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 de manera breve y razonada qué significado tiene en Maxima el carácter "=".

   Solución: El carácter "=" en Maxima es el operador de igualdad, es decir, indica que los valores de las expresiones a un lado y al otro son iguales, estableciendo así una ecuación (o sea, no una asignación). Se emplea, por ejemplo, en la función **solve**, que recibe el primer lugar una ecuación (o sistema de ecuaciones, formado por una lista de igualdades) y en segundo lugar la variable (o lista de variables) que se quiere resolver.

2. (1 punto) ¿Qué comando se emplea en Maxima para cargar un "paquete de funciones"? Como p. ej, el paquete "dynamics" que incluye (entre otras) la función rk.

   Solución: Para cargar paquetes de funciones, se emplea la función **load** con argumento una cadena con el nombre del paquete. Por ejemplo, **load**("dynamics");.

3. (2 puntos) ¿Cómo podríamos pedirle al Maxima que calcule la integral de una función $f(x)$?

   Solución: Maxima utiliza para calcular integrales indefinidas (primitivas), de manera simbólica, el comando **integrate**(expr, x), donde expr es la expresión del integrando y x es la variable respecto a la que se integra. Por lo tanto, para calcular la integral indefinida de f(x), se haría

   ```
   integrate(f(x), x);
   ```

   Para calcular la integral definida, en el intervalo [a, b] (de nuevo, simbólicamente), se añadirían los límites del intervalo a los argumentos:

   ```
   integrate(f(x), x, a, b);
   ```

Programación en C

1. (3 puntos) En un determinado programa estamos trabajando con vectores en el espacio tridimensional definidos a través de estructuras del siguiente modo:

   ```
   struct Vector {
           double x, y, z;
   };

   typedef struct Vector vector;
   ```

Defínase una función que calcule (y devuelva) el producto vectorial de dos vectores cualesquiera, introducidos como argumentos de la función.

Solución:

```
Vector producto_vectorial(vector a, vector b) {
        vector c;
        c.x=  a.y*b.z - a.z*b.y;
        c.y= -a.x*b.z + a.z*b.x;
        c.z=  a.x*b.y - a.y*b.x;

        return c;
}
```

2. (3 puntos) Para calcular manualmente la raíz de un número hacemos una primera aproximación a ella, luego dividimos el número por esta estimación y calculamos la media entre la estimación y ese cociente; el resultado nos da una nueva estimación y volvemos a repetir el proceso. Escríbase un programa en C que realice estas iteraciones y calcule el error respecto al valor exacto obtenido con sqrt() en función del número de veces que se haya iterado. Considere números entre 1 y 10, y que la primera estimación es siempre 1; deténgase cuando el error sea inferior a 10^{-6}.

Solución:

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

int main(int argc, char** argv) {
        double x, sx, esx, msx, err;
        int n;

        printf("x= ");
        scanf("%lf", &x);

        sx=sqrt(x);

        n=1;
        esx=1;
        do {
                msx=(esx+x/esx)/2;
                err=fabs(msx-sx);

                printf("n=%d: sqrt(x) ~ %f\terr= %f\n", n, msx, err);

                esx=msx;
        } while(err > 1e-6);

        return 0;
}
```