加一個'Even'的字串元素，否則增加'Odd'的字串元素，完成上述要求後再這個對應的串列回傳。

例如:

|  |  |
| --- | --- |
| **輸入** | **執行結果** |
| **print(make\_ends([12, 6, 35, 24, 13]))** | **['Even', 'Even', 'Even', 'Odd']** |
| **print(make\_ends([2, 3, 4, 9]))** | **['Even', 'Odd', 'Even', 'Odd']** |
| **print(make\_ends([122, 33, 24, 137, 1]))** | **['Even', 'Odd', 'Odd', 'Odd']** |
| **print(make\_ends([216, 63, 14, 32]))** | **['Even', 'Odd', 'Even', 'Even']** |

我的作答：

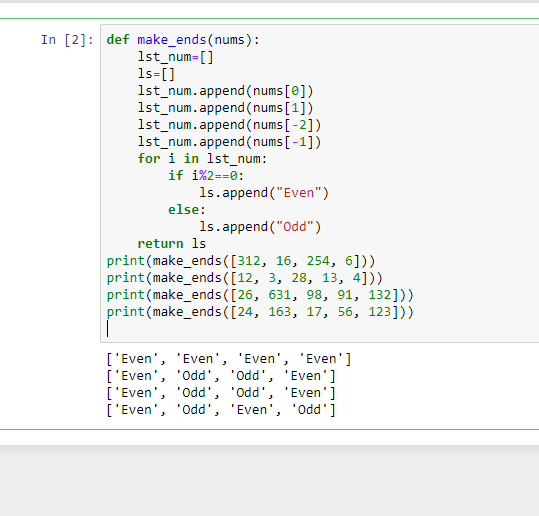
**請在下面欄位貼上程式碼：**

|  |
| --- |
| **def make\_ends(nums):**  **lst\_num=[]**  **ls=[]**  **lst\_num.append(nums[0])**  **lst\_num.append(nums[1])**  **lst\_num.append(nums[-2])**  **lst\_num.append(nums[-1])**  **for i in lst\_num:**  **if i%2==0:**  **ls.append("Even")**  **else:**  **ls.append("Odd")**  **return ls**  **print(make\_ends([312, 16, 254, 6]))**  **print(make\_ends([12, 3, 28, 13, 4]))**  **print(make\_ends([26, 631, 98, 91, 132]))**  **print(make\_ends([24, 163, 17, 56, 123]))** |

**請依下面要求輸入參數，並將輸出結果複製起來貼上對應欄位:**

|  |  |
| --- | --- |
| **print(make\_ends([312, 16, 254, 6]))** | **['Even', 'Even', 'Even', 'Even']** |
| **print(make\_ends([12, 3, 28, 13, 4]))** | **['Even', 'Odd', 'Odd', 'Even']** |
| **print(make\_ends([26, 631, 98, 91, 132]))** | **['Even', 'Odd', 'Odd', 'Even']** |
| **print(make\_ends([24, 163, 17, 56, 123]))** | **['Even', 'Odd', 'Even', 'Odd']** |

**執行結果擷圖：**



8: 請以Python來實現一個函式uniq\_items(lst)，該函式可將一個由整數所構成的串列(list)之重覆出現的項目刪除再回傳，僅保留不重覆的元素(提示:可利用in和not in的運算子來檢查元素是否在某一個串列之中)。

例如:

|  |  |
| --- | --- |
| **輸入** | **執行結果** |
| **print(uniq\_items([10,20,30,20,10,50,60,40,80,50,40]))** | **[10, 20, 30, 50, 60, 40, 80]** |
| **print(uniq\_items([2,3,2,5,6,5,6,4,8,15,13]))** | **[2, 3, 5, 6, 4, 8, 15, 13]** |
| **print(uniq\_items([5,44,32,15,7,98,32,40,32,5,15]))** | **[5, 44, 32, 15, 7, 98, 40]** |
| **print(uniq\_items([7,77,777,77,10,7]))** | **[7, 77, 777, 10]** |

我的作答：

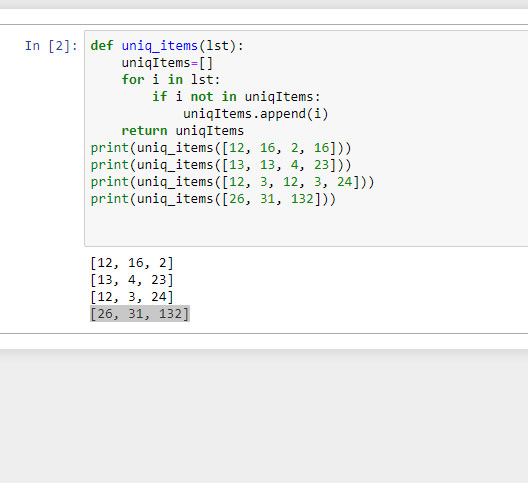
**請在下面欄位貼上程式碼：**

|  |
| --- |
| **def uniq\_items(lst):**  **uniqItems=[]**  **for i in lst:**  **if i not in uniqItems:**  **uniqItems.append(i)**  **return uniqItems**  **print(uniq\_items([12, 16, 2, 16]))**  **print(uniq\_items([13, 13, 4, 23]))**  **print(uniq\_items([12, 3, 12, 3, 24]))**  **print(uniq\_items([26, 31, 132]))** |

