

Funções

Roberto Rocha



E se o procedimento voltasse valor?

Função

Função é um bloco de programa, assim como são os procedimentos (sub-rotinas), e contem início e fim, sendo identificada por um nome.

Uma função é similar a sub-rotina (procedimento). A diferença está no fato de que função é uma estrutura de abstração de dados, que possui como característica operacional a capacidade de sempre retornar um conteúdo como resposta.

Os procedimentos não retornam conteúdos, a não ser com o uso de passagens de parâmetros por referência.

Funções também operam com passagens de parâmetro por valor e por referência.

```
função <nomefunção> ([var] parâmetros) : tipo  
var  
    <declaração das variáveis locais à função>  
início  
    <lista de comandos>  
    retorne (valor)  
fimfunção
```

Função

Exemplo: Elaborar uma função que retorne o sucessor de um número dado. Em seguida fazer um programa que chame essa função.

```
funcao sucessor(numero: inteiro) : inteiro
  var
    suc:inteiro
  inicio
    suc ← numero +1
    retorne suc
fimfuncao
```

Algoritmo “ex01”

```
var n,s:inteiro
Inicio
  leia(n)
  s←sucessor(n)
  escreva(“O sucessor é = “,s)
finalgoritmo
```

Função

Exemplo: Elaborar uma função que retorne o sucessor de um número dado. Em seguida fazer um programa que chame essa função.

```
→ funcao sucessor(numero:inteiro) :inteiro
    var
    → suc:inteiro
    inicio
    → suc ← numero +1
    → retorne suc
    fimfuncao
```

Algoritmo "ex02"

```
→ var n,s:inteiro
    Inicio
    → leia(n)
    → s ← sucessor(n)
    → escreva("O sucessor é = ",s)
    → finalgoritmo
```

Local	Variável	
Pprincipal	n	5
Pprincipal	s	6
sucessor	numero	5
sucessor	suc	6

O sucessor é = 6

Função que retorne o sucessor de um número dado.

```
funcao sucessor(numero: inteiro) :inteiro
var
    suc:inteiro
inicio
    suc ← numero +1
    retorne suc
fimfuncao
```

Algoritmo “ex02”

```
var n,s:inteiro
Inicio
    leia(n)
    s ← sucessor(n)
    escreva("O sucessor é = ",s)
finalgoritmo
```

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <locale.h>
4  int sucessor(int numero);
5  int main()
6  {
7      setlocale(LC_ALL, "portuguese");
8      int n,s;
9      printf("Digite um valor:");
10     scanf("%d",&n);
11     s=sucessor(n);
12     printf("O sucessor é = %d\n",s);
13     return 0;
14 }
15 int sucessor(int numero)
16 {
17     int suc;
18     suc = numero+1;
19     return suc;
20 }
```

```
Digite um valor:5
O sucessor é = 6
Process returned 0 (0x0)   execution time : 2.567 s
Press any key to continue.
```

Função que retorne o sucessor de um número dado.

```
funcao sucessor(numero:inteiro) :inteiro
var
    suc:inteiro
inicio
    suc ← numero +1
    retorne suc
fimfuncao
```

Algoritmo “ex02”

```
var n,s:inteiro
Inicio
    leia(n)
    s ← sucessor(n)
    escreva("O sucessor é = ",s)
finalgoritmo
```

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <locale.h>
4  int sucessor(int numero);
5  int main()
6  {
7      setlocale(LC_ALL, "portuguese");
8      int n,s;
9      printf("Digite um valor:");
10     scanf("%d",&n);
11     s=sucessor(n);
12     printf("O sucessor é = %d\n",s);
13     return 0;
14 }
15 int sucessor(int numero)
16 {
17     int suc;
18     suc = numero+1;
19     return suc;
20 }
```

Exercício: Faça uma função que receba dois valores e retorne a média entre eles

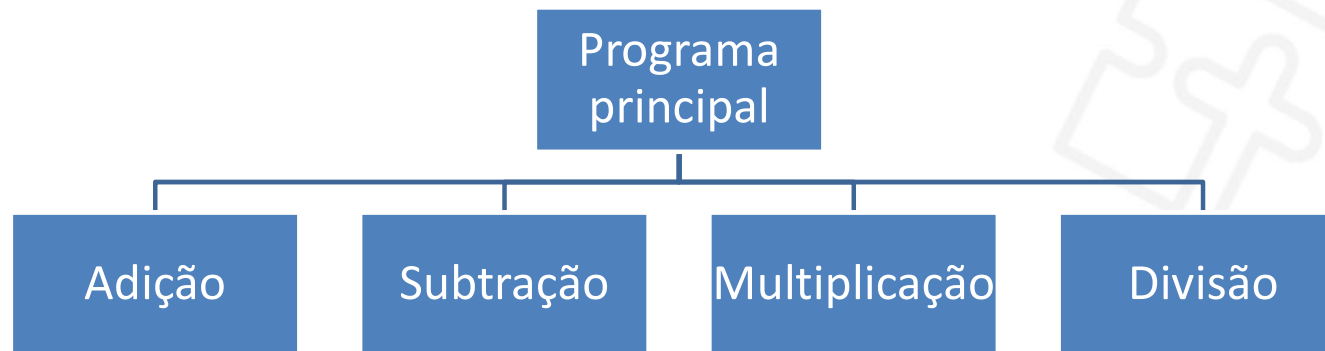
Funções

Exercício de fixação:

