

MÉTODOS MATEMÁTICOS (PARTE INFORMÁTICA)

EXAMEN EXTRAORDINARIO FEBRERO 2005

Ejercicio 1.

Determinar la salida del siguiente programa C

```
#include<stdio.h>
void main( )
{
int i,j,car,nl,a[10][10],b[10][10],c[10][10];
FILE *fent;
fent=fopen("entrada.txt","r");
nl=0;
while((car=fgetc(fent))!=EOF)
{
if(car=='\n') nl++;
}
fclose(fent);
printf("nl=%d \n",nl);
fent=fopen("entrada.txt","r");
for(i=0;i<nl/2;i++)
{
for(j=0;j<nl/2;j++)
{
fscanf(fent,"%d %d",&a[i][j] ,&b[i][j]);
if(a[i][j] <15) c[i][j]=0;
else c[i][j]=a[i][j]*b[i][j];
printf("c[%d][%d]=%d \n",i,j,c[i][j]);
}
}
for(i=0;i<nl/2;i++) for(j=0;j<nl/2;j++)
printf("a[%d][%d]=%d \n",i,j,a[i][j]);

for(i=0;i<nl/2;i++) for(j=0;j<nl/2;j++)
printf("b[%d][%d]=%d \n",i,j,b[i][j]);
}
```

Cuando el fichero de datos **entrada.txt** es el siguiente:

```
12 13 14 15 16
10 20 30 40 50
125 30 2 4 6
1 1 1 3 4
12 13 14 15 16
1 2 3 4 5
fin de fichero
```

Nota: La función **fgetc** (fich) lee el carácter del fichero **fich** correspondiente a su cursor de posición. Si encuentra el final del fichero devuelve EOF.

Examen extraordinario de Febrero 2005. Solución al ejercicio 1.

```
nl=6
c[0][0]=0
c[0][1]=0
c[0][2]=160
c[1][0]=600
c[1][1]=2000
c[1][2]=3750
c[2][0]=0
c[2][1]=0
c[2][2]=0
a[0][0]=12
a[0][1]=14
a[0][2]=16
a[1][0]=20
a[1][1]=40
a[1][2]=125
a[2][0]=2
a[2][1]=6
a[2][2]=1
b[0][0]=13
b[0][1]=15
b[0][2]=10
b[1][0]=30
b[1][1]=50
b[1][2]=30
b[2][0]=4
b[2][1]=1
b[2][2]=1
```

Ejercicio 2.

Sea una colonia de 1000 bacterias. Cada bacteria se multiplica a una velocidad de $r = 0.8$ por hora. (Es decir, un individuo produce 0.8 individuos nuevos cada hora por término medio). Se desea conocer la evolución del número de bacterias que forman la colonia a lo largo del tiempo. Si se considera que la colonia de bacterias crece continuamente sin ninguna restricción, se puede modelizar este crecimiento mediante la ecuación diferencial :

$$\frac{dN}{dt} = rN, N(0) = 1000$$

donde N es el tamaño de la población y t el tiempo. Dicho proceso se llama crecimiento exponencial y la ecuación anterior tiene una solución analítica (solución exacta) que viene dada por la expresión :

$$N(t) = N(0)e^{rt}$$

Dicha ecuación se puede resolver numéricamente (solución aproximada) aplicando el algoritmo de Euler:

$$N_{i+1} = N_i + rhN_i$$

que calcula el tamaño de la población en el instante $i+1$ en función del tamaño en el instante i . El valor inicial es $N_1 = 1000$, y h es el incremento de tiempo entre dos instantes consecutivos.

Se pide:

Diseñar un programa MATLAB (fichero .m) que calcule el tamaño de la población cada media hora hasta que transcurran 10 horas mediante:

- El método de Euler, almacenando el resultado en el vector N
- La solución analítica, almacenando el resultado en el vector N_{ex}

Dicho programa obtendrá como resultado:

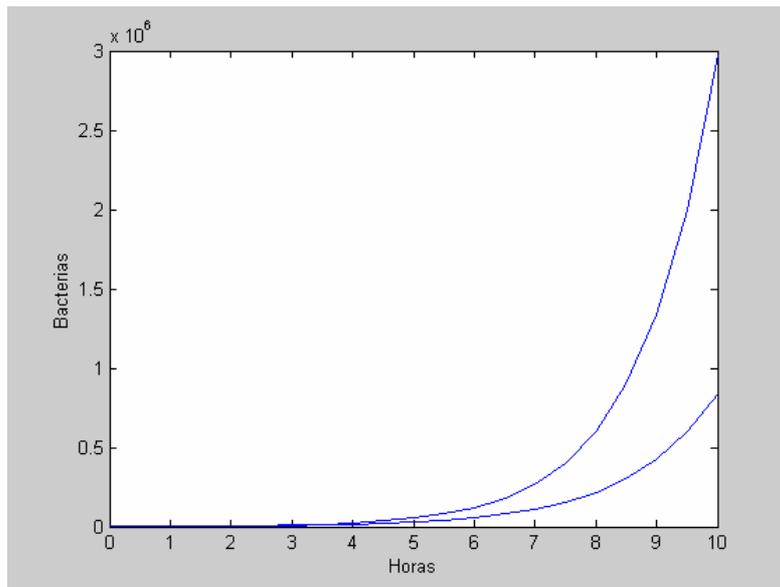
- La escritura en pantalla de una tabla de tres columnas de forma que en la primera figure el tiempo, en la segunda el número de bacterias calculado por el método de Euler y en la tercera el valor exacto resultado de aplicar la ecuación del crecimiento exponencial arriba indicada.
- Un gráfico en el que figuren las dos curvas tiempo-número de bacterias, la primera conseguida con el procedimiento numérico y la segunda con la solución analítica. Etiquetar los ejes de ordenadas y abscisas tal como aparece en el gráfico adjunto.

