

ECUACIONES POLINOMIALES Y ALGORITMOS

PRIMER CUATRIMESTRE 2018 – PRÁCTICA SINGULAR 1

Introducción a Singular

1. **Primeros pasos.** Al iniciar Singular, el sistema espera que el usuario ingrese una instrucción con el signo `>`. Todas las instrucciones deben terminar con `;`.

```
37 + 5;  
==> 42
```

Todos los objetos tienen un tipo, por ejemplo, las variables de tipo entero se definen con la palabra `int`. Las asignaciones se realizan con el signo `=`.

```
int k = 2;
```

Para evaluar igualdad o desigualdad se utiliza `==` resp. `!=` (o `<>`). El valor FALSO se representa por 0 y VERDADERO por 1.

```
k == 2;  
k != 2;  
k < 3;  
k >= 2;
```

¿Qué salida se obtiene con las siguientes instrucciones?

```
k;  
int j;  
j = k+1;  
j + 3;
```

2. **Anillos y polinomios.** Los tipos de dato de Singular dependen de una meta-estructura: los anillos, en los cuales viven esos datos. Las únicas excepciones son los tipos `int`, `intvec`, `intmat`, `string`, que no dependen de ningún anillo. Para realizar cualquier operación en Singular (salvo estas excepciones) es siempre necesario definir previamente el anillo sobre el que vamos a trabajar.

Algunas definiciones de anillos:

<code>ring R = 0, x, lp;</code>	$\mathbb{Q}[x]$, anillo de polinomios en una variable con coeficientes racionales y orden lexicográfico (vamos a ver órdenes más adelante). El primer 0 indica que el cuerpo de base es $\mathbb{Q}$, cuerpo de característica 0.
<code>ring S = 5, (x, y), lp;</code>	$\mathbb{Z}_5[x, y]$, anillo de polinomios en dos variables sobre el cuerpo de 5 elementos.

Ingresar los siguientes comando e interpretar los resultados (observar las distintas formas de ingresar los exponentes).

```

ring R = 3, x, lp;
basing;
7 + 1;
number a = 7 + 1;
a;
ring R = 0, x, lp;
poly p = 6x^3 - 10x + 1;
poly q = 17x^2+x;
poly s = p + q;
factorize(s);

```

3. Realizar las siguientes operaciones.

(a) Calcular el cuadrado del polinomio $p(x) = 3x^3 + 2x - 1$.

(b) Factorizar el polinomio $x^3y^3 + 2x^2y + 2xy^2 - 1$ in $\mathbb{Q}[x, y]$, $\mathbb{Z}_2[x, y]$ and $\mathbb{Z}_5[x, y]$.

4. **Mapas entre anillos.** Para utilizar datos definidos en un anillo en otro anillo, utilizamos los comandos `imap` o `fetch` (el primero respeta los nombres de las variables y el segundo la ubicación de las variables).

```

ring R = 0, (x,y,z), lp;
poly p = 6x^3y - 12zy + 1;
ring S = 5, (u, v, w), lp;
fetch(R, p);
imap(R, p);

```

Rehacer el último ejercicio utilizando este comando.

5. **Ciclos.** El lenguaje de programación en Singular es similar a C++. Por ejemplo, podemos calcular la suma de los números naturales del 1 al 100 con el siguiente ciclo.

```

int i, j;
j = 0;
for(i = 1; i <= 100; i++)
{
    j = j + i;
}
j;

```

(a) Calcular la suma de los cuadrados de 1 a 100.

(b) Calcular $50!$ mediante un ciclo `for`. (Utilizar tipo `bigint` en vez de `int` para evitar overflow.)

(c) Comenzando por un polinomio p cualquiera, utilizando ciclos y los comandos `deg` y `diff`, calcular el coeficiente principal de p . (Buscar en la ayuda de Singular cómo usar esos comandos.)