Desenvolver um programa de computador que simule as operações básicas de uma calculadora que opere com a entrada de dois valores do tipo real após a escolha da operação a ser executada.

O programa deve apresentar uma lista de opções (menu) com as operações matemáticas de **adição**, **subtração**, **multiplicação** e **divisão**, além de uma opção de saída do programa. Escolhida a opção desejada, deve ser solicitada a entrada de dois valores numéricos - **NO PROGRAMA PRINCIPAL** - para que seja possível executar o processamento escolhido. Após a execução da operação, o **programam principal** deve apresentar o resultado. Após a execução de qualquer uma das operações de calculo, o programa deve voltar para o menu de seleção.

Calculadora



Calculadora

Algoritmo “Calculadora”

var

op:inteiro

a,b:real

Inicio

faça

leValores(a,b)

op = menu()

escolha (op)

1: escreva(somaValores(a,b))

2: escreva(subtraiValores(a,b))

3: escreva(multiplicaValores(a,b))

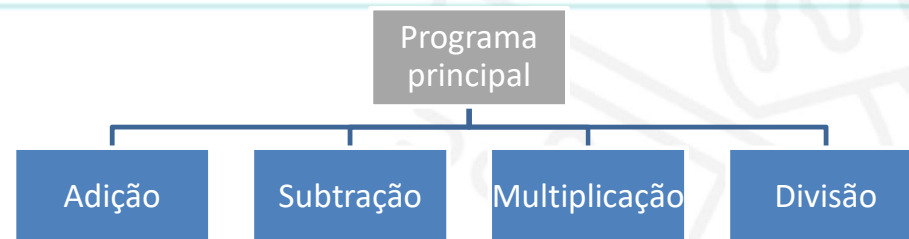
4:escreva(divideValores(a,b))

5:leValores(a,b)

fimescolha

enquanto (op<>9)

fimalgoritmo



funcao menu() : inteiro

var

item:inteiro

inicio

escreva("menu:")

escreva("1 – adição")

escreva("2 – subtração")

escreva("3 – multiplicação")

escreva("4 – divisão")

escreva("5 – novos números")

escreva("9 – sair do programa")

leia(item)

fimprocedimento

procedimento leValores(var a,b :real)

var

inicio

leia(a,b)

fimprocedimento

Calculadora

Algoritmo "Calculadora"

var

op:inteiro

a,b:real

Início

faça

leValores(a,b)

op = menu()

escolha (op)

1: escreva(somaValores(a,b))

2: escreva(subtraiValores(a,b))

3: escreva(multiplicaValores(a,b))

4: escreva(divideValores(a,b))

5: leValores(a,b)

fimescolha

enquanto (op<>9)

finalgoritmo

```
12 int main()
13 {
14     setlocale(LC_ALL, "portuguese");
15     int op;
16     float a,b;
17     leValores(&a,&b);
18     do
19     {
20         op=menu();
21         switch (op)
22         {
23             case 1:
24                 printf("O resultado da operação = %5.2f\n",somaValores(a,b));
25                 break;
26             case 2:
27                 printf("O resultado da operação = %5.2f\n",subtraiValores(a,b));
28                 break;
29             case 3:
30                 printf("O resultado da operação = %5.2f\n",multiplicaValores(a,b));
31                 break;
32             case 4:
33                 printf("O resultado da operação = %5.2f\n",divideValores(a,b));
34                 break;
35             case 5:
36                 leValores(&a,&b);
37                 break;
38         }
39         if (op!=9 && op!=5)
40         {
41             system("PAUSE");
42         }
43     }
44     while(op!=9);
45     return 0;
46 }
```

Calculadora

procedimento leValores(var a,b :real)

var

inicio

leia(a,b)

fimprocedimento

funcao menu() : inteiro

var

item:inteiro

inicio

escreva("menu:")

escreva("1 – adição")

escreva("2 – subtração")

escreva("3 – multiplicação")

escreva("4 – divisão")

escreva("5 – novos números")

escreva("9 – sair do programa")

leia(item)

fimprocedimento

O parâmetro já era
ponteiro!!!

