14/01/2014

Informatica per la Finanza

Informatica per le Scienze Economiche

Informatica e Programmazione per il Turismo

Esercizio

- Nel foglio di lavoro la colonna A contiene una serie di date. Scrivere la funzione ***elabora()*** che:
  - determina la data più antica fra quelle presenti (il tipo Date si confronta come un tipo numerico) e la scrive nella cella D1;
  - per ogni data presente nella colonna A scrive a suo fianco nella colonna B quanti **giorni** passano fra quella data e quella più antica;
  - per ogni data presente nella colonna A scrive a suo fianco nella colonna C quanti **anni** passano fra quella data e quella più antica;
- Il numero di valori nella colonna A non è noto a priori; la colonna termina con una cella vuota.

G20

	A	B	C	D
1	15/01/1931	11323,00	31,00	15/01/1900
2	15/04/1931	11413,00	31,00	
3	10/01/1963	23006,00	63,00	
4	15/01/1900	0,00	0,00	
5	16/01/2014	41640,00	114,00	
6	10/01/2014	41634,00	114,00	
7	15/01/2000	36525,00	100,00	

Esercizio

```
Option Explicit
Sub elabora()
  Dim dMin As Date
  Dim rg As Integer, i As Integer

  rg = 1
  dMin = Cells(1, 1)
  Do While Not IsEmpty(Cells(rg, 1))
    If dMin > Cells(rg, 1) Then
      dMin = Cells(rg, 1)
    End If
    rg = rg + 1
  Loop
  Range("D1") = dMin
  Range("D1").NumberFormat = "dd/mm/yyyy;@"

  For i = 1 To rg - 1
    Cells(i, 2) = DateDiff("d", Range("D1"), Cells(i, 1))
  Next

  For i = 1 To rg - 1
    Cells(i, 3) = DateDiff("yyyy", Range("D1"), Cells(i, 1))
  Next
End Sub
```

Ricavato usando il registratore di macro

Differenza rispetto ai giorni

Differenza rispetto agli anni
(con l'inizio dell'anno scatta il nuovo anno)

Esercizio

- Il file di testo dati.txt due numeri in ogni riga. Scrivere la procedura VBA ***calcola()*** che legge riga per riga questi valori e scrive
  - nella colonna A il resto della divisione dei due numeri se questi sono interi e positivi;
  - Nella colonna B la loro differenza se non sono interi e positivi
- La scrittura termina appena finiscono i valori nel file. Nelle colonne A e B non vanno lasciati spazi vuoti.

```
32    45
16     3
4.5   12.3
3     12.5
12.1   4
451    7
```

dati.txt

	A	B
1	32	-7,8
2	1	-9,5
3	3	8,1

Esercizio

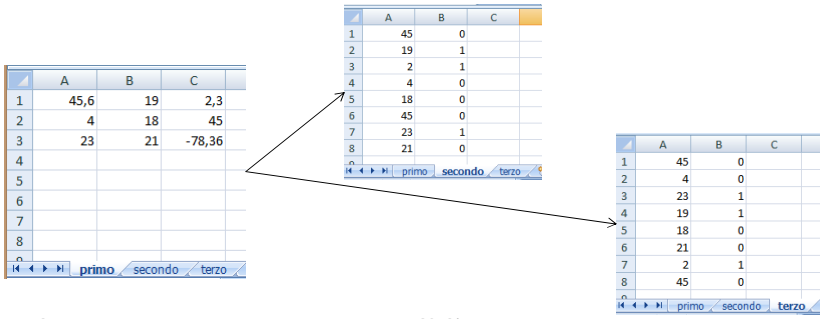
```
Option Explicit
Sub calcola()
  Dim A As Double, B As Double
  Dim rgA As Integer, rgB As Integer

  Open ThisWorkbook.Path & "\dati.txt" For Input As #1

  rgA = 1: rgB = 1
  Do While Not EOF(1)
    Input #1, A, B
    If eInteroPositivo(A) And eInteroPositivo(B) Then
      Cells(rgA, 1) = A Mod B
      rgA = rgA + 1
    Else
      Cells(rgB, 2) = A - B
      rgB = rgB + 1
    End If
  Loop

  Close (1)
End Sub

Function eInteroPositivo(A As Double) As Boolean
   eInteroPositivo = (A = CInt(A)) And (A > 0)
End Function
```

