**【作業6-1】**試寫一個自訂函數，計算下列方程式的值:

n若輸入10，result=0.99902344

**示範程式碼**

|  |  |
| --- | --- |
| 1  2  3  4  5 6 7 | def function\_calculate(n):  result = 0  for x in range(1,n+1,1):  result += 1/(2\*\*x)  return(result)  n = int(input('請輸入n值:'))  print('輸出值為 : %f' %(function\_calculate(n))) |

**【作業6-2】**試寫一個自訂函數產生魔術方陣，請設計一程式，要求使用者輸入階數n，程式能自動印出魔術方陣之內容，該方塊之各列和各行和與對角線之和均相等。程式中必須有須有判斷範圍的程式，若是超出題目所訂定的數值範圍則要求重新輸入。當輸入“0”時即結束此程式。條件: n<=10，且為奇數，假設，n=5，魔術方陣參考如下:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 17 | 24 | 1 | 8 | 15 |
| 23 | 5 | 7 | 14 | 16 |
| 4 | 6 | 13 | 20 | 22 |
| 10 | 12 | 19 | 21 | 3 |
| 11 | 18 | 25 | 2 | 9 |

**示範程式碼**

|  |  |
| --- | --- |
| 1  2  3  4  5 6 7 8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40 | def magic(n):  square = []  for i in range(n + 1):  square.append([0] \* (n + 1))  i = 0  j = (n + 1) // 2  for key in range(1, n \*\* 2 + 1):  if key % n == 1:  i += 1  else:  i -= 1  j += 1  if i == 0:  i = n  if j > n:  j = 1  square[i][j] = key  matrix = []    for i in range(n):  matrix.append([0] \* n)  for k in range(len(matrix)):  for l in range(len(matrix[0])):  matrix[k][l] = square[k + 1][l + 1]  return matrix  #給予初始數值  num = eval(input("請輸入階數n(不得大於10的奇數，輸入0則結束程式)："))  #若num不等於0，則程序會一直執行  while num != 0:  #驗證條件  if num > 0 and num <= 10 and num % 2 == 1:  matrix = magic(num)  for i in matrix:  print(i)  else:  print("請輸入正確的數值")  #輸入下一個數值  num = eval(input("請輸入階數n(不得大於10，輸入0則結束程式)：")) |

**【作業6-3】**試以自訂函數依串列模擬一維陣列，依照下列的公式來計算 sum 的值，並將其值印出。

假設 a與b的串列元素如下:

a=[18, -51, 23, 35, 10, 9, -3, 52, 81, 69]

b=[28, 32, -35, 40, 73, 17, 92, 32, 13, 29]

sum=7226

**示範程式碼**

|  |  |
| --- | --- |
| 1  2  3  4  5 6 7  8  9 | a = [18, -51, 23, 35, 10, 9, -3, 52, 81, 69]  b = [28, 32, -35, 40, 73, 17, 92, 32, 13, 29]  def sum\_function(n):  list\_sum = 0  for m in range(0,n+1,1):  list\_sum = list\_sum + (a[m] \* b[n-m])  return(list\_sum)  n = len(a) - 1  print('輸出值為 : %d' %(sum\_function(n))) |

**【作業6-4】**假設某一公司有五種產品A、B、C、D 與E，其單價分別為12、16、10、14 與15 元；而該公司共有三位銷售員，他們在某個月份的銷售量如下所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 銷售員 | 產品A | 產品B | 產品C | 產品D | 產品E |
| 1 | 33 | 32 | 56 | 45 | 33 |
| 2 | 77 | 33 | 68 | 45 | 23 |
| 3 | 43 | 55 | 43 | 67 | 65 |

試依自訂函數撰寫:

1. 每一個銷售員的銷售總金額。
2. 每一項產品的銷售總金額。
3. 有最好業績(銷售總金額為最多者)的銷售員。
4. 銷售總金額為最多的產品。

**示範程式碼**

|  |  |
| --- | --- |
| 1  2  3  4  5 6 7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45 | seller\_1 = [33, 32, 56, 45, 33]  seller\_2 = [77, 33, 68, 45, 23]  seller\_3 = [43, 55, 43, 67, 65]  product\_A = 12  product\_B = 16  product\_C = 10  product\_D = 14  product\_E = 15  def Total\_sales(A,B,C,D,E):  seller\_total\_1 = (seller\_1[0] \* A) + (seller\_1[1] \* B) +(seller\_1[2] \* C) + (seller\_1[3] \* D) + (seller\_1[4] \* E)  seller\_total\_2 = (seller\_2[0] \* A) + (seller\_2[1] \* B) + (seller\_2[2] \* C) +(seller\_2[3] \* D) + (seller\_2[4] \* E)  seller\_total\_3 = (seller\_3[0] \* A) + (seller\_3[1] \* B) + (seller\_3[2] \* C) +(seller\_3[3] \* D) + (seller\_3[4] \* E)  return(seller\_total\_1,seller\_total\_2,seller\_total\_3)  print('第1位銷售員的銷售總額為 : %d 元 \n第2位銷售員的銷售總額為 : %d 元 \n第3位銷售員的銷售總額為 : %d 元' %(Total\_sales(product\_A,product\_B,product\_C,product\_D,product\_E)))  def Total\_product\_sales(A,B,C,D,E):  product\_A\_sales = (seller\_1[0] + seller\_2[0] + seller\_3[0]) \* A  product\_B\_sales = (seller\_1[1] + seller\_2[1] + seller\_3[1]) \* B  product\_C\_sales = (seller\_1[2] + seller\_2[2] + seller\_3[2]) \* C  product\_D\_sales = (seller\_1[3] + seller\_2[3] + seller\_3[3]) \* D  product\_E\_sales = (seller\_1[4] + seller\_2[4] + seller\_3[4]) \* E  return(product\_A\_sales,product\_B\_sales,product\_C\_sales,product\_D\_sales,  product\_E\_sales)  print('第1項產品的銷售總額為 : %d 元 \n第2項產品的銷售總額為 : %d 元 \n第3項產品的銷售總額為 : %d 元 \n第4項產品的銷售總額為 : %d 元 \n第5項產品的銷售總額為 : %d 元' %(Total\_product\_sales(product\_A,product\_B,product\_C,product\_D,product\_E)))  def High\_sales\_seller(A,B,C,D,E):  sales = Total\_sales(A,B,C,D,E)  k = len(sales)  High\_sales\_reg = sales[0]  for m in range(0,k,1):  if sales[m] >= High\_sales\_reg:  High\_sales\_reg = sales[m]  seller = m+1  return(seller)  print('有最好業績的銷售員是第 %d 位銷售員' %(High\_sales\_seller(product\_A,product\_B,product\_C,product\_D,product\_E)))  def High\_sales\_product(A,B,C,D,E):  sales\_product = Total\_product\_sales(A,B,C,D,E)  k = len(sales\_product)  High\_sales\_product\_reg = sales\_product[0]  for m in range(0,k,1):  if sales\_product[m] >= High\_sales\_product\_reg:  High\_sales\_product\_reg = sales\_product[m]  product = m+1  return(product)  print('銷售總金額為最多的產品是第 %d 個產品' %(High\_sales\_product(product\_A,product\_B,product\_C,product\_D,product\_E))) |

**【作業6-5】**當X, Y兩隨機變數不互為獨立時，表示兩者間有關連。其關連的形式有很多種，最常見的關連為線性的共變關係。隨機變數X,Y間的線性關係可用兩個統計量來測量共變數(covariance)，共變數能衡量兩變數之間是否有線性關係存在。依計算樣本X與Y的共變數公式如下



共變數的範例





**示範程式碼**

|  |  |
| --- | --- |
| 1  2  3  4  5 6 7  8  9  10  11  12  13  14  15  16  17  18  19 | Xi = [3,5,1,6,8,7]  Yi = [6,12,3,13,14,12]  k = len(Xi)  def co\_var(X,Y):  X\_sum = 0  Y\_sum = 0  for m in range(0,k,1):  X\_sum = X\_sum + X[m]  Y\_sum = Y\_sum + Y[m]  Xii = X\_sum / k  Yii = Y\_sum / k  S\_xy\_sum = 0  for n in range(0,k,1):  S\_xy\_sum += ((X[n] - Xii) \* (Y[n] - Yii))  S\_xy = S\_xy\_sum / (k-1)  return(S\_xy)  P = co\_var(Xi,Yi)  print(P) |