passagem por referência
Utilizar * na frente da
variável

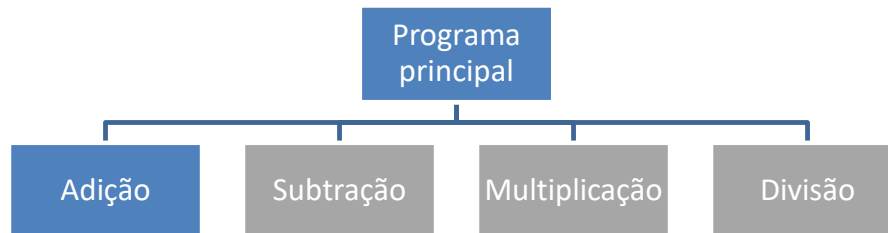
```
44 void leValores(int *a,int *b)
45 { printf("Valores para calcular:\n");
46   printf("Digite o 1 valor:");
47   scanf("%i",a);
48   printf("Digite o 2 valor:");
49   scanf("%f",b);
50 }
51 int menu()
52 {
53   int item;
54   system("CLS");
55   printf("menu:\n");
56   printf("1 - adição\n");
57   printf("2 - subtração\n");
58   printf("3 - multiplicação\n");
59   printf("4 - divisão\n");
60   printf("5 - novos numeros\n");
61   printf("9 - sair do programa\n");
62   printf("Escolha:");
63   scanf("%d",&item);
64   return item;
65 }
```

Por que aqui não
precisou utilizar & na
chamada do scanf?

Tipo da função

Retorno da função

Calculadora - adição

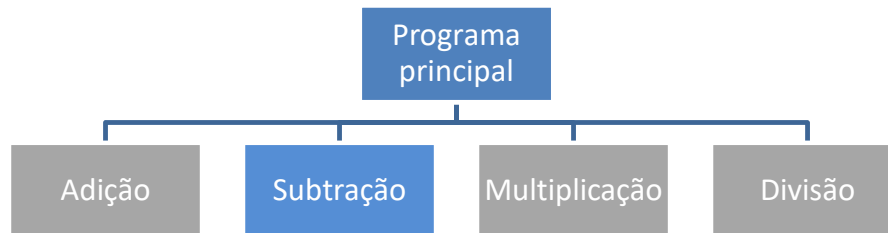


deve ser solicitada a entrada de dois valores numéricos para que seja possível executar o processamento escolhido e mostrar o resultado.

```
funcao somaValores(n1,n2:real):real
var
  soma:real
inicio
  soma ← n1 + n2
  retorne soma
fimprocedimento
```

```
66 float somaValores(float n1,float n2)
67 {
68     float soma;
69     soma =n1 + n2;
70     return soma;
71 }
```

Calculadora - Subtração



deve ser solicitada a entrada de dois valores numéricos para que seja possível executar o processamento escolhido e mostrar o resultado.

funcao subtraiValores(numero1,numero2:real):real

var

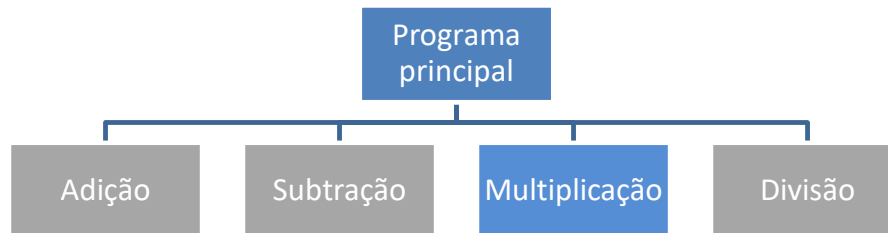
inicio

retorne (numero1 – numero2)

fimprocedimento

```
72 float subtraiValores(float numero1,float numero2)
73 {
74     return (numero1 - numero2);
75 }
```

Calculadora - Multiplicação



deve ser solicitada a entrada de dois valores numéricos para que seja possível executar o processamento escolhido e mostrar o resultado.

funcao multiplicaValores(valor1,valor2:real):real

var

 multiplica: real

inicio

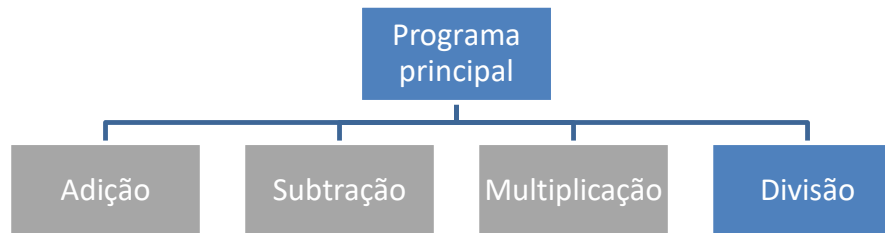
 multiplica \leftarrow valor1 * valor2

 retorne multiplica

fimprocedimento

```
76 float multiplicaValores(float valor1,float valor2)
77 {
78     float multiplica;
79     multiplica= valor1 * valor2;
80     return multiplica;
81 }
```


