

mini-cas



Arthur Gonçalves do Carmo

Luiz Carlos de Abreu Albuquerque

O que foi feito

- Classe para números inteiros
- Classe para números modulares
- Classe para números racionais

```
#ifndef __SIGNATURES__  
#define __SIGNATURES__  
  
class num_z;  
class num_q;  
struct div_tuple;  
struct mod_tuple;  
  
template<int64_t N>  
class num_zm;  
  
#endif
```



```

#include <stdio.h>
#include <gmp.h>

int main(){
    unsigned int i;
    mpz_t facto[1001];
    mpz_init_set_ui(facto[0], 1);
    mpz_init_set_ui(facto[1], 1);

    for(i = 2; i < 1001; i++){
        mpz_init(facto[i]);
        mpz_mul_ui(facto[i], facto[i-1], i);
    }

    while(scanf("%u",&i) == 1)
        gmp_printf("%u!\n%Zd\n", i, facto[i]);

    return 0;
}

```

```

#!/usr/bin/python
import sys;

facto = [1, 1];
for i in range(2, 1001):
    facto.append(facto[i-1] * i)

for line in sys.stdin:
    print str(int(line)) + "!"
    print facto[int(line)]

```

```

#include <mini-cas.h>

int main(){
    int i;
    num_z facto[1001];

    facto[0] = facto[1] = 1;

    for(i = 2; i < 1001; i++)
        facto[i] = facto[i-1] * i;

    while(std::cin >> i)
        std::cout << i << "!\n" << facto[i] << "\n";

    return 0;
}

```



```
arthur@arthur-System-Product-Name:~/Documents/exercicios/uva$ time ./uva623_py
```

```
real    0m0.066s  
user    0m0.066s  
sys     0m0.000s
```

```
arthur@arthur-System-Product-Name:~/Documents/exercicios/uva$ time ./uva623_gm
```

```
real    0m0.021s  
user    0m0.017s  
sys     0m0.004s
```

```
arthur@arthur-System-Product-Name:~/Documents/exercicios/uva$ time ./uva623_mi
```

```
real    0m0.030s  
user    0m0.029s  
sys     0m0.000s
```



O que será feito

- Classe para polinômios
- Classe para expressões
 - Parte simbólica
- Classe para funções (talvez)



Obrigado !

www.github.com/ArthurCarmo/mini-cas

