



PUC Minas

# PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

## CAMPUS EM POÇOS DE CALDAS

Claudio Correa  
correa@pucpcaldas.br

Alterações de adequação na calculador para as quatro operações aritméticas básicas

### Calculadora para aritmética básica.

```
1 #include <stdio.h>
2 #include <stdlib.h> // exit(0)
3
4 // Variaveis globais
5
6 int main() {
7     float x, y, resultado; // Variaveis locais
8     char op;
9
10    fprintf(stdout, "\n __.:__ Calculadora para as operacoes basicas da matematica __.:__ \n");
11    printf("\nInforme o primeiro numero: ");
12    scanf("%f", &x);
13    getchar();
14    printf("\nInforme o segundo numero: ");
15    scanf("%f", &y);
16    getchar();
17    printf("\n Informe a operacao [+ - / *]: ");
18    // scanf("%c", &op); getchar(); // Forma alternativa com scanf() para obter o operador
19    op = getchar(); // Problema: eh possivel informar qualquer operador
20
21    switch(op) {
22        case '+':
23            resultado = x + y;
24            break;
25        case '-':
26            resultado = x - y;
27            break;
28        case '*':
29            resultado = x * y;
30            break;
31        case '/':
32            if ( y != 0 ) {
33                resultado = x / y;
34            } else {
35                printf("\nDivisao por zero: a operacao nao pode ser realizada!\n");
36                exit(1);
37            }
38            break;
39        default:
40            printf("\nA operacao %c eh invalida!\n", op);
41            exit(1);
42    }
43
44    printf("\nO resultado da operacao %.2f %c %.2f eh: %.3f\n", x, op, y, resultado);
45    getchar(); // Esse getchar() nao eh obrigatorio. Eh utilizado somente para manter a janela aberta
46
47    return(0);
48 }
```

```

1  /* Compilar com: gcc -W -Wall calc1.c -o calc */
2  #include <stdio.h>
3  #include <stdlib.h> // exit();
4
5  int main(int, char **);
6  double soma(float, float);
7  double subt(float, float);
8  double mult(float, float);
9  double divi(float, float);
10 // Variaveis globais
11
12 int main(int argc, char **argv) {
13     char op = 0; // Variaveis locais
14     double result;
15     float nro_a, nro_b;
16
17     fprintf(stdout, "\n __.:__ Calculadora para as operacoes basicas da matematica __.:__ \n");
18
19     printf("\nInforme o primeiro valor: ");
20     scanf("%f", &nro_a);
21     getchar();
22     printf("\nInforme o segundo valor: ");
23     scanf("%f", &nro_b);
24     getchar();
25     printf("\nInforme a operacao [+ - * /]: ");
26     op = getchar();
27
28     switch(op) {
29     case '+':
30         result = soma(nro_a, nro_b);
31         break;
32     case '-':
33         result = subt(nro_a, nro_b);
34         break;
35     case '*':
36         result = mult(nro_a, nro_b);
37         break;
38     case '/':
39         if (nro_b != 0) {
40             result = divi(nro_a, nro_b);
41         } else {
42             printf("\nDivisao por zero: a operacao nao pode ser realizada!\n");
43             exit(0);
44         }
45         break;
46     default:
47         printf("\nA operacao %c eh invalida!\n", op);
48         exit(0);
49     }
50
51     printf("\nO resultado da operacao %.2f %c %.2f = %.3lf\n", nro_a, op, nro_b, result);
52     getchar(); // Esse getchar() nao eh obrigatorio. Eh utilizado somente para manter a janela aberta
53
54     return(0);
55 }
56
57 double soma(float x, float y) {
58     return(x + y);
59 }
60
61 double subt(float x, float y) {
62     return(x - y);
63 }
64
65 double mult(float x, float y) {
66     return(x * y);
67 }
68
69 double divi(float x, float y) {
70     return(x / y);
71 }