Calculadora - Divisão



deve ser solicitada a entrada de dois valores numéricos para que seja possível executar o processamento escolhido e mostrar o resultado.

funcao divideValores(a,b:real):real

var

divide : real

inicio

divide \leftarrow a / b

retorne divide

fimprocedimento

```
82 float divideValores(float a, float b)
83 {
84     float divide;
85     divide = (float) a / b;
86     return divide;
87 }
```


Calculadora

```
44 void leValores(int *a,int *b)
45 { printf("Valores para calcular:\n");
46   printf("Digite o 1 valor:");
47   scanf("%f",a);
48   printf("Digite o 2 valor:");
49   scanf("%f",b);
50 }
51 int menu()
52 {
53   int item;
54   system("CLS");
55   printf("menu:\n");
56   printf("1 = adição\n");
57   printf("2 = subtração\n");
58   printf("3 = multiplicação\n");
59   printf("4 = divisão\n");
60   printf("5 = novos numeros\n");
61   printf("9 = sair do programa\n");
62   printf("Escolha:");
63   scanf("%d",&item);
64   return item;
65 }
```

```
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
```

```
void leValores(int *a,int *b);
int menu();
float somaValores(float n1,float n2);
float subtraiValores(float numero1,float numero2);
float multiplicaValores(float valor1,float valor2);
float divideValores(float a,float b);
```

```
66 float somaValores(float n1,float n2)
67 {
68   float soma;
69   soma =n1 + n2;
70   return soma;
71 }
72 float subtraiValores(float numero1,float numero2)
73 {
74   return (numero1 - numero2);
75 }
76 float multiplicaValores(float valor1,float valor2)
77 {
78   float multiplica;
79   multiplica= valor1 * valor2;
80   return multiplica;
81 }
82 float divideValores(float a,float b)
83 {
84   float divide;
85   divide = (float) a / b;
86   return divide;
87 }
88 }
```

```
12 int main()
13 { setlocale(LC_ALL,"portuguese");
14   int op;
15   float a,b;
16   leValores(&a,&b);
17   do
18   { op=menu();
19     switch (op)
20     {case 1:
21       printf("O resultado da operação = %.2f\n",somaValores(a,b));
22       break;
23     case 2:
24       printf("O resultado da operação = %.2f\n",subtraiValores(a,b));
25       break;
26     case 3:
27       printf("O resultado da operação = %.2f\n",multiplicaValores(a,b));
28       break;
29     case 4:
30       printf("O resultado da operação = %.2f\n",divideValores(a,b));
31       break;
32     case 5:
33       leValores(&a,&b);
34       break;
35     }
36     if (op!=9 && op!=5)
37     {
38       system("PAUSE");
39     }
40   }
41   while(op!=9);
42   return 0;
43 }
```

Exercício de fixação

Crie uma função que receba um parâmetro passado por valor e retorne na própria chamada da função o somatório do número 1 até o parâmetro recebido.

Crie um programa principal que solicite a entrada de um valor e imprima o resultado devolvido pela função criada acima.

Exercício de fixação

- 1 - Elabore uma função para calcular e retornar o Fatorial de um número passado por parâmetro. Faça também um programa principal que chame essa função.
- 2 – Elabore uma função que receba um valor passado por parâmetro e retorne o termo correspondente ao valor passado da serie de Fibonacci. Exemplo suponha que seja passado o valor 6, então deverá retorna o 6º termo da série ou seja 1,1,2,3,5,8, ou seja, deverá atribuir a variável Fib o valor 8.
- 3 – Elabore uma função que receba um valor passado por parâmetro e retorne a soma dos termos da serie de Fibonacci do primeiro termo até o termo passado. Exemplo suponha que seja passado o valor 6, então deverá devolver a soma de 1+1+2+3+5+8, ou seja, 20.
- 4 – Elabore uma função que receba dois valores passados por parâmetro e retorne a quantidade de números ímpares entre o intervalo fechado correspondente aos valores passados. Por exemplo, suponha que os valores passados seja, 3 e 6, deverá devolver 2, pois existem os números impares 3 e 5 no intervalo.
- 5 - Elabore uma função que receba dois valores passados por parâmetro e retorne soma dos números ímpares entre o intervalo fechado correspondente aos valores passados. Por exemplo, suponha que os valores passados seja, 3 e 6, deverá devolver 8, pois a soma dos números impares 3 e 5 no intervalo é 8.



PUC Minas
Virtual