**【作業6-6】**試寫一個自訂函數計算有效年利率(effective annual rate, EAR)公式如下:

其中，i為名目年利率；m為1年複利的次數。

假設名目年利率為12%，部份輸出結果參考如下:

|  |  |  |
| --- | --- | --- |
| 複利時間單位 | 每年複利次數 | 有效年利率(EAR) |
| 年 | 1 | 12% |
| 半年 | 2 | 12.36% |
| 季 | 4 |  |
| 月 | 12 |  |
| 日 | 365 |  |

**示範程式碼**

|  |  |
| --- | --- |
| 1  2  3  4  5 6 7  8  9  10  11  12  13  14  15 | i = 0.12  def EAR\_function(x):  m = 1  EAR\_y = round((((1 + (x/m))\*\*m) - 1)\*100,2)  m = 2  EAR\_h\_y = round((((1 + (x/m))\*\*m) - 1)\*100,2)  m = 4  EAR\_q = round((((1 + (x/m))\*\*m) - 1)\*100,2)  m = 12  EAR\_m = round((((1 + (x/m))\*\*m) - 1)\*100,2)  m = 365  EAR\_d = round((((1 + (x/m))\*\*m) - 1)\*100,2)  return(EAR\_y,EAR\_h\_y,EAR\_q,EAR\_m,EAR\_d)  print('年 : %.2f \n半年 : %.2f \n季 : %.2f \n月 : %.2f \n日 : %.2f' %(EAR\_function(i))) |

**【作業6-7】**撰寫一個資本預算的程式，對企業長期投資案的資金進行規劃與決策。投資方案的評估方法包括「淨現值法, NPV」。計算公式說明如下:NPV: 將各期現金流量折現到期初之後加總。

其中，CFt為第t期的現金流量，可為正也可為負，CF0為第0期現金流量；k為折現率5%；n為投資計劃的期數。

資料來源: 假設某公司有A、B、C三個互斥投資方案要進行評估分析，現金流量如下:

|  |  |  |  |
| --- | --- | --- | --- |
| 年度 | A方案 | B方案 | C方案 |
| 0 | ($3000) | ($3000) | (1500) |
| 1 | 1900 | 600 | 500 |
| 2 | 1100 | 800 | 1200 |
| 3 | 800 | 1200 | 150 |
| 4 | 300 | 2000 | 160 |

部份輸出結果參考如下:

|  |  |
| --- | --- |
| 方案 | NPV |
| A | $745.14 |
| B |  |
| C |  |

**示範程式碼**

|  |  |
| --- | --- |
| 1  2  3  4  5 6 7  8  9  10  11  12  13  14  15  16  17 | program\_A = [-3000,1900,1100,800,300]  program\_B = [-3000,600,800,1200,2000]  program\_C = [-1500,500,1200,150,160]  n = len(program\_A)  k = 0.05  def NPV\_function(x,y):  NPV\_A = 0  NPV\_B = 0  NPV\_C = 0  for i in range(0,x,1):  NPV\_A = NPV\_A + program\_A[i]/(1+y)\*\*i  NPV\_B = NPV\_B + program\_B[i]/(1+y)\*\*i  NPV\_C = NPV\_C + program\_C[i]/(1+y)\*\*i  return(NPV\_A,NPV\_B,NPV\_C)  print('A方案的NPV為: $%.2f \nB方案的NPV為: $%.2f \nC方案的NPV為: $%.2f' %(NPV\_function(n,k))) |