L'espressione restituisce un valore booleano

Q5 31/03/2010 (Complementi)

Scrivere un programma Matlab che

a) legge dal file Excel **dati.xls** i valori contenuti nel foglio **primo** dalla cella A1 alla cella C3 e li pone nella matrice **M**
b) crea una matrice Matlab a due colonne denominata **MT** costruita mettendo nella prima colonna tutti i numeri positivi contenuti matrice **M** troncati (usare la funzione **floor()**) e nella seconda il valore 1 se il numero nella prima colonna è primo, 0 se non lo è (usare la funzione **isprime()**).
c) scrive la matrice **MT** nel foglio di lavoro **secondo** di **dati.xls**.



Q5 31/03/2010 (Complementi)

```
function [ output_args ] = esercizio( nomeFile )

percorso=fileparts(mfilename('fullpath'));
percorso=[percorso '\' nomeFile];

M = xlsread(percorso,'primo','A1:C3');

k=1;
for i=1:size(M,1)
    for j=1:size(M,2)
        if (isnumeric(M(i,j)) & M(i,j)>0)
             MT(k,1) = floor(M(i,j));
             MT(k,2) = isprime(floor(M(i,j)));
             k=k+1;
        end
    end
end

xlswrite(percorso, MT, 'secondo','A1');
end
```

Preleva il percorso dove è salvata la funzione Matlab per conoscere dove è il file Excel

Concatenazione di Stringhe in Matlab

Come si richiama la funzione: `>> esercizio('dati.xlsx')`

Q5 31/03/2010 (Complementi)

Versione vettoriale

```
function [] = esercizio( nomeFile )
  percorso=fileparts(mfilename('fullpath'));
  percorso=[percorso '\' nomeFile];

  M = xlsread(percorso,'primo','A1:C3');

  MT = M(isnumeric(M) & M>0);
  MT = floor(MT);
  MT = [MT,isprime(MT)];

  xlswrite(percorso, MT, 'terzo','A1');
end
```

Preleva solo i valori numerici positivi

Q8 31/03/2010 (Complementi)

Nella città di Ryazan il clima si comporta nel seguente modo:

- se un giorno piove il giorno dopo piove con probabilità 0,5, è bel tempo con probabilità 0, 4, nevica con probabilità 0,1
- se un giorno è bel tempo il giorno dopo è bel tempo con probabilità 0,8, piove con probabilità 0,2
- se un giorno nevica il giorno dopo nevica con probabilità 0,3, piove con probabilità 0,6 è bel tempo con probabilità 0,1

Si chiede di:

- scrivere un programma che stima le probabilità limite attraverso simulazioni di lunghezza 100, 1000, 10000 e conserva questi valori nella matrice H formata da tre righe una per ogni lunghezza, e tre colonne una per ogni probabilità limite (piove, bel tempo, nevica)

```
                                P      B      N
MatriceTransizione = P [0.5    0.4    0.1
                       B  0.2    0.8    0
                       N  0.6    0.1    0.3]
```


Q8 31/03/2010 (Complementi)

```
function [H, lim]=ryazan(MT)
   %1: piove; 2: bello; 3: nevica
   tempo=3;  %tempo giorno 0
   freqLim=zeros(1,3);
   r = rand(1,10^4);
   MTCum = (cumsum(MT'))';

   for i=1:10000
       tempo = find(r(i) <= Mat(tempo,:),1);
       freqLim(tempo) = freqLim(tempo) +  1;
       if (i==100)
           H(1,1:3) = freqLim/100;
       elseif (i==1000)
           H(2,1:3) = freqLim/1000;
       elseif (i==10000)
           H(3,1:3) = freqLim/10000;
       end
   end

   %calcolo probabilita' limite analitica
   T=-eye(3)+MT;
   R=T';
   R(3,1:3)=ones(1,3);
   lim=R\[0;0;1];
   lim=lim';
end
```

