

Exercícios da Lista de Máquina NORMA

Alunos:

Matheus Augusto Schiavon Parise – 107115

```
def mult(a1: int, a2: int, a3: int, b1: int, b2: int, b3: int) ->
(int, int, int):
    print("\nIniciando Operação...\n\n"
          + "A1: " + str(a1) + ", A2: " + str(a2) + ", A3: " + str(a3)
+
          + ", B1: " + str(b1) + ", B2: " + str(b2) + ", B3: " + str(b3)
+ "\n")

    continuar = 0
    # vai ser necessario 2 aux para não perder a contagem
    aux1 = 0
    aux2 = 0

    # se a3 ou b3 forem 0 da erro
    if a3 == 0 or b3 == 0:
        raise ValueError('O Divisor a3 ou b3 não pode ser 0.')

    # Caso base
    if a2 == 0 or b2 == 0:
        return (0,0,1)

    # Setando o aux1 = a2 para fazer o loop de valores de aux1 pra
aux2
    # no caso do dividendo
    while continuar == 0:
        aux1+=1
        a2-=1
        if a2 == 0:
            continuar +=1
    continuar -=1

    # Fazendo a multiplicação do Dividendo
    while continuar == 0:
        # Basicamente eu fico alternando a valor de a2 entre
        # aux1 e aux2 enquanto eu somo ele á a2, dessa forma eu
        # não perco a contagem.
        b2-=1
        print("Inicia Ciclo de soma")
        print("A1: " + str(a1) + ", A2: " + str(a2) + ", A3: " +
str(a3) + ", B1: " + str(b1) + ", B2: " + str(
        b2) + ", B3: " + str(b3))

        if aux1 == 0:
            while continuar == 0:
                a2 += 1
                aux2 -= 1
                aux1 += 1
                print("A1: " + str(a1) + ", A2: " + str(a2) + ", A3: "
+ str(a3) + ", B1: " + str(b1) + ", B2: " + str(
                    b2) + ", B3: " + str(b3))
                if aux2 == 0:
                    print("Fecha ciclo de soma\n")
                    continuar += 1
            continuar -= 1
        else:
```

```

        while continuar == 0:
            a2 += 1
            aux1 -= 1
            aux2 += 1
            print("A1: " + str(a1) + ", A2: " + str(a2) + ", A3: "
+ str(a3) + ", B1: " + str(b1) + ", B2: " + str(
            b2) + ", B3: " + str(b3))
            if aux1 == 0:
                print("Fecha ciclo de soma\n")
                continuar += 1
            continuar -= 1
        if b2 == 0:
            continuar+=1
        continuar -= 1
        aux1 = 0
        aux2 = 0

# Setando o aux = a3 para fazer o loop de valores de aux1 pra aux2
# no caso do divisor
while continuar == 0:
    aux1 += 1
    a3 -= 1
    if a3 == 0:
        continuar += 1
    continuar -= 1

# Fazendo a multiplicação do Divisor
while continuar == 0:
    # Basicamente eu fico alternando a valor de a3 entre
    # aux1 e aux2 enquanto eu somo ele á a3, dessa forma eu
    # não perco a contagem.
    b3-=1
    if aux1 == 0:
        while continuar == 0:
            a3 += 1
            aux2 -= 1
            aux1 += 1
            if aux2 == 0:
                continuar += 1
            continuar -= 1
    else:
        while continuar == 0:
            a3 += 1
            aux1 -= 1
            aux2 += 1
            if aux1 == 0:
                continuar += 1
            continuar -= 1
    if b3 == 0:
        continuar+=1

# Se A é positivo
if a1 == 0:
    # Se B é positivo
    if b1 == 0:
        return (0, a2, a3)
    # Se B é negativo
    else:
        return (1, a2, a3)

# Se A é negativo

