

Professor: Vinícius Mota

ATENÇÃO: Para a devida entrega desse exercício, pedimos que você disponibilize o link do seu código do replit, no ava. (Caso queiram correções com comentários em seus códigos do replit adicionem o seguinte email como convidado: abraao.santos@edu.ufes.br)

Para esta tarefa, você deve implementar um tipo de dados que suporte adição e multiplicação de grandes números inteiros não negativos. Para isto, o tipo `numero` deve ser um vetor de caracteres que suporte uma quantidade que julgue necessária de dígitos.

O arquivo `num.h` define uma interface para o tipo de dados. Seu trabalho é fornecer um arquivo `num.c` correspondente que implemente essas funções. Você também pode implementar qualquer outra função que seja útil.

Por segurança, funções que não devem ser acessadas por outros arquivos devem ser declaradas como estáticas.

Um **Num** representa um número inteiro não negativo possivelmente muito grande e pode ser inicializado fornecendo uma string terminada em nulo de dígitos ASCII para a função *numCreate*. Você precisará escolher uma representação apropriada para Nums que permita uma implementação razoavelmente eficiente das funções restantes.

Você pode assumir que um número terá no máximo  $2^{31} - 1$  dígitos decimais.

Seu programa deve receber como entrada 1 caractere indicando a operação (a - adicionar, m - multiplicar, e - elevar) e retorna como saída o resultado da operação. Exemplos:

**Entrada**

```
a
123 456
-----
m
1234456 767329123
-----
e
10000 10
```

## Saída

[illegible]