

# Calculadora utilizando funções em linguagem C.

Centro Universitário Avantis –  
UNIAVAN

Professor: Luiz Fernando Arruda

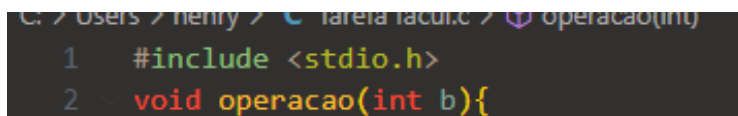
Aluno: Henry Gabriel Carvalho Soares

- **Introdução:**

Nesta atividade, será apresentado o funcionamento e a prática de uma calculadora que utiliza funções na linguagem C para realizar operações matemáticas básicas, como soma, subtração, multiplicação e divisão, assim, teremos uma visão mais aprofundada sobre como os cálculos são processados.

- **Funcionamento:**

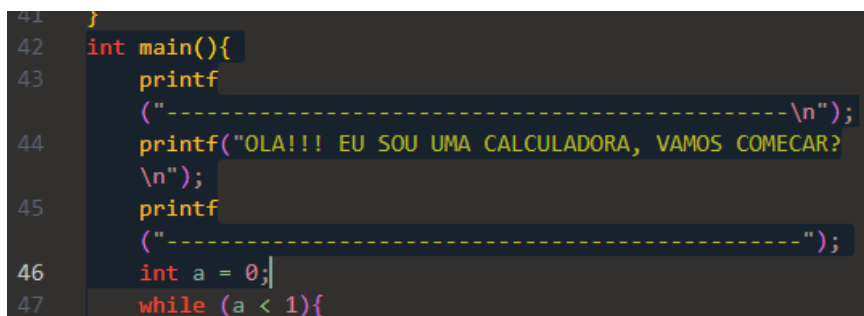
Na primeira linha deve-se incluir a biblioteca `stdio.h` para receber dados e mostrar os resultados da calculadora. A partir da linha 2 começa a função que realiza os cálculos das operações. Conforme a figura 1.



```
C:\Users\henry> C:\ladeira\calculadora> operacao(int)
1  #include <stdio.h>
2  void operacao(int b){
```

Figura 1

Mas para melhor entendimento, deve-se começar a explicação a partir na linha 42, conforme a figura 2, abre a função `"int main()"` para executar o código. Ainda na figura 2, tem a mensagem de apresentação da calculadora.



```
41 }
42 int main(){
43     printf
44     ("-----\n");
45     printf("OLA!!! EU SOU UMA CALCULADORA, VAMOS COMECAR?
46     \n");
47     printf
48     ("-----");
49     int a = 0;
50     while (a < 1){
```

Figura 2

Na linha 46 acrescenta-se o `"int a = 0;"` para somente auxiliar o laço de repetição `while` ser infinita.

Na linha 48, adicionamos a variável `"int b;"` para que será utilizada nas condições relacionadas às opções de operação. Essa variável `b` será atribuída pelo usuário de acordo com a mensagem apresentada.

Na linha 49, exibimos uma mensagem para que o usuário escolha uma das opções de operação: 1 para soma, 2 para subtração, 3 para multiplicação, 4 para divisão e 0 para sair ou encerrar o programa.

Logo abaixo tem o scanf para ler a variável "b" de acordo com a mensagem apresentada. Conforme a figura 3.

```
46     int a = 0;
47     while (a < 1){
48         int b;
49         printf("\nPor favor, digite o numero da
operacao:\n[0]SAIR\n[1]SOMA\n[2]SUBTRACAO\n[3]
MULTIPLICACAO\n[4]DIVISAO\nResposta: ");
50         scanf("%i", &b);
51         if (b == 0){
```

Figura 3

Na figura 4, tem a estrutura de condição com o uso da variável "b" para tomar decisões com base na escolha do usuário.

Se o valor de "b" for 0, é executado o comando "break" permitindo ao usuário sair do programa.