```

```

else:
    # Se B é positivo
    if b1 == 0:
        return (1, a2, a3)
    # Se B é negativo
    else:
        return (0, a2, a3)

def simplificacao_fracao(a2: int, a3: int) -> (int, int, int):
    c = 0

    # Precisaremos de 3 auxiliares aux2 pra guardar a2
    # aux3 pra a3 e aux pra devolver as contagens para os originais
    aux = 0
    aux2 = 0
    aux3 = 0

    # Caso Base
    if a2 == 0:
        return (0, 1)
    # a Base não pode ser 0
    if a3 == 0:
        raise ValueError('O Divisor a3 ou b3 não pode ser 0.')
    # Armazendo o valor de a2 em aux e aux2
    while c == 0:
        a2 -= 1
        aux += 1
        aux2 += 1

        if a2 == 0:
            c += 1
    c -= 1

    # Devolvendo para a2 o valor atraves de aux
    while c == 0:
        a2 += 1
        aux -= 1

        if aux == 0:
            c += 1
    c -= 1

    # Armazendo o valor de a3 em aux e aux3
    while c == 0:
        a3 -= 1
        aux += 1
        aux3 += 1

        if a3 == 0:
            c += 1
    c -= 1

    # Devolvendo para a3 o valor atraves de aux
    while c == 0:
        a3 += 1
        aux -= 1

        if aux == 0:
            c += 1
    c -= 1

```

```

# fazendo o MDC com aux2 e aux3, no fim aux2 terá o MDC
while c == 0:
    x1, x2, x3 = divisao(aux2, aux3)
    aux2, aux3 = aux3, x3
    if aux3 == 0:
        c+=1

a2 = divisao(a2, aux2)
a3 = divisao(a3, aux2)
# Não vai sobrar resto pois o MDC minimo é 1
return (a2[0], a3[0])

def divisao(a2: int, a3: int) -> (int, int, int):
    continuar = 0
    # vai ser necessario 2 aux para não perder a contagem
    aux1 = 0
    aux2 = 0
    sobra = 0

    # Setando o aux1 = a2 para salvar o valor do dividendo
    while continuar == 0:
        aux1 += 1
        a2 -= 1
        if a2 == 0:
            continuar += 1
    continuar -= 1

    # fazendo a divisão
    while continuar == 0:
        if a3 == 0:
            while continuar == 0:
                if aux1 == 0:
                    continuar += 1
                else:
                    aux1 -= 1
                    sobra += 1
                    aux2 -= 1
                    a3 += 1
                    if aux2 == 0:
                        sobra = 0
                        a2 += 1
                        continuar += 1
            continuar -= 1
            if aux1 == 0:
                continuar += 1
        else:
            while continuar == 0:
                if aux1 == 0:
                    continuar += 1
                else:
                    aux1 -= 1
                    aux2 += 1
                    sobra += 1
                    a3 -= 1
                    if a3 == 0:
                        sobra = 0
                        a2 += 1
                        continuar += 1
            continuar -= 1
            if aux1 == 0:
                continuar += 1

```

```

    # Dividendo \ Divisor \ Sobra
    return (a2, 1, sobra)

def soma(a1: int, a2: int, a3: int, b1: int, b2: int, b3: int) ->
(int, int, int):
    print("\nIniciando Operação...\n\n")
    +"A1: "+str(a1)+"", A2: "+str(a2)+"", A3: "+str(a3)
    +", B1: "+str(b1)+"", B2: "+str(b2)+"", B3: "+str(b3)

    # Variaveis
    c = 0
    aux1 = 0
    aux2 = 0
    maior = 0
    iguais = 0
    iguais += 1 # Vamos assumir que são iguais

    # 'Caso Base' das bases pois a base não pode ser 0
    if a3 == 0 or b3 == 0:
        raise ValueError('O Divisor a3 ou b3 não pode ser 0.')

    # Ver se as bases são iguais
    while c == 0:
        aux1 += 1
        a3 -= 1
        b3 -= 1
        if a3 == 0 and b3 == 0:
            # Desfaz as alterações nas base
            while c == 0:
                a3 += 1
                b3 += 1
                aux1 -= 1
                if aux1 == 0:
                    c += 1
            else:
                if a3 == 0:
                    # Soma o resto de b3 em 'maior' usando aux2 pois se a3
    == 0, então b3>a3
                    while c == 0:
                        aux2 += 1
                        b3 -=1
                        maior += 1
                        if b3 == 0:
                            c +=1
                    c -= 1

                    # Restaura b3 e reseta aux2
                    while c == 0:
                        aux2 -= 1
                        b3+=1
                        if aux2 == 0:
                            c+=1
                    c -= 1

                    # desfaz as alterações na base feitas pela verificação
e incrementa a diff em maior
                    while c == 0:
                        aux1 -= 1
                        a3 += 1
                        b3 += 1
                        maior += 1