```

```

1 // Compilar com: gcc -W -Wall calc1.c -o calc
2 #include <stdio.h>
3 #include <stdlib.h> // exit();
4 #include <ctype.h> // toupper();
5
6 int main(int, char **);
7 double soma(float, float);
8 double subtr(float, float);
9 double mult(float, float);
10 double divi(float, float);
11 unsigned short continuar(void);
12
13 // Variaveis globais
14
15 int main(int argc, char **argv) {
16     char op = 0; // Variaveis locais
17     double result;
18     float nro_a, nro_b;
19
20     fprintf(stdout, "\n __.:__ Calculadora para as operacoes basicas da matematica __.:__ \n");
21
22     while (continuar()) {
23         printf("\nInforme o primeiro valor: ");
24         scanf("%f", &nro_a);
25         getchar();
26         printf("\nInforme o segundo valor: ");
27         scanf("%f", &nro_b);
28         getchar();
29         printf("\nInforme a operacao [+ - * /]: ");
30         op = getchar(); // <-- Possivel inconsistencia: qualquer caracter eh aceito
31
32         switch(op) {
33             case '+':
34                 result = soma(nro_a, nro_b);
35                 break;
36             case '-':
37                 result = subtr(nro_a, nro_b);
38                 break;
39             case '*':
40                 result = mult(nro_a, nro_b);
41                 break;
42             case '/':
43                 if (nro_b != 0) {
44                     result = divi(nro_a, nro_b);
45                 } else {
46                     printf("\nDivisao por zero: a operacao nao pode ser realizada!\n");
47                     exit(0);
48                 }
49                 break;
50             default:
51                 printf("\nA operacao %c eh invalida!\n", op);
52                 exit(0);
53         }
54
55         printf("\n0 resultado da operacao %.2f %c %.2f = %.3lf\n", nro_a, op, nro_b, result);
56         getchar();
57     }
58     return(0);
59 }
60
61 // Funcoes
62 double soma(float x, float y) {
63     return(x + y);
64 }
65
66 double subtr(float x, float y) {
67     return(x - y);
68 }
69
70 double mult(float x, float y) {
71     return(x * y);
72 }
73
74 double divi(float x, float y) {
75     return(x / y);
76 }
77
78 unsigned short continuar(void) {
79     char resp;
80     int count = 3; // Ate 3 tentativas
81
82     while (count) {
83         printf("\nDeseja continuar [s/n]: ");
84         resp = toupper(getchar());
85         getchar(); // Esse getchar() eh necessario para consumir o ENTER que permanece no buffer
86     }

```

```
87     if (resp == 'S') {
88         return(1);
89     }
90     if (resp == 'N') {
91         return(0);
92     }
93     count--;
94 }
95 return(0);
96 }
```

---

```

1 // Compilar com: gcc -W -Wall calc1.c -o calc
2 #include <stdio.h>
3 #include <stdlib.h> // exit();
4 #include <ctype.h> // toupper();
5
6 int main(int, char **);
7 double soma(float, float);
8 double subtr(float, float);
9 double mult(float, float);
10 double divi(float, float);
11 unsigned short continuar(void);
12 // Variaveis globais
13 int main(int argc, char **argv) {
14     char op = 0; // Variaveis locais
15     double result;
16     float nro_a, nro_b;
17
18     fprintf(stdout, "\n __.:__ Calculadora para as operacoes basicas da matematica __.:__ \n");
19     do {
20         printf("\nInforme o primeiro valor: ");
21         scanf("%f", &nro_a);
22         getchar();
23         printf("\nInforme o segundo valor: ");
24         scanf("%f", &nro_b);
25         getchar();
26         printf("\nInforme a operacao [+ - * /]: ");
27         op = getchar(); // <-- Possivel inconsistencia: qualquer caracter eh aceito
28         switch(op) {
29             case '+':
30                 result = soma(nro_a, nro_b);
31                 break;
32             case '-':
33                 result = subtr(nro_a, nro_b);
34                 break;
35             case '*':
36                 result = mult(nro_a, nro_b);
37                 break;
38             case '/':
39                 if (nro_b != 0) {
40                     result = divi(nro_a, nro_b);
41                 } else {
42                     printf("\nDivisao por zero: a operacao nao pode ser realizada!\n");
43                     exit(0); // <-- Acao indesejada
44                 }
45                 break;
46             default:
47                 printf("\nA operacao %c eh invalida!\n", op);
48                 exit(0); // <-- Acao indesejada
49         }
50         printf("\n0 resultado da operacao %.2f %c %.2f = %.3lf\n", nro_a, op, nro_b, result);
51         getchar();
52     } while (continuar());
53     return(0);
54 }
55 // Funcoes
56 double soma(float x, float y) {
57     return(x + y);
58 }
59 double subtr(float x, float y) {
60     return(x - y);
61 }
62 double mult(float x, float y) {
63     return(x * y);
64 }
65 double divi(float x, float y) {
66     return(x / y);
67 }
68
69 unsigned short continuar(void) {
70     char resp;
71     int count = 3; // Ate 3 tentativas
72     while (count) {
73         printf("\nDeseja continuar [s/n]: ");
74         resp = toupper(getchar());
75         getchar(); // Esse getchar() eh necessario para consumir o ENTER que permanece no buffer
76         if (resp == 'S') {
77             return(1);
78         }
79         if (resp == 'N') {
80             return(0);
81         }
82         count--;
83     }
84     return(0);
85 }

