

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) Abrimos una sesión nueva de wxMaxima y tecleamos

```
pi ;  
% pi ;  
float ( pi ) ;  
float(%pi) ;
```

¿Qué output obtenemos?

Respuesta: `pi` produce como output la variable `pi` o el valor que tenga asignado, mientras que `%pi` produce como output la letra `pi` del alfabeto griego, que en este caso a diferencia del anterior, tiene asignado a efectos de cálculo el valor del número π .

Análogamente, `float(pi)` produce como output la variable `pi` o el resultado de evaluar numéricamente el valor que tenga asignado, mientras que `float(%pi)` produce 3.141592653589793, la aproximación numérica del número π .

2. (1 punto) Abrimos una sesión nueva de wxMaxima y tecleamos

```
sin ( pi / 2 ) ;  
sin(%pi / 2) ;  
sin ( float ( pi / 2 ) ) ;  
sin ( float(%pi / 2) ) ;
```

¿Qué output obtenemos?

Respuesta: `sin(pi/2)` produce como output `sin(pi/2)`, a no ser que la variable `pi` tenga un valor asignado.

`sin(%pi/2)` produce como output 1, al entenderse que es el seno del ángulo $\pi/2$.

`sin(float(pi/2))` produce como output `sin(0.5*pi)`, a no ser que la variable `pi` tenga un valor asignado.

`sin(float(%pi/2))` produce como output 1.0, en este caso, al calcular el seno del resultado de `float(%pi/2)` se pierde la precisión infinita que teníamos con `sin(%pi/2)`.

3. (1 punto) Indique la finalidad de la función `concat` en Maxima.

Respuesta: La finalidad de `concat` es concatenar cadenas de caracteres, es decir, unir cadenas de caracteres una a continuación de otra.

4. (1 punto) Abrimos una sesión nueva en wxMaxima y tecleamos

```
a:1;  
b:2$
```

Indique qué resultado obtenemos al evaluar estas instrucciones y qué output aparecerá en la pantalla.

Respuesta: El resultado es que las variables globales "a" y "b" quedan asignadas a los valores 1 y 2, respectivamente. El único output que se visualiza en pantalla es el de la primera asignación (es decir, 1), el output de la segunda asignación no se visualiza, al estar esta instrucción cerrada con el símbolo \$.

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) El objetivo de este ejercicio es escribir un programa que proporcione una estimación del número π por el método numérico de Monte Carlo. Para ello supondremos que disponemos de la función `rnd()`, declarada del siguiente modo

```
double rnd();
```

que devuelve cada vez que es llamada un número real (double) aleatorio uniformemente distribuido entre -1 y 1 . El método consiste en generar parejas de números aleatorios que corresponderán a las coordenadas de puntos (x, y) . De este modo, los puntos generados caerán en el cuadrado de lado 2 centrado en el origen de coordenadas $(0, 0)$. La probabilidad P de que un punto caiga en el círculo de radio 1 centrado en el origen vendrá dada por el área total del círculo dividida por el área del cuadrado:

$$P = \frac{\text{área}_{\text{círculo}}}{\text{área}_{\text{cuadrado}}} = \frac{\pi R^2}{(2R)^2} = \frac{\pi}{4}$$

El programa debe aproximar esta probabilidad para N puntos (valor que debe ser introducido por línea de comandos al ejecutar el programa). Si definimos como n el número de esos N puntos que han caído dentro del círculo, tendremos que

$$P \simeq \frac{n}{N}$$

Igualando este valor con el resultado anterior tenemos que nuestra estimación de π es

$$\pi \simeq \frac{4n}{N}$$

Éste es el valor que debe retornar el programa para un N dado.

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

int main(int argc, char** argv) {
    int N,n,i;
    double x,y;
    N=atoi(argv[1]);
    n=0;
    for (i=0; i<N; i++) {
        x=rnd();
        y=rnd();
        if (x*x+y*y<=1) n++;
    }
    printf("La estimacion del valor de Pi por el "
        "metodo de Monte Carlo con %d puntos es %g\n",
        N, 4.0*n/N);
    return 0;
}
```