Calcolo cumulata (*cumsum()* lavora sulle colonne)

```
>> [x y]=ryazan(MT)

x =   0.4900    0.4700    0.0400
      0.3560    0.5940    0.0500
      0.3207    0.6341    0.0452

y =   0.3111    0.6444    0.0444
```

```
[P B N] [0.5  0.4  0.1  = [ P
         0.2  0.8   0       B
         0.6  0.1  0.3]     N ]
```

```
T=[-1  0  0  + [0.5 0.4 0.1  = [ -0.5  0.4  0.1
    0 -1  0     0.2 0.8  0        0.2 -0.2   0
    0  0 -1]    0.6 0.1 0.3]      0.6  0.1 -0.7]
```

```
R=[-0.5   0.2    0.6
    0.4  -0.2    0.1
    0.1    0    -0.7]
```

```
R=[-0.5   0.2   0.6
    0.4  -0.2   0.1
     1     1     1  ]
```

Q4 19/07/2011 (Complementi)

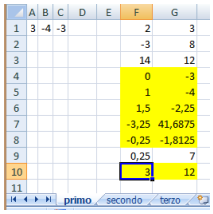
Sapendo che data una parabola di equazione $y = a \cdot x^2 + b \cdot x + c$ le coordinate del suo vertice V e del suo fuoco F sono date da:

$$V = \left(-\frac{b}{2a}, -\frac{b^2 - 4 \cdot a \cdot c}{4a} \right) \quad F = \left(-\frac{b}{2a}, \frac{1 - b^2 + 4 \cdot a \cdot c}{4a} \right)$$

scrivere la funzione Matlab/Octave **modificaExcel()** che legge dal file Excel di nome ***dati.xls*** dal foglio di lavoro **primo** la riga A1:C1 che contiene i coefficienti di una parabola (A1 contiene il coefficiente di a, B1 il coefficiente b e C1 il coefficiente c). Nello stesso foglio di lavoro nelle colonne F e G, a partire dalla riga 1 fino alla riga 30, ogni riga contiene le coordinate di un punto dello spazio cartesiano 2D.

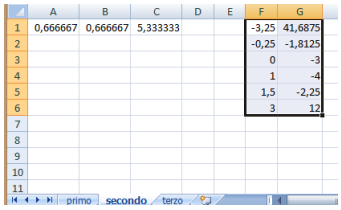
La funzione deve:

- scrivere nel foglio di lavoro **secondo** in A1, B1 il valore del vertice ed in A2,B2 il valore del fuoco della parabola;
- scrivere nel foglio di lavoro **secondo** nelle colonne C e D a partire dalla riga 1, senza lasciare spazi vuoti, i punti del foglio di lavoro **primo** che appartengono alla parabola;
- disegnare il grafico della parabola con le coordinate ricavate dal punto precedente.



	A	B	C	D	E	F	G
1	3	-4	-3			2	3
2						-3	8
3						14	12
4						0	-3
5						1	-4
6						1,5	-2,25
7						-3,25	41,6875
8						-0,25	-1,8125
9						0,25	7
10						3	12
11							