**【作業6-8】**經濟訂購量: 試寫一個函數計算經濟訂購量(Q\*)。Q\*用來決定每年持有成本及訂購成本，總合最小化的訂購量。存貨不能過多或過少，以期將存貨成本降至最低。

Q\*＝

Q\*表示每批經濟訂購單數量

S表示一段期間存貨之總需求量

O表示每次訂購之訂購成本

C表示某段期間每單位存貨的儲存成本

試算範例: 假設每年需要60000單位原料，每次訂購成本120元，每單位儲存成本10元，則最每次訂購量為1200單位。

**示範程式碼**

|  |  |
| --- | --- |
| 1  2  3  4  5 6 7 | S = 60000 #S表示一段期間存貨之總需求量  O = 120 #O表示每次訂購之訂購成本  C = 10 #C表示某段期間每單位存貨的儲存成本  def Q\_function(x,y,z):  Q = ((2\*x\*y)/z)\*\*(1/2) #Q表示每批經濟訂購單數量  return(Q)  print('訂購量為: %d 單位' %(Q\_function(S,O,C))) |

**【作業6-9】**會計的折舊方法-生產數量法公式如下:

　　　折舊費用=單位折舊率本年實際使用量

　　　延用【範例6-3】的資料，另預計總使用量為200,000人次，則

單位折舊率＝＝0.45

若實際使用量為100,000則折舊費用＝0.45100,000＝45,000

每年提列的折舊費用，視當年實際使用量而定，假設1到10年的實際使用量見下表，請寫以自訂函數撰寫方法完成下表其他資料。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 年度 | 實際使用量 | 折舊費用 | 累計折舊 | 期末帳面價值 |
| 1 | 10000 | 4500 | 4500 | 99500 |
| 2 | 18000 |  |  |  |
| 3 | 12000 |  |  |  |
| 4 | 13000 |  |  |  |
| 5 | 15000 |  |  |  |
| 6 | 20000 |  |  |  |
| 7 | 25000 |  |  |  |
| 8 | 22000 |  |  |  |
| 9 | 28000 |  |  |  |
| 10 | 27000 |  |  |  |

**示範程式碼**

|  |  |
| --- | --- |
| 1  2  3  4  5 6 7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30 | cost = 100000  salvage = 10000  use = 200000  Actual\_use = [10000,18000,12000,13000,15000,20000,25000,22000,28000,27000]  use\_len = len(Actual\_use)  def D\_rate(A,B,C):  return((A-B)/C)  def D\_cost(A,B,C,D):  row = []  Acc\_exp = 0  balance = A  result = []  for x in range(0,D,1):  exp = D\_rate(A,B,C) \* Actual\_use[x]  Acc\_exp = Acc\_exp + exp  balance = balance - exp  row.append(x+1)  row.append(exp)  row.append(Acc\_exp)  row.append(balance)  result.append(row)  row = []  return(result)  r\_table = D\_cost(cost,salvage,use,use\_len)  print('%3s%8s%8s%8s' %('年度','折舊費用','累計折舊','期末帳面價值'))  for i in range(0,use\_len,1):  print('%4d%10d%12d%14d' %(r\_table[i][0],r\_table[i][1],r\_table[i][2],r\_table[i][3])) |

**【作業6-10】**會計的折舊方法-年數合計法公式如下:

第一年折舊費用：

第二年折舊費用：

以此類推（折舊率會不斷的改變）

延用【範例6-3】的資料，計算第折舊費用說明如下:

第一年折舊費用：＝16,364

第二年折舊費用：＝14,727

以此類推，請寫以自訂函數撰寫方法完成下表其他資料

**示範程式碼**

|  |  |
| --- | --- |
| 1  2  3  4  5 6 7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28 | cost = 100000  salvage = 10000  n = 10  def D\_cost(A,B,C):  row = []  Acc\_exp = 0  balance = A  result = []  reg = 0  for x in range(1,C+1,1):  reg = reg + x  for x in range(0,C,1):  exp = round((((C-x)/reg) \* (A-B)),0)  Acc\_exp = Acc\_exp + exp  balance = balance - exp  row.append(x+1)  row.append(exp)  row.append(Acc\_exp)  row.append(balance)  result.append(row)  row = []  return(result)  r\_table = D\_cost(cost,salvage,n)  print('%3s%8s%8s%8s' %('年度','折舊費用','累計折舊','期末帳面價值'))  for i in range(0,n,1):  print('%4d%10d%12d%14d' %(r\_table[i][0],r\_table[i][1],r\_table[i][2],r\_table[i][3])) |