A continuación y para que sirva de guía, se incluye parte de la tabla de resultados tal y como se desea que quede, así como el dibujo de las curvas. **Se incluye la parte inicial del programa a diseñar**, para que ajustándose a ésta, se complete hasta finalizarlo.

(NOTA: $\exp(x)$ realiza la operación e^x)

0	1000.0000	1000.0000
0.5000	1400.0000	1491.8200
1.0000	1960.0000	2225.5400
.....		
.....		
9.5000	597630.4000	1998195.9000

10.0000 836682.5500 2980957.9900



fichero .m (programa a completar)

```
h=0.5;  
r=0.8;  
a=0;  
b=10;  
m=(b-a)/h;  
N=zeros(1, m+1);
```

completar

Examen extraordinario de Febrero 2005. Solución al ejercicio 2.

```
h=0.5;
r=0.8;
a=0;
b=10;
m=(b-a)/h;
N=zeros(1, m+1);
t=a:h:b;
N(1)=1000;
for i= 1:m
    N(i+1)=N(i) +r*h*N(i);
end
Nex =N(1) *exp(r*t);

disp ( [t' N' Nex'] )

plot(t, N), xlabel( 'Horas'), ylabel( 'Bacterias' )
hold on
plot (t, Nex), hold off
```

Ejercicio 3.

Dada una matriz real cuadrada A de dimensión n ($n < 100$), se desea calcular su inversa por el método de los determinantes. A continuación aparece la fórmula para calcular el elemento ij de la inversa de A según este método:

$$\text{Inv_A}_{(i,j)} = (-1)^{i+j} * \frac{\text{det er min ante}(X^{i,j})}{\text{det er min ante}(A)}$$

Siendo X^{ij} la matriz obtenida al eliminar la fila i y la columna j de la traspuesta de A. En la fórmula anterior los índices i, j se refieren a la fila i y columna j respectivamente comenzando la numeración en 1.

La cabecera del programa que desarrolla este método se encuentra a continuación y consta de las funciones **main**, **determinante** e **inversa**. **Se pide diseñar** las funciones **main** e **inversa** siguiendo las normas de programación indicadas a continuación:

- Las matrices no se dimensionan dinámicamente.
- Se leerá el valor de la dimensión n por teclado.
- Los elementos de la matriz A se leerán por teclado ordenados por columnas.
- El cálculo de la inversa se realizará llamando a la función **inversa**.
- Desde la función **main** se imprimirá el resultado (matriz inversa) en el fichero de datos 'datos.txt' ordenada por filas (cada fila en una línea) y con cuatro decimales de precisión seguida de la frase **La inversa de la matriz es:**
- La función **determinante** se supone dato y devuelve el determinante de la matriz A (primer argumento) de dimensión n (segundo argumento).

Dato: la función **pow(a,b)** realiza la potenciación a^b

La cabecera del programa es la siguiente:

```
#include <stdio.h>
#include <math.h>
#define DIM 100
float determinante(float A[][DIM], int n);
void inversa(float A[][DIM], int n, float A_inv[][DIM]);
```

Examen extraordinario de Febrero. Solución al ejercicio 3.

```
#include <stdio.h>
#include <math.h>
#define DIM 100
float determinante(float A[][DIM], int n);
void inversa(float A[][DIM], int n, float A_inv[][DIM]);
void main()
{
    float A[DIM][DIM],A_inv[DIM][DIM];

    int n,i,j;
    FILE* fsal;
    fsal=fopen("datos.txt","w");
    printf("Introduce la dimensión de la matriz:");
    scanf("%d",&n);
    printf("Introduce los elementos de la matriz por columnas\n");
    for(j=0;j<n;j++)for(i=0;i<n;i++)scanf("%f",&A[i][j]);
    /*llamada a la función inversa*/
    inversa(A,n,A_inv);
    fprintf(fsal,"La matriz inversa es:\n");
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)fprintf(fsal,"%f ",A_inv[i][j]);
        fprintf(fsal,"\n");
    }
    fclose(fsal);
}

void inversa(float A[][DIM], int n, float A_inv[][DIM])
{
    float A_trasp[DIM][DIM];
    float X[DIM][DIM];
    int i,j,k,t;
    /*cálculo de la traspuesta*/
    for(i=0;i<n;i++)for(j=0;j<n;j++)A_trasp[i][j]=A[j][i];

    for(i=0;i<n;i++)
        for(j=0;j<n;j++)
            {
                for(k=0;k<i;k++)
                {
                    for(t=0;t<j;t++) X[k][t]=A_trasp[k][t];
                    for(t=j+1;t<n;t++) X[k][t-1]=A_trasp[k][t];
                }
                for(k=i+1;k<n;k++)
                {
                    for(t=0;t<j;t++) X[k-1][t]=A_trasp[k][t];
                    for(t=j+1;t<n;t++) X[k-1][t-1]=A_trasp[k][t];
                }
            }
    A_inv[i][j]=(float)(pow(-1,(i+j+2)))*determinante(X,n-1)/determinante(A,n);
}
}
```