RELATÓRIO – BigInt

Integrantes: Daniel Brito e Alimpio Brito

1 – Descrição do trabalho

Desenvolvimento de calculadora com capacidade de armazenar e realizar operações sobre números de tamanho qualquer utilizando estruturas de dados.

2 – Divisão do trabalho

A maioria das funções foram implementadas em reuniões, no entanto, em algumas outras, um membro da equipe surgia com a lógica por meio de uma explicação enquanto o outro implementava. Logo após, ambos testavam o programa à procura de bugs e sugestões de otimização.

3 – Resolução do problema

I - Definimos as nossas estruturas para manipulação dos dados, bem com os TADs, onde criamos um para manipular os números e outro para realizar as operações.

II - Desenvolvemos a função que recebe um número pelo usuário e o armazena em uma lista duplamente encadeada, onde cada dígito (char) do número é armazenado em um nó.

III - Implementamos a função de Adição.

IV - Implementamos a função de Comparação.

V - Implementamos a função de Multiplicação.

VI - Uma vez implementada a Multiplicação, a utilizamos para compor a função Quadrado.

VII - Implementamos uma função de Subtração para nos auxiliar no desenvolvimento da Fatorial.

VIII - Com auxílio da função de Multiplicação e Subtração, implementamos a função de Exponenciação.

IX - Utilizando a função de Comparação, implementamos a função de Módulo.

X - A função de Módulo, por sua vez, foi primordial para implementação da função de MDC.

XI - No final, definimos uma interface com menu para facilitar o uso do programa pelo usuário.

4 – Dificuldades

Uma das maiores dificuldades encontradas foi definir quais as estruturas seriam utilizadas para manipulação dos dados. Por uma questão de performance, optamos por utilizar duas estruturas, uma para a lista, que representa os números digitados pelo usuário, onde casa dígito ocupa um nó; e outra para referenciar o primeiro e o último dígito dos mesmos.

Durante a implementação das primeiras funções, encontramos muitos bugs em relação a como os dados eram inseridos, os quais acarretavam em resultados inesperados após a realização das operações. Portanto, decidimos fazer um tratamento dos mesmos para evitar transtornos. A função *removeZeroInicio* é um exemplo.

Enfrentamos muitos *segmentations fault.* No entanto, com a ajuda do gdb, conseguimos identificar onde estava o problema. Os erros de semântica, por outro lado, foram os mais difíceis de tratar, pois sabíamos que o resultado estava incorreto, mas não sabíamos onde estava o erro. Neste caso, debugávamos o código no papel ou fazíamos algumas impressões na tela até encontrar e solucionar o problema.

5 – Manual

Ao iniciar o programa, selecione a operação que deseja realizar. Logo após, entre com os valores solicitados. Feito isso, será exibido o resultado juntamente com uma mensagem onde pergunta-se o que fazer a seguir: 1 - Retornar (ao menu principal) ou 0 - Terminar o programa.

6 – Log de testes

```
########################################
            SOMA
########################################

Digite a primeira parcela: 78
Digite a segunda parcela: 0

Resultado = 78

########################################
            SOMA
########################################

Digite a primeira parcela: 5
Digite a segunda parcela: 5

Resultado = 10

########################################
```

SOMA
##

Digite a primeira parcela: 999
Digite a segunda parcela: 1

Resultado = 1000

##
SOMA
##

Digite a primeira parcela: 000000000009
Digite a segunda parcela: 1

Resultado = 10

##
SOMA
##

Digite a primeira parcela: 9998
Digite a segunda parcela: 3

Resultado = 10001

##
SOMA
##

Digite a primeira parcela: 000109
Digite a segunda parcela: 01000

Resultado = 1109

##
SOMA
##

Digite a primeira parcela: 8
Digite a segunda parcela: 00002

Resultado = 10

##
COMPARAÇÃO

##

Digite o primeiro número: 1
Digite o segundo número: 2

O primeiro número é menor que o segundo.

##
COMPARAÇÃO
##

Digite o primeiro número: 00009
Digite o segundo número: 008

O primeiro número é maior que o segundo.

##
COMPARAÇÃO
##

Digite o primeiro número: 9998
Digite o segundo número: 0099998

O primeiro número é menor que o segundo.

##
COMPARAÇÃO
##

Digite o primeiro número: 00101
Digite o segundo número: 101

Os números são iguais.