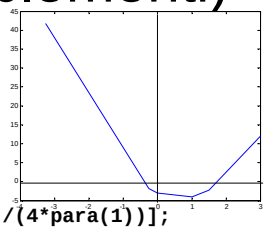
primo secondo terzo



	A	B	C	D	E	F	G
1	0,666667	0,666667	5,333333			-3,25	41,6875
2						-0,25	-1,8125
3						0	-3
4						1	-4
5						1,5	-2,25
6						3	12
7							
8							
9							
10							
11							

primo secondo terzo

Q4 19/07/2011 (Complementi)



```
function []=modificaExcel(nomeFile)
    para =xlsread(nomeFile,'primo','A1:C1');
    x = xlsread(nomeFile,'primo','F1:F30');
    y = xlsread(nomeFile,'primo','G1:G30');

    Vertice=[-para(2)/(2*para(1)),
                    -(para(2)^2-4*para(1)*para(3))/(4*para(1))];
    Fuoco = [Vertice(1),1-Vertice(2)];
    xlswrite(nomeFile, Vertice, 'secondo','A1')
    xlswrite(nomeFile, Fuoco, 'secondo','B1')

    pt=[x,y];
    pos = polyval(para,pt(:,1)) == pt(:,2);
    punti=pt(pos,:);
    punti = sortrows(punti,1);

    xlswrite(nomeFile, punti, 'secondo','F1')
    plot(punti(:,1),punti(:,2))
end
```

sort() ordina indipendentemente le colonne;
sortrows() in base alla riga tutte le colonne

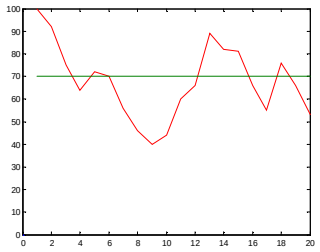
Q6 19/07/2011 (Complementi)

- Una popolazione di animali formata da **POP** elementi si comporta nel seguente modo:
  - quando la numerosità della popolazione è inferiore o uguale ad **N** (N>0) questa si incrementa casualmente di un valore compreso fra 0 ed **X** (0<X <1) di **POP**
  - quando supera il valore **N** la popolazione si decrementa per **M** volte di un valore casuale diverso compreso fra 0 ed **Y** (0<Y<1) di **POP** e quindi
    - ricomincia a crescere come nel punto precedente se la popolazione è inferiore ad N,
    - altrimenti si ripete questo punto.
- Scrivere la funzione **evoluzione()** che ha in ingresso:
  - i valori di **N**, **M**, **X** , **Y**;
  - **A**, numero di anni per cui ripetere la simulazione;
  - **POP,** popolazione iniziale**;**
- e che restituisce:
  - un messaggio di errore se i parametri in ingresso sono insufficienti;
  - un messaggio di errore se i parametri in ingresso non rispettano le condizioni date;
  - **popFinale** la popolazione dopo **A** anni.
- La funzione visualizza in un grafico l'andamento della popolazione.
- Si noti che N ed M e la popolazione sono dei valori interi e positivi o nulli e quindi i risultati andranno arrotondati (usare la funzione **round()**).

Q6 19/07/2011 (Complementi)

```
function [ popFinale ] = evoluzione( N, M, X, Y, A, POP)
    N = round(N); M = round(M); POP = round(POP);  A = round (A);
    if (N<=0 || M<=0 || POP <= 0 || A <0 )  error('popolazione limite errata')   end
    if (X >=1 || X <0)   error('X errato')   end
    if (Y >=1 || Y <0)   error('Y errato')   end
    decr=0; popFinale = POP;   popEvol=[];

   for i=1:A
        if decr == 0  %non sto decrementando
            if popFinale < N
                incr = round(popFinale * rand * X);
                popFinale = popFinale + incr;
            elseif popFinale >= N
                de = round(popFinale * rand * Y);
                popFinale = popFinale - de;
                decr = M - 1;
            end
        else %sono in fase di decremento
            de = round(popFinale * rand*Y);
            popFinale = popFinale - de;
            decr = decr -1;
        end
        if popFinale < 0   popFinale = 0     end
        popEvol = [popEvol popFinale];
    end
    plot(1:A,popEvol,'r',0,0,1:A, ones(1,A)*N)
end
```

```
>> evoluzione(70, 4, 0.4, 0.2, 20, 100)
ans =     53
```



Disegna riga di riferimento per N