

# Calculadora na linguagem C utilizando assembly

```
/*Calculadora ASM
Faculdade Anhanguera
Aluno: Matheus Guedes RA: 1800586981 2ºSemestre */
#include <stdio.h>
#include <locale.h>

int soma();
int sub();
int mult();
float div();

int num1, num2, r;

int main(void)
{
    int opc;
    setlocale(LC_ALL, "");

    printf("\tCALCULADORA ASM\n\n");
    printf("[1] SOMA\t\t\t[2] SUBTRAÇÃO\n[3] MULTIPLICAÇÃO\t[4] DIVISÃO\n\n");
    scanf(" %d", &opc);

    switch(opc)
    {
        case 1:
            system("cls");
            printf("SOMA ASM\n\n");
            printf("Digite o primeiro valor: ");
            scanf(" %d", &num1);
            printf("Digite o segundo valor: ");
            scanf(" %d", &num2);
            printf("Resultado = %d\n", soma());
            break;
        case 2:
            system("cls");
            printf("SUBTRAÇÃO ASM\n\n");
            printf("Digite o primeiro valor: ");
            scanf(" %d", &num1);
            printf("Digite o segundo valor: ");
            scanf(" %d", &num2);
            printf("Resultado = %d\n", sub());
            break;
        case 3:
            system("cls");
            printf("MULTIPLICAÇÃO ASM\n\n");
            printf("Digite o primeiro valor: ");
            scanf(" %d", &num1);
            printf("Digite o segundo valor: ");
            scanf(" %d", &num2);
            printf("Resultado = %d\n", mult());
            break;
        case 4:
            system("cls");
            printf("DIVISÃO ASM\n\n");
            printf("Digite o primeiro valor: ");
            scanf(" %d", &num1);
            printf("Digite o segundo valor: ");
            scanf(" %d", &num2);
            printf("Resultado = %.1f\n", div());
            break;
        default :
            main();
            break;
    }
    return 0;
}
```

```

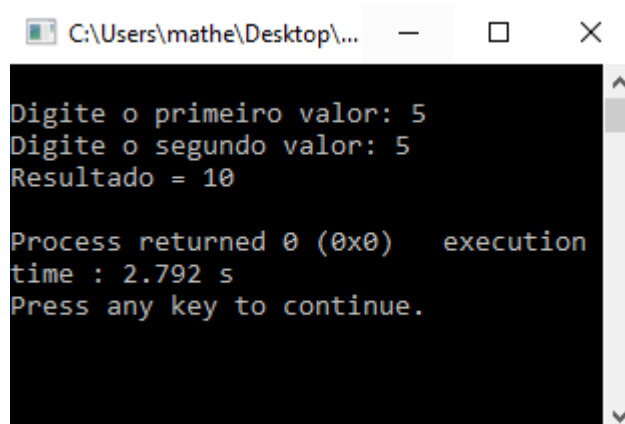
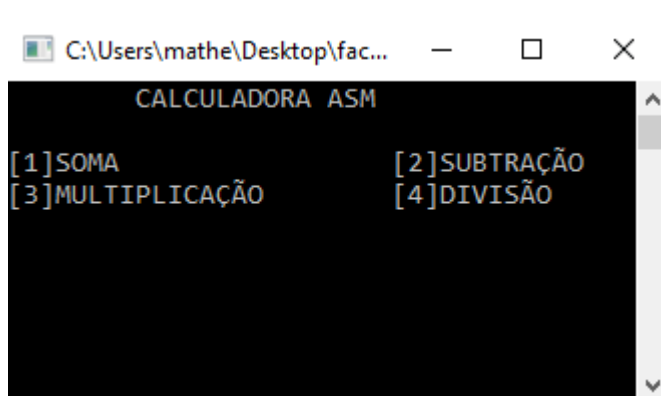
int soma(void)
{
    __asm("movl _num1, %ebx ") ; // Mover o que esta na variavel num1 para o registrador b
    __asm("movl _num2, %eax") ; // Mover o que esta na variavel num2 para o registrador a
    __asm("addl %ebx,%eax"); // Somar o que esta na registrador b no a
    __asm("movl %eax, _r") ; // Mover o que esta no registrador a para a variavel r
    return r; // retornar o q esta na variavel r
}

int sub(void)
{
    __asm("movl _num1, %ebx ") ; // Mover o que esta na variavel num1 para o registrador b
    __asm("movl _num2, %eax") ; // Mover o que esta na variavel num2 para o registrador a
    __asm("sub %eax,%ebx"); // Subtrair o que estã; na registrador a no b
    __asm("movl %ebx, _r") ; // Mover o que esta no registrador b para a variavel r
    return r; // retornar o q esta na variavel r
}

int mult(void)
{
    __asm("movl _num1, %ebx ") ; // Mover o que esta na variavel num1 para o registrador b
    __asm("movl _num2, %eax") ; // Mover o que esta na variavel num2 para o registrador a
    __asm("imull %ebx,%eax"); // Multiplica o registrador bx com o ax e armazena em ax
    __asm("movl %eax, _r") ; // Mover o que esta no registrador a para a variavel r
    return r; // retornar o q esta na variavel r
}

float div(void)
{
    __asm("movl _num1, %eax ") ; // Mover o que esta na variavel num1 para o registrador a
    __asm("movl _num2, %ebx") ; // Mover o que esta na variavel num2 para o registrador b
    __asm("divl %ebx"); // Divide o registrador ax por bx
    __asm("movl %eax, _r") ; // Mover o que esta no registrador a para a variavel r
    return r; // retornar o q esta na variavel r
}

```



Intel Code		AT&T Code	
mov	eax,1	movl	\$1,%eax
mov	ebx,0ffh	movl	\$0xff,%ebx
int	80h	int	\$0x80
mov	ebx, eax	movl	%eax, %ebx
mov	eax,[ecx]	movl	(%ecx),%eax
mov	eax,[ebx+3]	movl	3(%ebx),%eax
mov	eax,[ebx+20h]	movl	0x20(%ebx),%eax
add	eax,[ebx+ecx*2h]	addl	(%ebx,%ecx,0x2),%eax
lea	eax,[ebx+ecx]	leal	(%ebx,%ecx),%eax
sub	eax,[ebx+ecx*4h-20h]	subl	-0x20(%ebx,%ecx,0x4),%eax

A utilização de assembly neste programa é feita para otimizar e melhorar o desempenho na execução das equações pois esta trabalhando diretamente na memoria.

## Fluxograma do programa apresentado na página anterior