```

```

        if aux1 == 0:
            c += 1
        # AQUI a3 e b3 são iguais aos originais e maior é o
maior numero
        iguais -= 1
    if b3 == 0:
        # Soma o resto de a3 em 'maior' usando aux2 pois se b3
== 0, então a3>b3
        while c == 0:
            aux2 += 1
            a3 -= 1
            maior += 1
            if a3 == 0:
                c += 1
        c -= 1

        # Restaura a3 e reseta aux2
        while c == 0:
            aux2 -= 1
            a3 += 1
            if aux2 == 0:
                c += 1
        c -= 1

        # desfaz as alterações na base feitas pela verificação
e incrementa a diff em maior
        while c == 0:
            aux1 -= 1
            a3 += 1
            b3 += 1
            maior += 1
            if aux1 == 0:
                c += 1
        # AQUI a3 e b3 são iguais aos originais e maior é o
maior numero
        iguais -= 1
    c-=1

# Se for diferente
if iguais == 0:
    # Usando MMC para equalizar as bases
    SobreA3 = 0
    SobreB3 = 0
    MultA3 = 0
    MultB3 = 0
    while c == 0:
        MultA3, nda1, SobreA3 = divisao(maior, a3)
        MultB3, nda1, SobreB3 = divisao(maior, b3)
        if SobreA3 == 0 and SobreB3 == 0:
            print("Maior (Que agora é o r do MMC): "+str(maior))
            c += 1
        else:
            maior += 1
    c -= 1
    a1, a2, a3 = mult(a1, a2, a3, 0, MultA3, MultA3)
    b1, b2, b3 = mult(b1, b2, b3, 0, MultB3, MultB3)

# Após a resolução dos conflitos de Base
# Executaremos a soma dos dividendos.

```

```

# Se A é positivo
if a1 == 0:
    # Se B é positivo
    if b1 == 0:
        if a2 == 0:
            print("A1: "+str(a1)+", A2: "+str(a2)+ ", A3: "+
str(a3) +", B1: "+str(b1)+", B2: "+str(b2)+", B3: "+str(b3)+"\n")
            # pra mão retornar -0
            if b2 == 0:
                b1 == 0
            return (b1, b2, b3)
        while c == 0:
            if b2 == 0:
                print("")
                # pra mão retornar -0
                if b2 == 0:
                    b1 == 0
                return (a1, a2, a3)
            a2+=1
            b2-=1
            print("A1: "+str(a1)+", A2: "+str(a2)+ ", A3: "+
str(a3) +", B1: "+str(b1)+", B2: "+str(b2)+", B3: "+str(b3))
        # Se B é negativo
    else:
        while c == 0:
            if a2 == 0:
                print("")
                # pra mão retornar -0
                if b2 == 0:
                    b1 == 0
                return (b1, b2, b3)
            if b2 == 0:
                print("")
                # pra mão retornar -0
                if a2 == 0:
                    a1 == 0
                return (a1, a2, a3)
            a2 -= 1
            b2 -= 1
            print("A1: "+str(a1)+", A2: "+str(a2)+ ", A3: "+
str(a3) +", B1: "+str(b1)+", B2: "+str(b2)+", B3: "+str(b3))
        # Se A é negativo
    else:
        # Se B é positivo
        if b1 == 0:
            while c == 0:
                if a2 == 0:
                    print("")
                    # pra mão retornar -0
                    if b2 == 0:
                        b1 == 0
                    return (b1, b2, b3)
                if b2 == 0:
                    print("")
                    # pra mão retornar -0
                    if a2 == 0:
                        a1 == 0
                    return (a1, a2, a3)
                a2 -= 1
                b2 -= 1
            print("A1: "+str(a1)+", A2: "+str(a2)+ ", A3: "+

