

## 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) ¿Qué caracteres empleamos en Maxima para definir una lista?

**Respuesta:** Se emplean "[", "]" y ",". Los dos primeros indican el comienzo y final de lista, respectivamente, mientras que la coma se emplea para separar los elementos de la lista.

2. (1 punto) Explique qué se obtiene y qué sale por pantalla después de evaluar este código:

```
a : tan(%pi)$  
b : a * c^2;
```

**Respuesta:** La primera instrucción asigna a la variable a el valor 0, ya que éste es el valor de la tangente de  $\pi$ ; esto es, evalúa el lado derecho del operador de asignación y asigna ese valor a la variable que figura al lado izquierdo. Como la instrucción está terminada por un \$, no produce ninguna salida en la consola.

La segunda instrucción asigna a la variable b el valor resultante de evaluar  $a*c^2$ . Dado que en la instrucción anterior se asignó a la variable a el valor 0, el resultado será 0. Ahora, como la instrucción finaliza con un ;, se mostrará el valor en la consola.

3. (2 puntos) Escriba una función en Maxima que reciba como argumentos una lista de coeficientes (como p.ej.  $[a_0, a_1, a_2, \dots, a_n]$ ) y una variable (como p.ej.  $x$ ) y que construya a partir de este *input* el polinomio en la variable dada

$$a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

**Respuesta:** Llamemos a la función Pa(as) donde as es la lista de coeficientes; el número de coeficientes (que es el grado del polinomio más uno, vale length(as)). La forma más sencilla de calcular el polinomio es mediante la función sum() aplicada a los monomios  $a_i x^i$ :

```
Pa(as):=sum( as[i]*x^i, i, 0, length(as)-1 );
```

## Programación en C

1. (2 puntos) Escribir una función que, dada una lista de números complejos en la forma de un array de estructuras

```
struct complexf {  
    float re, im;  
};
```

devuelva el número complejo resultante de multiplicar todos ellos.

```
Solución: struct complexf complex_prod(int N, struct complexf x[]) {
    int i;
    float aux;
    struct complexf p;
        p.re=1.0;
        p.im=0.0;
    for(i=0; i < N; i++) {
        aux=p.re;
        p.re=p.re*x[i].re-p.im*x[i].im;
        p.im=aux*x[i].im+p.im*x[i].re;
    }
    return p;
}
```

2. (4 puntos) Supongamos que tenemos un archivo de datos "datos.dat" con  $N$  pares de puntos que representan el comportamiento de una función  $f(x)$  en un intervalo dado. La primera columna del archivo corresponde a los valores  $x_i$  (que están ordenados de menor a mayor) mientras que la segunda muestra los valores de la función  $f(x_i)$ .

Escribir un programa que lea esos puntos y obtenga los valores de  $x$  en los que la función muestra los mínimos y máximos relativos dentro del intervalo. El valor de  $N$  debe ser introducido por línea de comandos como argumento de la función "main" cuando se ejecuta el programa. El programa debe mostrar en pantalla el resultado del siguiente modo (es un ejemplo):

```
"La función tiene tres mínimos relativos en los puntos x = 0.1, 1.2, 4.5"
```

```
"La función tiene dos máximos relativos en los puntos x = 0.9, 2.9"
```

Del mismo modo el programa también deberá indicar si la función no tiene mínimos ni máximos relativos.

```
Solución: #include <stdio.h>
#include <stdlib.h>
```

```
int main(int argc, char** argv) {
    int N, i, nmin, nmax;
    N=atoi(argv[1]);
    FILE *fin;
    double x[N],y[N], xmin[N], xmax[N];

    fin=fopen("datos.dat", "r");
    for (i=0; i<N; i++) {
        fscanf(fin, "%lf_%lf", &x[i], &y[i]);
    }
    fclose(fin);

    nmin=nmax=0;

    for(i=1; i<N-1; i++) {
        if(y[i-1]>y[i] && y[i+1]>y[i]) {
            xmin[nmin]=x[i];
            nmin++;
        }
    }
}
```

```

    }
    if (y[i-1]<y[i] && y[i+1]<y[i]) {
        xmax[nmax]=x[i];
        nmax++;
    }
}

if (nmin==0)
    printf("La_funcion_no_tiene_ningun_minimo_relativo\n");
else {
    printf("La_funcion_tiene_%d_minimos_relativos "
           "en_los_puntos_x=_", nmin);
    for (i=0; i<nmin; i++)
        printf("%g,_", xmin[i]);
    printf("%\n");
}

if (nmax==0)
    printf("La_funcion_no_tiene_ningun_maximo_relativo\n");
else {
    printf("la_funcion_tiene_%d_maximos_relativos "
           "en_los_puntos_x=_", nmax);
    for (i=0; i<nmax; i++)
        printf("%g,_", xmax[i]);
}

return 0;
}

```