

# mini-cas

Uma biblioteca para matemática  
simbólica em C++

Arthur Gonçalves do Carmo

Luiz Carlos de Abreu Albuquerque

# Sumário

- Decisões tomadas (até agora)
  - O que há de novo
- Detalhes sobre o que já funciona



# Decisões tomadas

Implementação da classe para polinômios

-> Criação de uma classe para monômios

Classes para representação de expressões  
continuam no escopo do projeto

Classes para vetores e matrizes fora do escopo  
(mas devem ser implementadas eventualmente)



# Monômios

Resumidamente:

$$128xy^3z^2$$

é um monômio

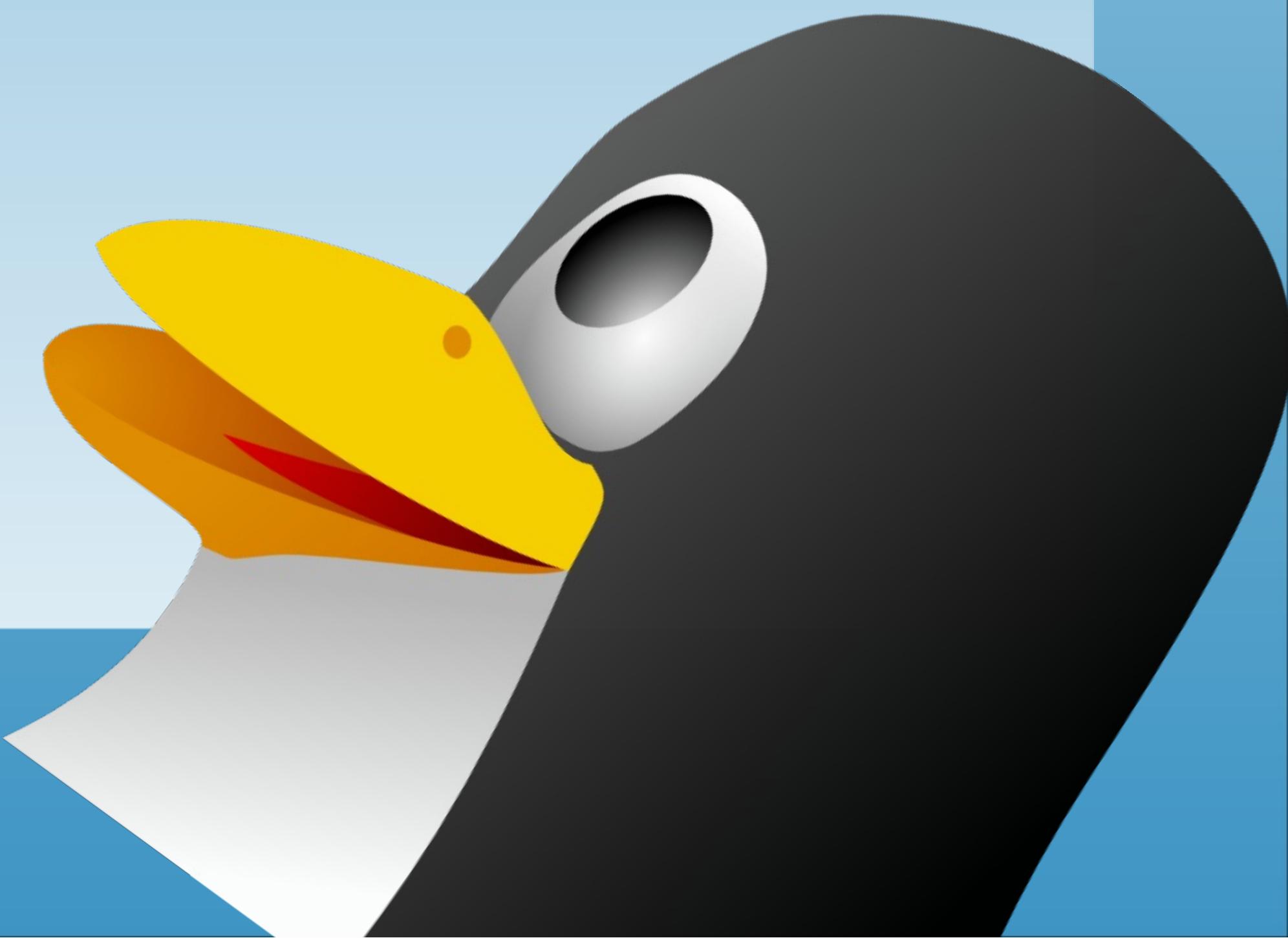
$$(1 + y) x$$

não é um monômio



# Primeira tentativa

```
num_z_exponent[26];
num_q_coefficient;
bool_variables[255];
int_var_counter;
```



# Como está agora

```
num_q_coefficient;
std::map<std::string, num_z> _literals;
num_z_degree;
```

```
monomial mono(num_q(159, 1383), 'x', 1, "z", 3, 'y', 2);
monomial unity(num_q(1), 'x', 2, 'y', 4);
monomial x('x');
```



```
template<class... Args>
monomial(const Number &coef, Args... args) {
    this->_coefficient = coef.q_value();
    if(this->_coefficient != 0)
        this->__construct_monomial(args...);
}
```

```
template<class... Args>
void __construct_monomial(const char &v,
                           const num_z &exp,
                           Args... args)
{
    ...
    this->__construct_monomial(args...);
}
```

# Polinômios

```
std::set<monomial, monomial_comp_class> _terms;
```

[\*\*monomial\\_comp\\_class:\*\*](#)