Se o valor de "b" for 1, 2, 3 ou 4, a função "operacao(valor\_b)" é chamada para realizar os cálculos correspondentes à operação escolhida.

Se o valor de "b" for diferente dos mencionados anteriormente, o programa encontra o comando "continue". Isso faz com que o programa ignore o restante do código dentro do laço while e retorne ao início para solicitar uma nova escolha do usuário. Na linha 60 tem a chave de fechamento do laço while e a da 61 da função main.

```
51         if (b == 0){
52             break;
53         }
54         else if (b == 1 || b == 2 || b == 3 ||
b == 4){
55             operacao(b);
56         }
57         else{
58             continue;
59         }
60     }
61 }
```

Figura 4

Finalmente, na figura 5, a função "operacao" tipo void pois a função pode ter resultados inteiros e flutuantes (no caso da divisão) com o parâmetro b. Tem também os valores do tipo long int para a calculadora realizar cálculos com números ainda maiores com as variáveis "x" e "y" ambos atribuídos a valor 0.

```
1 //include <stdio.h>
2 void operacao(int b){
3     long int x = 0;
4     long int y = 0;
5     if (b == 1){
```

Figura 5

Tem aqui, na figura 6, a primeira condição na função: se o b for igual a 1 (conforme a figura 3) será executado a operação de soma.

A "calculadora" vai solicitar ao usuário para inserir valores para atribuir às variáveis "x" e "y" (nos comandos "scanf" utiliza %ld indicando o tipo de variável é long int e "&" para a "variável ler e armazenar esse valor").

E após disso tem o printf exibindo " valor x + valor y = a soma entre eles", a operação se encerra e retorna ao laço while para executar a próxima operação.

E além disso a estrutura para a subtração (figura 7) e multiplicação (figura 8) é semelhante à da soma, com a única diferença sendo o sinal na operação correspondente. O processo de leitura dos valores, cálculo e exibição do resultado é realizado da mesma forma.

```
5     if (b == 1){
6         printf("Digite o primeiro valor: ");
7         scanf("%ld", &x);
8         printf("Digite o segundo valor: ");
9         scanf("%ld", &y);
10        printf("%ld + %ld = %ld", x, y, x+y);
11    }
```

Figura 6

```
12    else if (b == 2){
13        printf("Digite o primeiro valor: ");
14        scanf("%ld", &x);
15        printf("Digite o segundo valor: ");
16        scanf("%ld", &y);
17        printf("%ld - %ld = %ld", x, y, x-y);
18    }
```

Figura 7

```

19     else if (b == 3){
20         printf("Digite o primeiro valor: ");
21         scanf("%ld", &x);
22         printf("Digite o segundo valor: ");
23         scanf("%ld", &y);
24         printf("%ld x %ld = %ld", x, y, x*y);
25     }
26     else{

```

Figura 8

Na divisão, figura 9, a estrutura é um pouco diferente, começando com o "else" pois é a última (e única) alternativa da função.

E após da calculadora solicitar valores, há duas estruturas de condição, a primeira é se o resto da divisão entre "x" e "y" for 0 (ou seja, o resultado da divisão é um número inteiro) ocorre o printf de "x / y = x/y". E a segunda é se o resto não for 0 (ou seja, é um número fracionário) são adicionadas mais duas

```

26     else{
27         printf("Digite o primeiro valor: ");
28         scanf("%ld", &x);
29         printf("Digite o segundo valor: ");
30         scanf("%ld", &y);
31         if (x%y == 0)
32         {
33             printf("%ld / %ld = %ld", x, y, x/y);
34         }
35         else{
36             float z = x;
37             float w = y;
38             printf("%ld / %ld = %f", x, y, z/w);
39         }
40     }
41 }

```

variável ("z" e "w") do tipo float para as variáveis "x" e "y" serem convertidas para float para que a operação de divisão seja mais precisa. Em seguida, a divisão é realizada e exibida entre "z" e "w".

