

MÉTODOS MATEMÁTICOS (PARTE INFORMÁTICA)

SEGUNDO EXAMEN PARCIAL 2004

Ejercicio 1.

Sea un conjunto de puntos en el plano. Las coordenadas de los puntos se encuentran almacenadas en un fichero de nombre 'puntos.txt', de forma que en cada línea del fichero están las dos coordenadas de uno de los puntos hasta completar tantos registros como puntos del conjunto.

Se desea obtener el valor máximo del área de todos los posibles triángulos que se pueden formar con los puntos del conjunto, así como los puntos que forman el triángulo de área máxima

Se pide **diseñar un programa C** que consiga este objetivo siguiendo obligatoriamente las normas indicadas a continuación:

- Las coordenadas de los puntos (reales de doble precisión) se leerán del fichero y se guardarán en una matriz $P[n][2]$, siendo n el número de puntos.
- La matriz P se dimensionará dinámicamente (antes se debe obtener el valor de n)
- El área de los triángulos se obtendrá mediante la llamada reiterada a la función *areatri* que también debe programarse. (El área de un triángulo se puede calcular como la mitad del módulo del producto vectorial de dos de sus lados)
- La salida a pantalla debe ser: (ejemplo)

El triangulo de área máxima está formado por los puntos 1,6,13

El valor del área máximo es: 45.230000

NOTA: Se debe comenzar el programa incluyendo los archivos cabecera indicados a continuación.

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
```

Solución al ejercicio 1

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
double areatri(double p[],double q[],double r[]);

void main()
{
    int i,j,k,n=0,p1,p2,p3;
    double **P,area_nueva,areamaxi=0;
    char aux[100];
    FILE *pf;
    pf=fopen("puntos.txt","r");
    while(!feof(pf))
    {
        fgets(aux,100,pf);
        n=n+1;
    }
    rewind(pf);
    P=(double **)malloc(n*sizeof(double *));
    if(P==NULL)
    {
        printf("Error de localización");
        exit(0);
    }
    for(i=0;i<n;i++)
    {
        P[i]=(double *)malloc(2*sizeof(double));
        if(P[i]==NULL)
        {
            printf("Error de localización");
            exit(0);
        }
    }
    for(i=0;i<n;i++)fscanf(pf,"%lf %lf",&P[i][0],&P[i][1]);

    for(i=0;i<n-2;i++)
        for(j=i+1;j<n-1;j++)
            for(k=j+1;k<n;k++)
            {
                area_nueva=areatri(P[i],P[j],P[k]);
                if (area_nueva>areamaxi)
                {
                    areamaxi=area_nueva;
                    p1=i+1;p2=j+1;p3=k+1;
                }
            }
    printf("El triangulo de area maxima esta formado por los puntos %d, %d, %d\n",p1,p2,p3);
    printf("El valor del area maximo es: %f\n",areamaxi);
}

double areatri(double p[],double q[],double r[])
{
    double v[2],w[2],a;
    int i;
    for (i=0;i<2;i++)
    {
        v[i]=q[i]-p[i];
        w[i]=r[i]-p[i];
    }
    a=1/2.*fabs(v[0]*w[1]-v[1]*w[0]);
    return a;
}
```

Ejercicio 2.

Escribir los valores de los argumentos a, b y c cuando la siguiente función de **Matlab** es ejecutada según los datos que abajo se indican:

```
function[a,b,c]=f(x,i,j)
a=[];b=[];
y1=x(:,i)
y2=x(:,j)
x1=[y1',2]
x2=[y2',7]
x3=[x(j,:),0]
x4=[x(i,:),-1]
z1=[cross(x1(2:4),x2(1:3)),5,0]
z2=x3.*x4
c=sum(x1.*x4+x3.*x4+z1.*z2);
n=length(z1);
for i=1:n
    a=[a;z1];
    b=[b;z2];
end
b=b';
d=ones(size(b));
a=[a,d];
b=b.*d;
b=blkdiag(a,b);
```

* Alumnos con número de matrícula par:

$x = [1 \ 3 \ 5 \ 1; -7 \ 2 \ 1 \ 0; -1 \ 1 \ 2 \ 1; -3 \ 2 \ 1 \ 0]$ $i=2$ y $j=4$.

