

Subalgoritmos com retorno de valor

Aula 22



Subalgoritmos – Retorno de valor

Como permitir que um subalgoritmo **retorne o valor de uma variável** para o local de onde ele foi chamado?

PROBLEMA:

Escreva um subalgoritmo chamado **calculaSerie** que receba como entrada um inteiro que representa a quantidade de termos e retorne o valor de S para a série abaixo.

$$S = 1 + \frac{1}{2} + \frac{1}{3} + \dots$$

Escreva um algoritmo que calcule e imprima o valor de S para 4 termos da série acima. O valor deve ser obtido com a chamada ao subalgoritmo **calculaSerie**.



Subalgoritmos – Retorno de valor

$$S = 1 + \frac{1}{2} + \frac{1}{3} + \dots$$

`float calculaSerie(int n)`

`s ← 0`

`a ← 1; a ≤ n; a++`

`s ← s + 1/a`

retorna s

Algoritmo principal

`v ← calculaSerie(4)`

Escreva v

O valor retornado é
armazenado na variável

```
#include <stdio.h>

float calculaSerie(int n);
```

```
main(){
    float v;
    v = calculaSerie(4);
    printf("Valor: %f\n",v);
}
```

Tipo retornado pela função

```
float calculaSerie(int n){
    int a;
    float s;
```

Comando que retorna o valor

```
s=0;
for (a=1; a<=n; a++)
    s = s + (float) 1/a;
return s;
}
```

Subalgoritmos



```
main(){  
float v;
```

```
v = calculaSerie(4);
```

```
...
```

O argumento **4** é passado para o parâmetro
n declarado na função **calculaSerie**.

calculaSerie (int n)

2.08

s

i

4

n

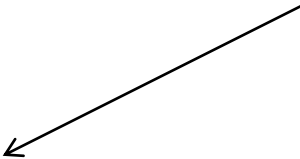
O valor de **s** é retornado para o ponto
onde a função **calculaSerie** foi chamada.

```
#include <stdio.h>
#include <math.h>
float areaCirculo(float raio);
```

```
main(){
    float a,r;

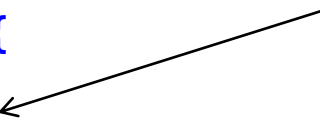
    printf("Informe o raio:");
    scanf("%f", &r);
    printf("Area: %f\n", areaCirculo(r));
}
```

O valor retornado pode ser diretamente enviado como argumento de outra função



```
float areaCirculo(float raio){
    return M_PI * raio * raio;
}
```

O comando **return** pode retornar o resultado de uma expressão.



OBS: Com o comando **return** só é possível retornar **1** valor.



Retorno de valor sem parâmetros de entrada

Uma função pode retornar um valor **sem que haja parâmetro de entrada**.

PROBLEMA:

- a) Escreva uma função chamada **obtemPositivo** que faz a leitura de um número inteiro e o retorna se for um número positivo. Caso seja negativo ou zero a leitura deve ser repetida.

- b) Escreva um algoritmo que leia inteiro N positivo (chamando o subalgoritmo **obtemPositivo**) e calcule o valor da série do exemplo anterior (chamando **calculaSerie**) para o N termos.

$$S = 1 + \frac{1}{2} + \frac{1}{3} + \dots$$

```
#include <stdio.h>

float calculaSerie(int n);
int obtemPositivo(void);

main(){
    float v;
    int n;

    n = obtemPositivo();
    v = calculaSerie(n);
    printf("Valor:%f\n", v);
}
```

```
int obtemPositivo(){
    int num;

    do{
        printf("Informe um valor positivo:");
        scanf("%d", &num);
    }while (num <= 0);
    return num;
}
```

continuação ...

```
float calculaSerie(int n){
    int a;
    float s;

    s=0;
    for (a=1; a<=n; a++)
        s = s + (float) 1/a;
    return s;
}
```




Outra possibilidade de main ()

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

```
float calculaSerie(int n);
```

```
int  obtemPositivo(void);
```

```
main(){
```

```
    printf("Valor:%f\n", calculaSerie(obtemPositivo()));
```

```
}
```

```
...
```



Mais de um return dentro de uma função

```
#include <stdio.h>

int ehDivisivel(int a, int b);

main(){
    int cod, n1, n2;

    printf("Informe um valor: ");
    scanf("%d", &n1);
    printf("Informe outro valor: ");
    scanf("%d", &n2);

    cod = ehDivisivel(n1, n2);
    if (cod==1)
        printf("%d é divisível por %d\n", n1, n2);
    else
        printf("%d não é divisível por %d\n", n1, n2);
}
```

continuação ...

```
int ehDivisivel(int a, int b){
    if (a % b == 0)
        return 1;
    else
        return 0;
}
```

Quando o **return** ocorre
a execução da função termina.



Outra possibilidade de main()

```
#include <stdio.h>

int ehDivisivel(int a,int b);

main(){
    int n1,n2;

    printf("Informe um valor: ");
    scanf("%d",&n1);
    printf("Informe outro valor: ");
    scanf("%d",&n2);

    if (ehDivisivel(n1,n2)==1)
        printf("%d é divisível por %d\n",n1,n2);
    else
        printf("%d não é divisível por %d\n",n1,n2);
}

...
```



Observações sobre o tipo de retorno

A ausência do tipo de retorno no cabeçalho da função indica que ela retorna um **int**, embora seja uma **boa prática definir o tipo retornado**.

...

```
main(){  
    int s;  
    s = soma(10, 20);  
    ...  
}
```

Indica que a função
retorna um **int**

```
soma(int a, int b){  
    return a + b;  
}
```



Retorno da função main()

Usado para indicar que a função

main() **não retorna valor**

`#include <stdio.h>`

`void main(){`

`printf("Alô mundo");`

`}`

`#include <stdio.h>`

`int main(){`

`printf("Alô mundo");`

`return 0;`

`}`

Indica que a função main()

encerrou corretamente.

Subalgoritmos com retorno de valor

Aula 22