certifica que monômios semelhantes  
compartilhem um mesmo lugar na árvore



# Até isso já é bem estudado!

## Ordenação de monômios, divisão em anéis de polinômios de várias variáveis e as Bases de Groebner

Danilo Adrian Marques\* Prof. Cícero Carvalho†

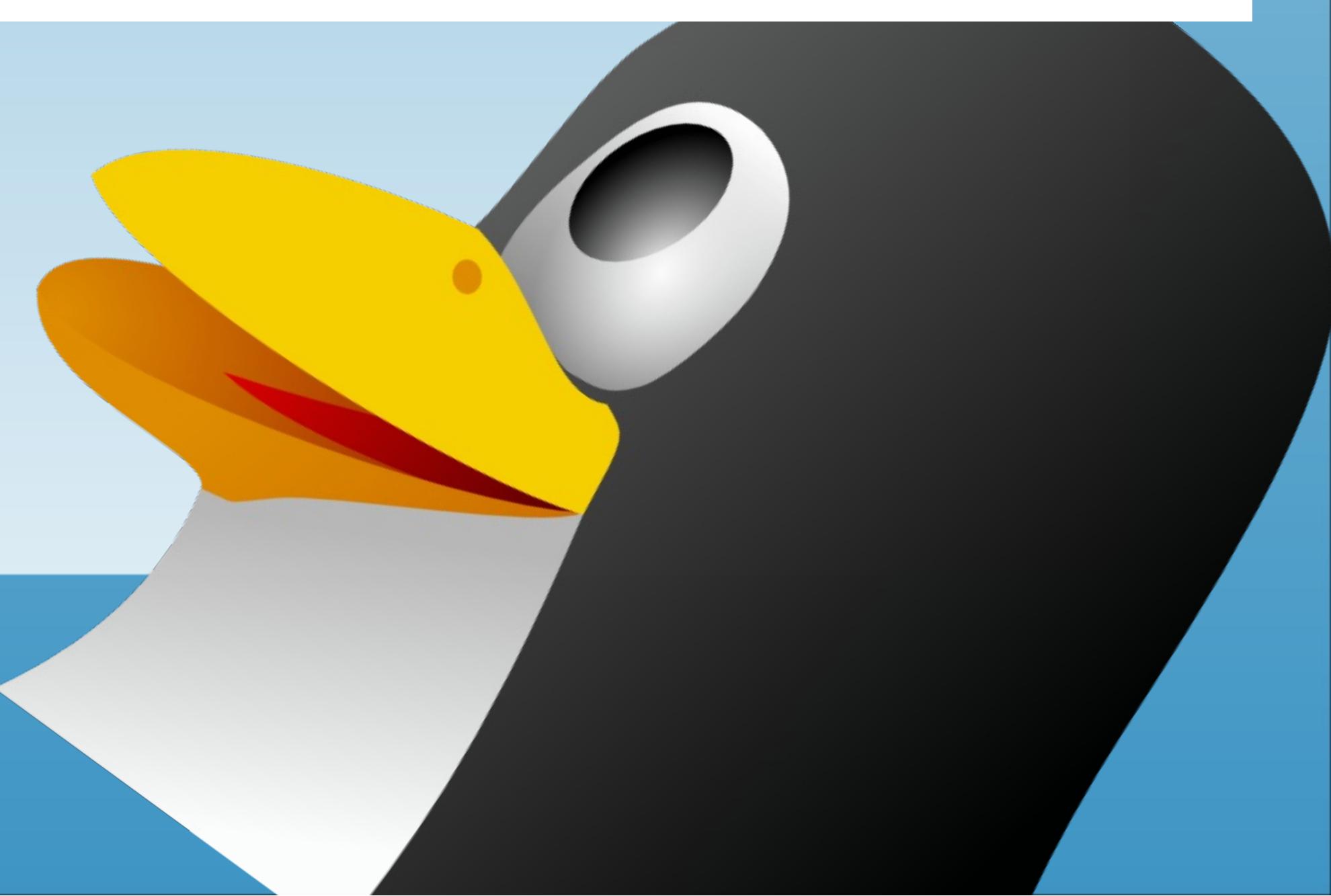
Faculdade de Matemática - FAMAT

Universidade Federal de Uberlândia - UFU

38408-100, Uberlândia - MG

Junho 2007

FAMAT em Revista nº 9, Out 2007



Basicamente é necessário que para todo monônio M, N e P:

1.  $M \leq N \iff MP \leq NP$
2.  $M \leq MP.$

Qual a ordem mais interessante para  
 $xy^2z, z^2, x^3$  e  $x^2z^2$ ?

$x^3 > x^2z^2 > xy^2z > z^2$  (lex)

$x^2z^2 > x^1y^2z^1 > x^3 > z^2$  (glex)

$x^1y^2z^1 > x^2z^2 > x^3 > z^2$  (grevlex)



# Declarando polinômios

```
monomial x('x'), y('y'), z('z');  
polynomial p1(x^2, y^2, z^2, x*y, x*z, y*z);  
cout << p1 << endl;
```

$$x^2 + x*y + y^2 + x*z + y*z + z^2$$

```
monomial x('x'), y('y');  
polynomial u = 4*x*y + 4*y + 8,  
v = y.pow(2);
```