* Alumnos con número de matrícula impar:

$x = [2 \ 1 \ 3 \ 5; -7 \ 2 \ 1 \ 0; -1 \ 1 \ 2 \ 2; -3 \ 2 \ 1 \ 0]$ $i=1$ y $j=3$.

Solución al ejercicio 2.

Matrícula Par:

a=

```
1 0 -1 5 0 1 1 1 1 1
1 0 -1 5 0 1 1 1 1 1
1 0 -1 5 0 1 1 1 1 1
1 0 -1 5 0 1 1 1 1 1
1 0 -1 5 0 1 1 1 1 1
```

b=

```
1 0 -1 5 0 1 1 1 1 1 0 0 0 0 0
1 0 -1 5 0 1 1 1 1 1 0 0 0 0 0
1 0 -1 5 0 1 1 1 1 1 0 0 0 0 0
1 0 -1 5 0 1 1 1 1 1 0 0 0 0 0
1 0 -1 5 0 1 1 1 1 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 21 21 21 21 21
0 0 0 0 0 0 0 0 0 0 4 4 4 4 4
0 0 0 0 0 0 0 0 0 0 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

c=28

Matrícula Impar:

a=

```
1 5 -4 5 0 1 1 1 1 1
1 5 -4 5 0 1 1 1 1 1
1 5 -4 5 0 1 1 1 1 1
1 5 -4 5 0 1 1 1 1 1
1 5 -4 5 0 1 1 1 1 1
```

b=

```
1 5 -4 5 0 1 1 1 1 1 0 0 0 0 0
1 5 -4 5 0 1 1 1 1 1 0 0 0 0 0
1 5 -4 5 0 1 1 1 1 1 0 0 0 0 0
1 5 -4 5 0 1 1 1 1 1 0 0 0 0 0
1 5 -4 5 0 1 1 1 1 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 -2 -2 -2 -2 -2
0 0 0 0 0 0 0 0 0 0 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 6 6 6 6 6
0 0 0 0 0 0 0 0 0 0 10 10 10 10 10
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

c=21

Ejercicio 3.

Se pretende escribir un programa que lea una matriz cuadrada real, por filas, del fichero de datos 'matriz.dat', y mediante la llamada a la función *suma*, escriba en pantalla la suma de todos los elementos de la matriz que no están en la diagonal principal.

Desafortunadamente en el código del programa se han infiltrado varios errores (inexactitudes u omisiones).

Se pide que el alumno **encuentre los errores** y los marque claramente explicando cuál es el error y lo que debería escribirse para corregirlo.

Nota: No se puntuará ningún error bien señalado que no venga acompañado de la explicación y corrección pertinente. (Se sugiere señalar el error en el programa con un número y en la parte posterior de la hoja proceder a la explicación y corrección)

```
#include<stdio.h>
#define filas 5

void main()
{
    int i,j;
    float M[filas][filas];
    pf=open("matriz.dat", "r");
        for (i=0;i<filas;i++)
            for (j=0;j<filas;j++)
                fscanf("%f",&M[i][j]);

printf("La suma es %f \n",suma(M));
}

int suma(float A[][])
{
    int i,j;
    float s;
    for(i=0;i<filas;i++)
        for(j=0;j>filas;j++)
            if( !(i=j))
                s=A[i][j];
    return s;
}
```

Segundo Parcial 2004. Solución al ejercicio 3

Se señalan a continuación los errores del enunciado corregidos convenientemente:

```
#include<stdio.h>
#define filas 5

float suma(float A[][filas]);

void main()
{
    int i,j;
    float M[filas][filas];
    FILE *pf;
    pf=fopen("matriz.dat","r");
    for (i=0;i<filas;i++)
        for (j=0;j<filas;j++)
            fscanf(pf,"%f",&M[i][j]);

    printf("La suma es %f \n",suma(M));
}

float suma(float A[][filas])
{
    int i,j;
    float s=0;
    for(i=0;i<filas;i++)
        for(j=0;j<filas;j++)
            if( !(i==j))
                s+=A[i][j];
    return s;
}
```