##
QUADRADO
##

Digite um número: 0

Resultado = 0

##
QUADRADO
##

Digite um número: 9

Resultado = 81

##
QUADRADO
##

Digite um número: 999

Resultado = 998001

##
QUADRADO
##

Digite um número: 00010

Resultado = 100

##
MULTIPLICAÇÃO
##

Digite o primeiro fator: 5
Digite o segundo fator: 0

Resultado = 0

##
MULTIPLICAÇÃO
##

Digite o primeiro fator: 0
Digite o segundo fator: 7

Resultado = 0

##
MULTIPLICAÇÃO
##

Digite o primeiro fator: 99
Digite o segundo fator: 1

Resultado = 99

##
MULTIPLICAÇÃO
##

Digite o primeiro fator: 99
Digite o segundo fator: 7

Resultado = 693

##
MULTIPLICAÇÃO
##

Digite o primeiro fator: 7
Digite o segundo fator: 10

Resultado = 70

##
MDC
##

Digite o primeiro número: 48
Digite o segundo número: 18

Resultado = 6

##
MDC*
##

Digite o primeiro número: 0
Digite o segundo número: 56

Resultado =

##
MDC*
##

Digite o primeiro número: 999
Digite o segundo número: 0

Resultado =

*Obs.: Como não existe divisao por zero, retornamos NULL.

##
MDC**
##

Digite o primeiro número: 997
Digite o segundo número: 991

Resultado = 1

##
MDC**
##

Digite o primeiro número: 2467
Digite o segundo número: 3769

Resultado = 1

**Obs.: Números primos. Quando acima da casa dos 2000, o computador começa a demorar um pouco para exibir o resultado.

##
MDC
##

Digite o primeiro número: 90
Digite o segundo número: 45

Resultado = 45

##
MDC
##

Digite o primeiro número: 1002
Digite o segundo número: 305

Resultado = 1

##
MDC
##

Digite o primeiro número: 900

Digite o segundo número: 39

Resultado = 3

##
MDC
##

Digite o primeiro número: 8
Digite o segundo número: 97

Resultado = 1

##
MDC
##

Digite o primeiro número: 6
Digite o segundo número: 98

Resultado = 2

##
FATORIAL
##

Digite um número: 0

Resultado = 1

##
FATORIAL
##

Digite um número: 1

Resultado = 1

##
FATORIAL
##

Digite um número: 0005

Resultado = 120

##
FATORIAL
##

Digite um número: 99

Resultado =
933262154439441526816992388562667004907159682643816214685929638952175999932299156089414639761565182862536979208272237582511852109168640000000000000000000000

##
FATORIAL
##

Digite um número: 10

Resultado = 3628800

##
EXPONENCIAÇÃO
##

Digite a base: 7
Digite o expoente: 0

Resultado = 1

##
EXPONENCIAÇÃO
##

Digite a base: 0
Digite o expoente: 0

Resultado = 1

##
EXPONENCIAÇÃO
##

Digite a base: 0
Digite o expoente: 8

Resultado = 0

##

EXPONENCIAÇÃO
##

Digite a base: 2
Digite o expoente: 10

Resultado = 1024

##
EXPONENCIAÇÃO
##

Digite a base: 100
Digite o expoente: 2

Resultado = 10000

##
EXPONENCIAÇÃO
##

Digite a base: 0005
Digite o expoente: 00000002

Resultado = 25

##
EXPONENCIAÇÃO
##

Digite a base: 9
Digite o expoente: 25

Resultado = 717897987691852588770249

##
MÓDULO***
##

Digite o dividendo: 0
Digite o divisor: 0

Resultado =

***Obs.: Como não existe divisao por zero, retornamos NULL.

```
########################################
            MÓDULO
########################################

Digite o dividendo: 0
Digite o divisor: 8

Resultado = 0

########################################
            MÓDULO
########################################

Digite o dividendo: 9
Digite o divisor: 7

Resultado = 2

########################################
            MÓDULO
########################################

Digite o dividendo: 99
Digite o divisor: 50

Resultado = 49

########################################
            MÓDULO
########################################

Digite o dividendo: 100
Digite o divisor: 50

Resultado = 0

########################################
            MÓDULO
########################################

Digite o dividendo: 9999
Digite o divisor: 8888

Resultado = 1111

########################################
```

MÓDULO

##

Digite o dividendo: 555555
Digite o divisor: 9999

Resultado = 5610