```

---

```

1 // Compilar com: gcc -W -Wall calc1.c -o calc
2 #include <stdio.h>
3 #include <stdlib.h> // exit();
4 #include <ctype.h> // toupper();
5
6 int main(int, char **);
7 double soma(float, float);
8 double subtr(float, float);
9 double mult(float, float);
10 double divi(float, float);
11 unsigned short continuar(void);
12 char operador(void);
13
14 // Variaveis globais
15
16 int main(int argc, char **argv) {
17     char op = 0; // Variaveis locais
18     double result;
19     float nro_a, nro_b;
20     unsigned short flag = 1;
21
22     fprintf(stdout, "\n __.:__ Calculadora para as operacoes basicas da matematica __.:__ \n");
23
24     do {
25         printf("\nInforme o primeiro valor: ");
26         scanf("%f", &nro_a);
27         getchar();
28         printf("\nInforme o segundo valor: ");
29         scanf("%f", &nro_b);
30         getchar();
31
32         // printf("\nInforme a operacao [+ - * /]: ");
33         // op = getchar();
34
35         op = operador(); // Funcao elaborada para tratar operadores os validos
36         flag = 1;
37         switch(op) {
38             case '+':
39                 result = soma(nro_a, nro_b);
40                 break;
41             case '-':
42                 result = subtr(nro_a, nro_b);
43                 break;
44             case '*':
45                 result = mult(nro_a, nro_b);
46                 break;
47             case '/':
48                 if (nro_b != 0) {
49                     result = divi(nro_a, nro_b);
50                 } else {
51                     printf("\nDivisao por zero: a operacao nao pode ser realizada!\n");
52                     flag = 0;
53                 }
54                 break;
55             default: // Esse caso (padrao) nunca serah utilizado. Tratamos os operadores validos com a funcao operador()
56                 printf("\nA operacao %c eh invalida!\n", op);
57                 flag = 0;
58                 break;
59         }
60
61         if (flag) {
62             printf("\n0 resultado da operacao %.2f %c %.2f = %.3lf\n\n", nro_a, op, nro_b, result);
63             getchar();
64         }
65
66     } while (continuar());
67
68     return(0);
69 }
70
71 // Funcoes
72 double soma(float x, float y) {
73     return(x + y);
74 }
75
76 double subtr(float x, float y) {
77     return(x - y);
78 }
79
80 double mult(float x, float y) {
81     return(x * y);
82 }
83
84 double divi(float x, float y) {
85     return(x / y);
86 }

```

```

87
88 unsigned short continuar(void) {
89     char resp;
90     int count = 3; // Ateh 3 tentativas
91
92     while (count) {
93         printf("\nDeseja continuar [s/n]: ");
94         resp = toupper(getchar());
95         getchar(); // Esse getchar() eh necessario para consumir o ENTER que permanece no buffer
96
97         if (resp == 'S') {
98             return(1);
99         }
100         if (resp == 'N') {
101             return(0);
102         }
103         count--;
104     }
105     return(0);
106 }
107
108 char operador(void) {
109     char op = 0;
110
111     do {
112         printf("\nInforme a operacao [+ - * /]: ");
113         op = getchar();
114         getchar(); // Esse getchar() eh necessario para consumir o ENTER que permanece no buffer
115
116         switch(op) {
117             case '+':
118             case '-':
119             case '*':
120             case '/':
121                 break;
122             default:
123                 printf("\nA operacao %c eh invalida!\n", op);
124                 op = 0;
125                 break;
126         }
127     } while (op == 0);
128
129     return(op);
130 }

```

---



## Fatorial para um inteiro positivo.

```
1 // Program to find factorial of a number
2 // For a positive integer n, factorial = 1 * 2 * 3 ... n
3
4 #include <stdio.h>
5
6 int main(void) {
7     int number;
8     long long factorial;
9
10    printf("Enter an integer: ");
11    scanf("%d", &number);
12    getchar();
13
14    factorial = 1;
15
16    // loop terminates when number is less than or equal to 0
17    while (number > 0) {
18        factorial *= number; // factorial = factorial * number;
19        --number;
20    }
21
22    printf("\nFactorial= %lld", factorial);
23    return(0);
24 }
```

