

Lea atentamente estas instrucciones antes de comenzar:

- Responda a cada una de las cuestiones prácticas.
- Es necesario aprobar este examen (5 puntos) para aprobar la asignatura.
- No es suficiente este examen para aprobar la asignatura: se deben aprobar los dos trabajos de la parte práctica.

Sistemas de álgebra computacional: Maxima

1. (1 punto) Explique qué se obtiene y qué sale por pantalla después de evaluar este código:

```
a : tan(pi)$
```

Respuesta: Si la variable pi no ha sido definida (recuérdese que el número π se representa en máxima por %pi), el lado derecho del signo de asignación (:) se dejará sin evaluar, como tan(pi) y esta expresión será asignada a la variable a. Como la instrucción finaliza con \$, no se mostrará salida por consola.

2. (1 punto) Después de evaluar este código

```
f(x) := block([a], a:3, x^a);
```

¿cuál será el valor de la variable a?

Respuesta: Este código define una función, por lo tanto, en tanto no se evalúe, no afectará al valor de ninguna variable. De todos modos, aún si se ejecutase como

```
f(x);
```

la variable a sería una variable local (se incluye en el corchete que va como primer argumento de block()), con valor sólo dentro de la función, por lo que al ejecutar ésta, a seguirá valiendo lo que valía antes de ejecutarla.

3. (2 puntos) La identidad $e^{i\pi} + 1 = 0$, debida a Euler, contiene todos los números importantes de las matemáticas. Escriba una instrucción condicional en Maxima que diga “Euler tenía razón”, en caso de que la igualdad anterior sea cierta, o “Euler estaba equivocado”, en caso de que, para Maxima, no sea así. **Nota:** para emitir los mensajes, utilice la función disp(“cadena”).

Respuesta: En Maxima la función condicional es de la forma

```
if condicion then consecuencia else alternativa
```

así que escribiríamos

```
if exp(%i*%pi)+1=0 then disp("Euler tenia razon") else disp("Euler estaba equivocado")$
```

Programación en C

1. (6 puntos) El objetivo de este ejercicio es diseñar un programa que calcule el centro de masas de un sistema discreto de partículas.
 - El programa debe estar compuesto de una única función “main”.
 - Supondremos que los datos de las N partículas se encuentran en un archivo con cuatro columnas: las tres primeras columnas corresponden a las coordenadas (x_i, y_i, z_i) de cada partícula i , mientras que la cuarta columna corresponde a la masa m_i . Aunque no conocemos exactamente el número de partículas que componen el sistema, sabemos que es menor que 1000. El nombre del archivo que contiene los datos deberá ser introducido por línea de comandos (como argumento de la función “main”) cuando se ejecute el programa.
 - Para leer y almacenar los datos de las partículas, se deberá definir una estructura denominada “Punto” que representa una partícula con su posición y masa. Después, en la función “main” de deberá declarar un “array” de “Puntos”, es decir, un array de estas estructuras, para almacenar en él la información de todo el sistema de partículas.
 - El programa debe escribir en pantalla la posición del centro de masas del sistema

$$\mathbf{r}_{CM} = \frac{\sum_{i=1}^N m_i \mathbf{r}_i}{\sum_{i=1}^N m_i}$$

```
Solución: #include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define N 1000

struct punto {
    double x, y, z, masa;
};

typedef struct punto Punto;

int main(int argc, char** argv) {
    int i, n, npts;
    double x, y, z, masa, masaCM;
    Punto P[N];
    Punto PCM;
    FILE *fin;

    fin=fopen(argv[1], "r");
    npts=0;

    do {
        n=fscanf(fin, "%lf_%lf_%lf_%lf", &x, &y, &z, &masa);
        if (n==4) {
            P[npts].x=x;
            P[npts].y=y;
            P[npts].z=z;
            P[npts].masa=masa;
            npts++;
        }
    } while (n);

    masaCM=0;
    for (i=0; i<npts; i++)
        masaCM+=P[i].masa;

    if (masaCM>0)
        printf("Centro de masas: (%f, %f, %f)\n",
            (x*P[0].masa+y*P[0].masa+z*P[0].masa)/masaCM,
            (y*P[0].masa+z*P[0].masa+x*P[0].masa)/masaCM,
            (z*P[0].masa+x*P[0].masa+y*P[0].masa)/masaCM);
    else
        printf("El centro de masas coincide con el origen.\n");

    return 0;
}
```

```
    }  
} while (n==4);  
  
fclose ( fin );  
  
PCM.x=0;  
PCM.y=0;  
PCM.z=0;  
masaCM=0;  
  
for (i=0; i<npts; i++) {  
    PCM.x+=(P[i].masa)*(P[i].x);  
    PCM.y+=(P[i].masa)*(P[i].y);  
    PCM.z+=(P[i].masa)*(P[i].z);  
    masaCM+=P[i].masa;  
}  
  
printf ("La posicion del Centro de Masas es (%g, %g, %g)\n",  
        PCM.x/masaCM, PCM.y/masaCM, PCM.z/masaCM);  
  
return 0;  
}
```