Figura 9

Chegamos ao final da explicação sobre o funcionamento da calculadora. No próximo tópico, será apresentado os testes realizados para verificar o desempenho e a precisão da calculadora.

- Resultado Final:

```
PS C:\Users\henry\output> & .\Tarefa facul.exe
-----
OLA!!! EU SOU UMA CALCULADORA, VAMOS COMECAR?
-----
Por favor, digite o numero da operacao:
[0]SAIR
[1]SOMA
[2]SUBTRACAO
[3]MULTIPLICACAO
[4]DIVISAO
Resposta: 0
PS C:\Users\henry\output>
```

Na figura 10, em conforme com a figura 4, quando o valor 0 foi inserido nas opções de operação, ocorreu o break, ou seja, saiu do programa.

Figura 10

Na figura 11, em conforme com a figura 4 e 6, quando o valor 1 foi inserido realizou a operação de soma entre os valores, e após a realização do mesmo,

```
PS C:\Users\henry\output> & .\Tarefa facul.exe'
-----
OLA!!! EU SOU UMA CALCULADORA, VAMOS COMECAR?
-----
Por favor, digite o numero da operacao:
[0]SAIR
[1]SOMA
[2]SUBTRACAO
[3]MULTIPLICACAO
[4]DIVISAO
Resposta: 1
Digite o primeiro valor: 123
Digite o segundo valor: 456
123 + 456 = 579
Por favor, digite o numero da operacao:
```

ocorreu o loop do while, esperando a próxima operação. E ocorreu de maneira semelhante nas figuras 12 (de subtração), 13 (de multiplicação), 14 e 15 (ambos de divisão).

Figura 11

```
123 + 456 = 579
Por favor, digite o numero da operacao:
[0]SAIR
[1]SOMA
[2]SUBTRACAO
[3]MULTIPLICACAO
[4]DIVISAO
Resposta: 2
Digite o primeiro valor: 123
Digite o segundo valor: 456
123 - 456 = -333
Por favor, digite o numero da operacao:
```

Figura 12

```
123 - 456 = -333
Por favor, digite o numero da operacao:
[0]SAIR
[1]SOMA
[2]SUBTRACAO
[3]MULTIPLICACAO
[4]DIVISAO
Resposta: 3
Digite o primeiro valor: 12
Digite o segundo valor: 34
12 x 34 = 408
Por favor, digite o numero da operacao:
```

Figura 13

```
Por favor, digite o numero da operacao:
[0]SAIR
[1]SOMA
[2]SUBTRACAO
[3]MULTIPLICACAO
[4]DIVISAO
Resposta: 4
Digite o primeiro valor: 90
Digite o segundo valor: 5
90 / 5 = 18
Por favor, digite o numero da operacao:
```

Figura 14

```
Por favor, digite o numero da operacao:
[0]SAIR
[1]SOMA
[2]SUBTRACAO
[3]MULTIPLICACAO
[4]DIVISAO
Resposta: 4
Digite o primeiro valor: 10
Digite o segundo valor: 20
10 / 20 = 0.500000
Por favor, digite o numero da operacao:
```

Figura 15

```
Por favor, digite o numero da operacao:  
[0]SAIR  
[1]SOMA  
[2]SUBTRACAO  
[3]MULTIPLICACAO  
[4]DIVISAO  
Resposta: 6
```

```
Por favor, digite o numero da operacao:  
[0]SAIR  
[1]SOMA  
[2]SUBTRACAO  
[3]MULTIPLICACAO  
[4]DIVISAO  
Resposta: 5
```

```
Por favor, digite o numero da operacao:  
[0]SAIR  
[1]SOMA  
[2]SUBTRACAO  
[3]MULTIPLICACAO  
[4]DIVISAO  
Resposta: 9
```

Na figura 16, em conforme com a figura 4, o comando “continue” é executado pois são valores diferentes mencionados anteriormente, e por isso, ignora o resto do programa e recomeça o loop while até que os valores 0, 1, 2, 3 ou 4 seja inserido.

Figura 16