**請依下面要求輸入參數，並將輸出結果複製起來貼上對應欄位:**

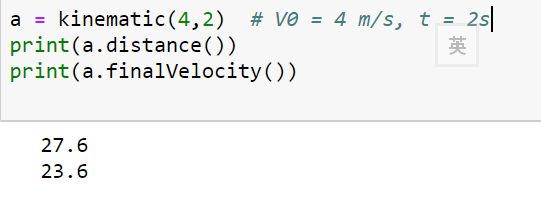
|  |  |
| --- | --- |
| **print(uniq\_items([12, 16, 2, 16]))** | **[12, 16, 2]** |
| **print(uniq\_items([13, 13, 4, 23]))** | **[13, 4, 23]** |
| **print(uniq\_items([12, 3, 12, 3, 24]))** | **[12, 3, 24]** |
| **print(uniq\_items([26, 31, 132]))** | **[26, 31, 132]** |

**執行結果擷圖：**



9: 對於運動學公式: S = Vot + 1/2 gt2，V = Vo + gt；其中Vo代表初速，t為時間，g = 9.8。試寫出一個命名為kinematic的Python類別，該類別的輸入參數為初速Vo和時間t，而類別裡的方法(distance, finalVelocity)分別能計算出移動距離S和末速V。

例如:



我的作答：

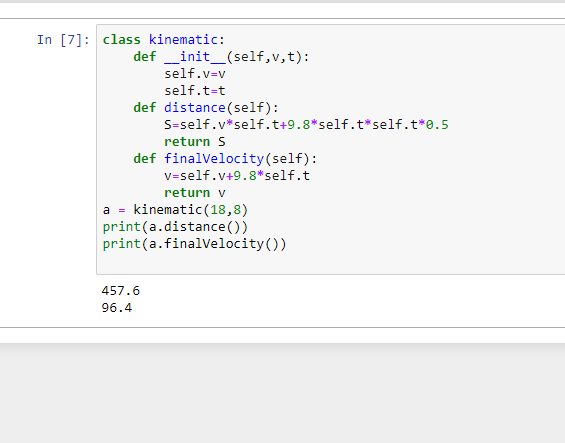
**請在下面欄位貼上程式碼：**

|  |
| --- |
| **class kinematic:**  **def \_\_init\_\_(self,v,t):**  **self.v=v**  **self.t=t**  **def distance(self):**  **S=self.v\*self.t+9.8\*self.t\*self.t\*0.5**  **return S**  **def finalVelocity(self):**  **v=self.v+9.8\*self.t**  **return v**  **a = kinematic(18,8)**  **print(a.distance())**  **print(a.finalVelocity())** |

**請依下面要求輸入參數，並將輸出結果複製起來貼上對應欄位::**

|  |  |
| --- | --- |
| **a = kinematic(18,8)**  **print(a.distance())**  **print(a.finalVelocity())** | 457.6  96.4 |

**執行結果擷圖：**



10: 試撰寫一個函式UpperEqualWord(str)，在函式裡將輸入的字串以空格為分割點，轉換成一個串列(list)；此外，程式要將串列中與第一個元素(word)相同的所有元素全改成大寫，包含第一個元素。

例如:

|  |  |
| --- | --- |
| **輸入** | **執行結果** |
| **print(UpperEqualWord('digital theorey and digital circuit design'))** | **['DIGITAL', 'theorey', 'and', 'DIGITAL', 'circuit', 'design']** |
| **print(UpperEqualWord('a dog and a cat'))** | **['A', 'dog', 'and', 'A', 'cat']** |
| **print(UpperEqualWord('Peter and Jane and me'))** | **['PETER', 'and', 'Jane', 'and', 'me']** |

我的作答：

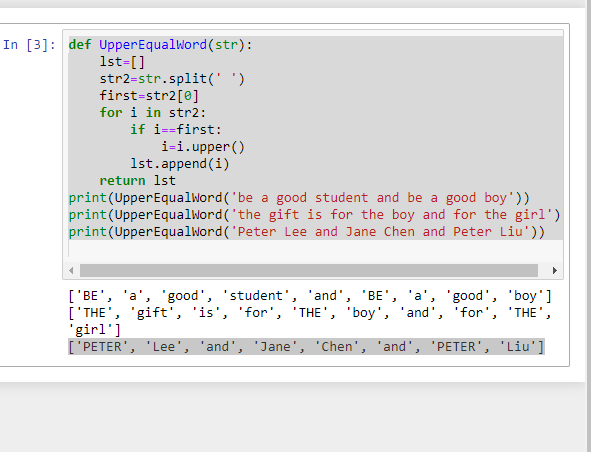
**請在下面欄位貼上程式碼：**

|  |
| --- |
| **def UpperEqualWord(str):**  **lst=[]**  **str2=str.split(' ')**  **first=str2[0]**  **for i in str2:**  **if i==first:**  **i=i.upper()**  **lst.append(i)**  **return lst**  **print(UpperEqualWord('be a good student and be a good boy'))**  **print(UpperEqualWord('the gift is for the boy and for the girl'))**  **print(UpperEqualWord('Peter Lee and Jane Chen and Peter Liu'))** |

**請依下面要求輸入參數，並將輸出結果複製起來貼上對應欄位::**

|  |  |
| --- | --- |
| **print(UpperEqualWord('be a good student and be a good boy'))** | ['BE', 'a', 'good', 'student', 'and', 'BE', 'a', 'good', 'boy'] |
| **print(UpperEqualWord('the gift is for the boy and for the girl'))** | ['THE', 'gift', 'is', 'for', 'THE', 'boy', 'and', 'for', 'THE', 'girl'] |
| **print(UpperEqualWord('Peter Lee and Jane Chen and Peter Liu'))** | ['PETER', 'Lee', 'and', 'Jane', 'Chen', 'and', 'PETER', 'Liu'] |

**執行結果擷圖：**

****