---

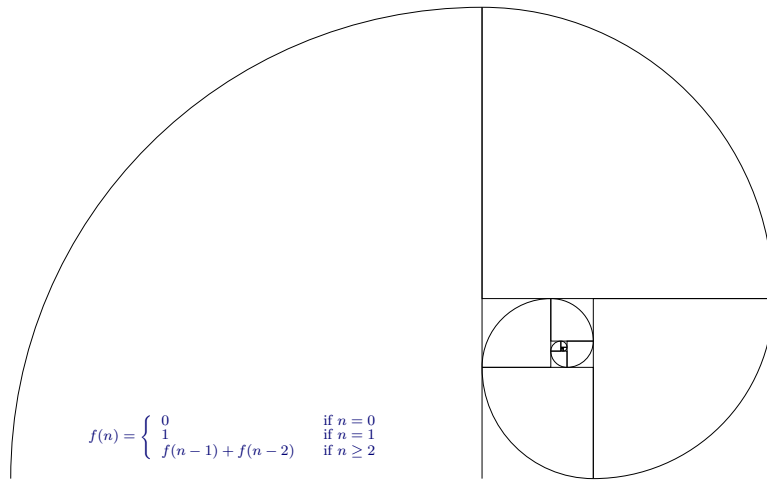
Somatório interrompido informando zero.

```
1 // Program to add numbers until user enters zero
2
3 #include <stdio.h>
4
5 int main(void) {
6     double number, sum = 0;
7
8     // body of loop is executed at least once
9     do {
10         printf("\nEnter a number: ");
11         scanf("%lf", &number);
12         getchar();
13         sum += number;
14     } while (number != 0.0);
15
16     printf("\nSum = %.2lf", sum);
17
18     return(0);
19 }
```

---

```
1 // Sum of natural numbers using for Loop
2 #include <stdio.h>
3
4 int main() {
5     int n, i, sum = 0;
6
7     printf("\nEnter a positive integer: ");
8     scanf("%d", &n);
9     getchar();
10
11     for (i = 1; i <= n; ++i) {
12         sum += i; // sum = sum+i;
13     }
14
15     printf("\nSum = %d", sum);
16     return(0);
17 }
18
19 // Sum of natural numbers using while Loop
20 #include <stdio.h>
21
22 int main() {
23     int n, i, sum = 0;
24
25     printf("\nEnter a positive integer: ");
26     scanf("%d", &n);
27     getchar();
28
29     i = 1;
30     while (i <= n) {
31         sum += i;
32         ++i;
33     }
34
35     printf("\nSum = %d", sum);
36     return(0);
37 }
38
39 // Program to Read Input Until User Enters a Positive Integer
40 #include <stdio.h>
41
42 int main() {
43     int n, i, sum = 0;
44
45     do {
46         printf("\nEnter a positive integer: ");
47         scanf("%d", &n);
48         getchar();
49     } while (n <= 0);
50
51     for (i = 1; i <= n; ++i) {
52         sum += i; // sum = sum + i;
53     }
54
55     printf("\nSum = %d", sum);
56     return(0);
57 }
```

---



0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144

Figura 1: Sequência de Fibonacci (Fibonacci sequence).

## Sequência de Fibonacci

```

1 // You should have the knowledge of following C programming topics:
2 // . Operators
3 // . while and do...while loop
4 // . for loop
5 // . break and continue statement
6
7 // The Fibonacci sequence is a series where the next term is the sum of pervious two terms.
8 // The first two terms of the Fibonacci sequence is 0 followed by 1.
9
10 // The Fibonacci sequence: 0, 1, 1, 2, 3, 5, 8, 13, 21
11 // \href{http://www.mathsisfun.com/numbers/fibonacci-sequence.html}{Fibonacci sequence}
12
13 // Fibonacci Series up to n number of terms
14 #include <stdio.h>
15
16 int main() {
17     int i, n, t1 = 0, t2 = 1, nextTerm;
18
19     printf("\nEnter the number of terms: ");
20     scanf("%d", &n);
21     getchar();
22
23     printf("\nFibonacci Series: ");
24
25     for (i = 1; i <= n; ++i) {
26         printf("%d, ", t1);
27         nextTerm = t1 + t2;
28         t1 = t2;
29         t2 = nextTerm;
30     }
31     return(0);
32 }
33
34 // Fibonacci Sequence Up to a Certain Number
35 #include <stdio.h>
36
37 int main() {
38     int t1 = 0, t2 = 1, nextTerm = 0, n;
39
40     printf("\nEnter a positive number: ");
41     scanf("%d", &n);
42     getchar();
43
44     // Displays the first two terms which is always 0 and 1
45     printf("\nFibonacci Series: %d, %d, ", t1, t2);
46
47     nextTerm = t1 + t2;
48
49     while (nextTerm <= n) {
50         printf("%d, ", nextTerm);
51         t1 = t2;
52         t2 = nextTerm;
53         nextTerm = t1 + t2;
54     }
55     return(0);
56 }

```

```
1 // Sum of Natural Numbers Using Recursion
2
3 #include <stdio.h>
4
5 int addNumbers(int);
6
7 int main() {
8     int num;
9
10    printf("\nEnter a positive integer: ");
11    scanf("%d", &num);
12    getchar();
13
14    printf("\nSum = %d", addNumbers(num));
15
16    return(0);
17 }
18
19 int addNumbers(int n) {
20
21     if (n != 0) {
22         return(n + addNumbers(n - 1));
23     } else {
24         return(n);
25     }
26 }
```

---

*Bom trabalho!*