**Examen ordinario 12 de Enero de 2018**

**1**. Queremos realizar un programa que conste de:

* Un procedimiento llamado **LeeMatriz** que almacene enteros introducidos por teclado en una matriz **M** (**N**x**N)**, **N** constante conocida. *(0.1 puntos)*
* Un procedimiento **ExtraeFila** que reciba como parámetros una matriz **M** y un número entero **num** y devuelva un vector **V** que contenga los valores de la fila de la matriz indicada por dicho número. *(0.4 puntos)*
* Una función **Mayor** que reciba como parámetro un vector **V** de dimensión **N** y devuelva el mayor de sus valores. *(0.4 puntos)*
* Un procedimiento **CreaVector** que reciba como parámetro una matriz **M** y mediante el procedimiento y la función anteriores devuelva un vector **V** de dimensión **N** compuesto por los mayores elementos de cada fila de la matriz. *(0.7 puntos)*
* Un procedimiento **HazMatriz** que reciba como parámetro un vector **V** y devuelva una matriz cuya primera columna sea dicho vector y las posteriores se formen restando **1** a la anterior. Ningún elemento de las columnas a partir de la primera podrá tomar un valor menor que **0**. En ese caso, se rellenará el resto de la fila con 0. *(1.5 puntos)*
* Un procedimiento **EscribeMatriz** que muestre una matriz por pantalla. *(0.15 puntos)*
* Un programa principal que solicite una matriz por teclado y mediante los procedimientos anteriores cree una nueva matriz como la del ejemplo. Este programa también imprimirá por pantalla la matriz original y la resultante. *(0.25 puntos)*

**def** LeeMatriz():  
 print(**'Escribe una matriz cuadrada de tamaño {0}x{0} por filas '**.format(n))  
 m=[]  
 **for** i **in** range(n):  
 fila=input().split()  
 **for** j **in** range (n):  
 fila[j]=int(fila[j])  
 m.append(fila)  
 **return** m

**def** ExtraeFila(m,num):  
 v=m[num]  
 **return** v

**def** Mayor(v):  
 **return** max(v)  
  
**def** CreaVector(m):  
 v=[0]\*n  
 **for** i **in** range(n):  
 w=ExtraeFila(m,i)  
 v[i]=Mayor(w)  
 **return** v

**def** HazMatriz(v):  
 m2=[[0 **for** i **in** range(n)]**for** j **in** range(n)]  
 **for** i **in** range(n):  
 m2[i][0]=v[i]  
 **for** i **in** range(n):  
 **for** j **in** range(1,n):  
 **if** m2[i][j-1]>0:  
 m2[i][j]=m2[i][j-1]-1  
 **return** m2

**def** EscribeMatriz(m2):  
 print(**'La matriz resultante es '**)  
 **for** i **in** range(n):  
 **for** j **in** range(n):  
 print(**'{0:5d}'**.format(m2[i][j]),end=**' '**)  
 print()

n=4  
mat=LeeMatriz()  
vec=CreaVector(mat)  
mat2=HazMatriz(vec)  
EscribeMatriz(mat2)

1. Escribe un programa que, a partir de una matriz, calcule en otra matriz las sumas acumuladas de la primera desde la posición [1,1] hasta cada posición de la matriz. Para ello, implementa los siguientes procedimientos y funciones:

* Un procedimiento **LeeMatriz** que lea del usuario **N**x**M** (constantes conocidas) números enteros y los almacene en una matriz.*(0,1 puntos)*
* Un procedimiento **SubMatriz** que reciba una matriz, un número de fila **F** y un número de columna **C**. Este procedimiento debe devolver una matriz del mismo tamaño que la original en la que los elementos del cuadrado formado desde la posición [1,1] hasta la posición [F,C] sean iguales que en la matriz original y el resto de elementos valgan 0. *(0,9 puntos)*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Matriz original   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 1 | 4 | 7 | 3 | 8 | | 10 | 9 | 5 | 3 | 6 | | 3 | 12 | 1 | 0 | 9 | | 2 | 5 | 4 | 8 | 2 | | Submatriz hasta la posición [2, 3]   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **1** | **4** | **7** | 0 | 0 | | **10** | **9** | **5** | 0 | 0 | | 0 | 0 | 0 | 0 | 0 | | 0 | 0 | 0 | 0 | 0 | |

* Una función **SumaMatriz** que reciba una matriz y devuelva la suma de todos los elementos almacenados en ella. *(0,5 puntos)*
* Un procedimiento **Sumas** que, utilizando el procedimiento y la función anterior, reciba una matriz y cree otra nueva matriz en la que en cada posición se almacene la suma de todos los elementos desde la posición [1,1] hasta dicha posición de la matriz original. *(1,3 puntos)*
* Un programa principal que, después de leer los números de una matriz y construir la matriz de sumas acumuladas, muestre la suma de los elementos de la matriz de sumas acumuladas. *(0,7 puntos)*

**def** LeeMatriz():  
 print(**'Escribe una matriz cuadrada de tamaño {0}x{1} por filas '**.format(n,m))  
 m1=[]  
 **for** i **in** range(n):  
 fila=input().split()  
 **for** j **in** range (m):  
 fila[j]=int(fila[j])  
 m1.append(fila)  
 **return** m1

**def** Submatriz(m1,f,c):  
 m2=[[0 **for** i **in** range(m)]**for** j **in** range(n)]  
 **for** i **in** range(n):  
 **for** j **in** range(m):  
 **if** i<=f **and** j<=c:  
 m2[i][j]=m1[i][j]  
 **return** m2

**def** SumaMatriz(m3):  
 s=0  
 **for** i **in** range(n):  
 **for** j **in** range(m):  
 s=s+m3[i][j]  
 **return** s  
  
**def** Sumas(m1):  
 m4 = [[0 **for** i **in** range(m)] **for** j **in** range(n)]  
 **for** i **in** range(n):  
 **for** j **in** range(m):  
 m3=Submatriz(m1,i,j)  
 m4[i][j]=SumaMatriz(m3)  
 **return** m4  
  
**def** EscribeMatriz(m2):  
 print(**'La matriz resultante es '**)  
 **for** i **in** range(n):  
 **for** j **in** range(m):  
 print(**'{0:5d}'**.format(m2[i][j]),end=**' '**)  
 print()

n=4  
m=5  
mat=LeeMatriz()  
mat2=Sumas(mat)  
EscribeMatriz(mat2)