```

```

str(a3) +", B1: "+str(b1)+", B2: "+str(b2)+", B3: "+str(b3))
# Se B é negativo
else:
    if a2 == 0:
        print("A1: "+str(a1)+", A2: "+str(a2)+ ", A3: "+
str(a3) +", B1: "+str(b1)+", B2: "+str(b2)+", B3: "+str(b3)+"\n")
        # pra mão retornar -0
        if b2 == 0:
            b1 == 0
        return (b1, b2, b3)
    while c == 0:
        if b2 == 0:
            print("")
            # pra mão retornar -0
            if a2 == 0:
                a1 == 0
            return (a1, a2, a3)
        a2+=1
        b2-=1
        print("A1: "+str(a1)+", A2: "+str(a2)+ ", A3: "+
str(a3) +", B1: "+str(b1)+", B2: "+str(b2)+", B3: "+str(b3))

def main():
    Continuar = True
    while Continuar:
        print("\n[0] > Informação")
        print("[1] > Soma")
        print("[2] > Multiplicação")
        print("[3] > Sair")
        opcao = input("\nEscolha: ")
        if opcao == '0':
            print("\nEsse algoritmo foi feito com o intuito de simular
a maquina de NORMA\n")
            "as unicas operações possíveis são:\n"
            "1- Colocar valor 0\n"
            "2- Somar 1\n"
            "3- Subtrair 1\n"
            "4- Comparar com 0\n"
            "Para repetir varias somas ou subtração o uso de
laços de repetição\n"
            "é permitido, contudo as condições devem testar se o
registrador é 0.\n"
            "Para representar -3 nos registradores da Máquina
NORMA utilizamos dois\n"
            "registradores A1 para o sinal (0 para '+' e 1 para
'-'), A2 para a\n"
            "magnitude do número, e A3 para Divisor do Numero
existem um total de 2 registradores A e B.")
        elif opcao == '1':
            print("Escolha os valores para A1,A2,A3,B1,B2,B3:")
            A1 = int(input("\nA1 (0 para '+' ou 1 para '-'): "))
            A2 = int(input("A2 (Dividendo): "))
            A3 = int(input("A3 (Divisor, não pode ser 0): "))
            B1 = int(input("B1 (0 para '+' ou 1 para '-'): "))
            B2 = int(input("B2 (Dividendo): "))
            B3 = int(input("B3 (Divisor, não pode ser 0): "))
            A1, A2, A3 = soma(A1, A2, A3, B1, B2, B3)
            print("RESULTADO:\nA1: " + str(A1) + ", A2: " + str(A2) +
", A3: " + str(A3))

            sinal = ""

```



```

        if A1 == 0:
            sinal = "+"
        else:
            sinal = "-"

        print("ou seja: " + sinal + str(A2) + "/" + str(A3))

        A2_2, A3_2 = simplificacao_fracao(A2, A3)
        print("\nSIMPLIFICAÇÃO INTEIRA:\nA1: " + str(A1) + ", A2: " + str(A2_2) + ", A3: " + str(A3_2))

        print("ou seja: " + sinal + str(A2_2) + "/" + str(A3_2))

    elif opcao == '2':
        print("Escolha os valores para A1,A2,A3,B1,B2,B3:")
        A1 = int(input("\nA1 (0 para '+' ou 1 para '-'): "))
        A2 = int(input("A2 (Dividendo): "))
        A3 = int(input("A3 (Divisor, não pode ser 0): "))
        B1 = int(input("B1 (0 para '+' ou 1 para '-'): "))
        B2 = int(input("B2 (Dividendo): "))
        B3 = int(input("B3 (Divisor, não pode ser 0): "))
        A1, A2, A3 = mult(A1, A2, A3, B1, B2, B3)
        print("RESULTADO:\nA1: " + str(A1) + ", A2: " + str(A2) + ", A3: " + str(A3))

        sinal = ""
        if A1 == 0:
            sinal = "+"
        else:
            sinal = "-"

        print("ou seja: " + sinal + str(A2) + "/" + str(A3))

        A2_2, A3_2 = simplificacao_fracao(A2, A3)
        print("\nSIMPLIFICAÇÃO INTEIRA:\nA1: " + str(A1) + ", A2: " + str(A2_2) + ", A3: " + str(A3_2))
        print("ou seja: " + sinal + str(A2_2) + "/" + str(A3_2))

    elif opcao == '3':
        print("Chá de sumiço!")
        Continuar = False

    else:
        print("\nEscolha Invalida!!!")

if __name__ == '__main__':
    print("\n(UEM - Ciencias da computação) 6896/01 - COMPUTABILIDADE\n"
          + "///MAQUINA DE NORMA///\n"
          + "- Matheus Augusto Schiavon Parise - 107115\n"
          + "- João Gilberto Pelisson Casagrande - 112684")
    main()

```