

Leonardo Jae Yong Noh RA:11202230594

Rodrigo Bomfim Carvalho de Almeida RA: 11202230253

Victor Martim Nascimento RA:11202230763

## 1- Organização do código

### 1. Declaração de Variáveis

- x e y: Ponteiros para armazenar as strings que representam os números grandes digitados.
- tamanhoX e tamanhoY: Variáveis que armazenam o tamanho alocado para as strings x e y. São usadas pela função getline.

### 2. Loop Principal

- vai parar se todo arquivo for lido ou um erro for detectado

### 3. Leitura de ambos os números

- getline: Lê uma linha do padrão de entrada (stdin) e armazena o conteúdo no ponteiro x.
- x[strlen(x) - 1] = '\0': Remove o caractere \n (nova linha) no final da string lida, substituindo-o por \0 (fim de string).

### 4. Leitura do Operador

- scanf("%c", &signal): Lê o operador (+, -, \*, ou /) ignorando espaços em branco anteriores (o espaço antes de %c é importante).
- == EOF: Verifica se o fim da entrada foi alcançado.
- getchar(): Remove o caractere de nova linha (\n) que ficou no buffer após o scanf.

### 5. Conversão das Strings para BigNumber

### 6. Liberação de Memória das Strings

### 7. Realização da Operação

### 8. Impressão do Resultado e Liberação da memória

### 9. Reinicialização de Variáveis

## 2- Funcionalidades implementadas

### *Função Soma(BigNumber a, BigNumber b)*

Cria um BigNumber vazio para armazenar o total e começa a soma pelos dígitos menos significativos (tail) de ambos os números. A cada soma de dígitos, verifica se o valor é maior ou igual a 10; se for, utiliza uma retenção que transfere o valor excedente para a próxima casa decimal. O resultado de cada soma é adicionado no início da lista do BigNumber total. Após percorrer todas as casas, ajusta o sinal, se necessário, e remove zeros à esquerda.

### *Função Subtrai(BigNumber a, BigNumber b)*

Cria um BigNumber vazio para o resultado e determina qual número é maior entre os dois. A subtração é feita dígito a dígito, do tail ao início, considerando três casos: se o dígito do subtraendo é nulo, apenas o minuendo importa; se o dígito do minuendo é maior ou igual ao do subtraendo, realiza-se a subtração direta; caso contrário, aplica-se o sistema de "empréstimo", subtraindo de um dígito não nulo na sequência e ajustando os valores intermediários. No final, ajusta o sinal e remove zeros à esquerda.

### *Função Divide(BigNumber a, BigNumber b)*

Inicialmente, cria um BigNumber para o quociente e verifica condições de erro, como divisor igual a zero ou maior que o dividendo. Otimiza o cálculo buscando um múltiplo próximo do numerador utilizando os valores auxiliares i e j. A partir desse ponto inicial, incrementa o quociente para cada subtração válida do divisor ao numerador até que o divisor não possa mais ser subtraído. Assim, o quociente final é obtido.

### *Função Multiplica(BigNumber a, BigNumber b)*

Cria um BigNumber vazio para o total e processa cada dígito de b (do tail ao início). Para cada dígito de b diferente de zero, cria um BigNumber para a soma parcial, somando o valor de a repetidamente conforme o valor do dígito atual. Adiciona zeros ao final da soma parcial de acordo com a posição decimal do dígito em b. Após somar ao total, reinicia a soma parcial e avança para o próximo dígito de b. Finaliza ajustando o sinal e removendo zeros à esquerda.

## **3- Funcionalidades extras**

### **Função Resto**

Retorna NULL se b for igual a 0, pois não é possível dividir por 0. Cria um BigNumber para o resto e inicializa com o valor de a. Utiliza a função que compara a relação entre a e b: se b for maior, o valor de a é mantido como resto. Caso contrário, executa um loop que subtrai b de resto repetidamente até que b seja maior que o valor restante. No final, ajusta o sinal do resto para ser igual ao de a e remove zeros à esquerda.

## **4- Interface pública do BigNumber**

Estrutura do Nó da Lista Duplamente Encadeada

- Um int que armazena o valor do dígito.

- Um ponteiro para o próximo nó (prox).

- Um ponteiro para o nó anterior (ant).

Estrutura do BigNumber

Uma lista duplamente encadeada que possui:

- Ponteiro para o início da lista (head).

- Ponteiro para o final da lista (tail).

- Tamanho do número (tamanho).

- Indicação se é negativo (eh\_negativo).

criar\_BigNumber

Inicializa um BigNumber vazio: head e tail apontam para NULL, e o tamanho é 0.

adiciona\_no\_final

Adiciona um nó ao final da lista.

adiciona\_no\_inicio

Adiciona um nó no início da lista.

string\_para\_BigNumber

Converte uma string em um BigNumber, criando nós para cada dígito com adiciona\_no\_final.

Remove\_zero

Remove os zeros à esquerda do número.

maiorInt

Compara dois inteiros e retorna o maior.

maiorBigNumber  
Compara dois BigNumbers e retorna o maior deles.

soma\_BigNumber  
Realiza a soma de dois BigNumbers.

subtrai\_BigNumber  
Realiza a subtração de dois BigNumbers.

multiplica\_BigNumber  
Realiza a multiplicação de dois BigNumbers.

divide\_BigNumber  
Realiza a divisão de dois BigNumbers.

resto\_BigNumber  
Calcula o resto da divisão entre dois BigNumbers.

multiplica\_BigNumber\_inteiro  
Multiplica um BigNumber por um inteiro.

printBigNumber  
Imprime os valores armazenados no BigNumber.

alterasinal  
Altera o sinal dos resultados, se necessário.

liberar\_BigNumber  
Libera a memória ocupada por um BigNumber.

operacao  
Recebe dois BigNumbers e um operador (+, -, \*, /, %) e chama a função correspondente para realizar a operação.

## 5- Divisão entre membros do grupo

Leonardo

- Structs BigNumber e nó da lista duplamente encadeada
- Soma

Rodrigo

- Multiplicação
- Resto
- Client.c

Victor

- Subtração
- Divisão
- Operação e funções de sinal