# Operador ^ tem precedência baixa!

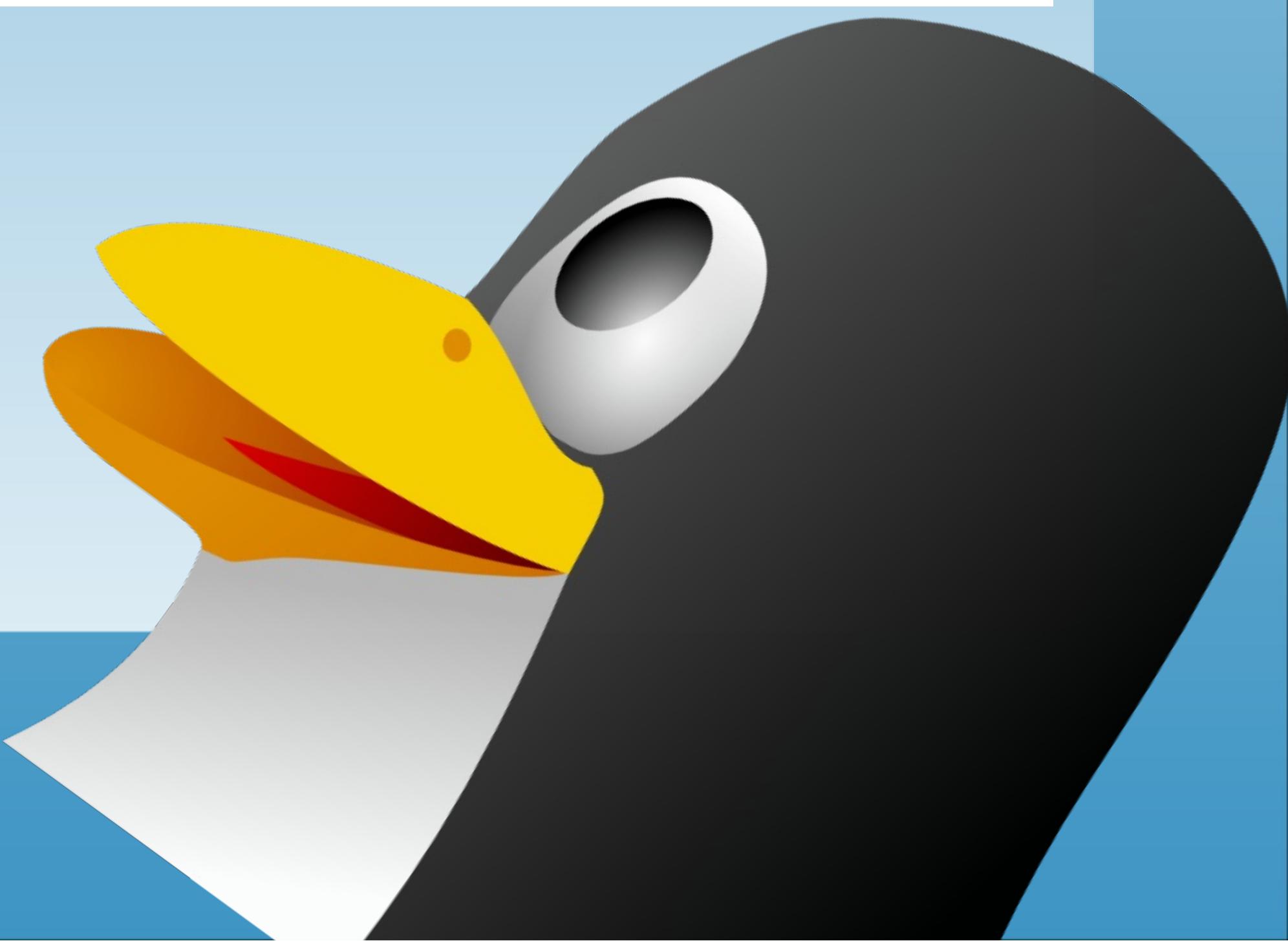
```
polynomial p1(x^3 - y^3);
```

Não compila

```
polynomial p2((x^3) - (y^3));
```

```
polynomial p3(x.pow(3) - y.pow(3));
```

Esses sim



# Só mais uns exemplos...

```
monomial x('x'), y('y');  
polynomial p1(x^3, -y^3);  
polynomial p2(x - y);
```

```
cout << p1 / p2 << endl;
```

$$x^2 + x^*y + y^2$$



# O método de divisão importa!

$$u = 2x^2y + 3x^2 + 4xy + 5x + 6y + 7$$

$$v = xy$$

$$u = (2x)^*v + (3x^2 + 4xy + 5x + 6y + 7)$$

$$u = (2x + 4)^*v + (3x^2 + 5x + 6y + 7)$$

u / v

```
monomial_based_div(u, v);  
lc_based_div(u, v);
```



```
u = (z + y)*(y - x)*(x + y);  
v = (y + z)*(x - y);  
cout << p_gcd(u, v) << endl;
```

$$x^*y + y^2 + x^*z + y^*z$$

```
u = 24;  
v = 16;  
p_gcd(u, v);  
// 1  
p_gcd(u, v, NO_UNIT);  
// 8
```

```

template<class T>
T g_pow(const T &a, const num_z &N){
    T res;

    if(N == 0) return T(1);
    if(N == 1) return a;
    div_tuple q = N / 2;
    res = g_pow(a, q.q);
    if(q.r == 1) return a * res * res;
    return res * res;
}

```

```

cout << g_pow(2 + x, 3) << endl;

```

$$x^3 + 6x^2 + 12x + 8$$



# Obrigado!

<https://github.com/ArthurCarmo/mini-cas>

Contato: [arthur.carmo@ufv.br](mailto:arthur.carmo@ufv.br)

<https://github.com/ArthurCarmo/mini-cas/blob/master/minicas-beta.tar.gz>



# Referências

[livrozilla.com/doc/1343906/ordenação-de-monômios--divisão-em-anéis-de-polinômios-de](http://livrozilla.com/doc/1343906/ordenação-de-monômios--divisão-em-anéis-de-polinômios-de)

