Programação em Linguagem C

Funções

SENAI

Estrutura básica
função main()

main.c

```c
#include <stdio.h>

int main()
{
    printf("Hello World");

    return 0;
}

```

Programa principal (main)

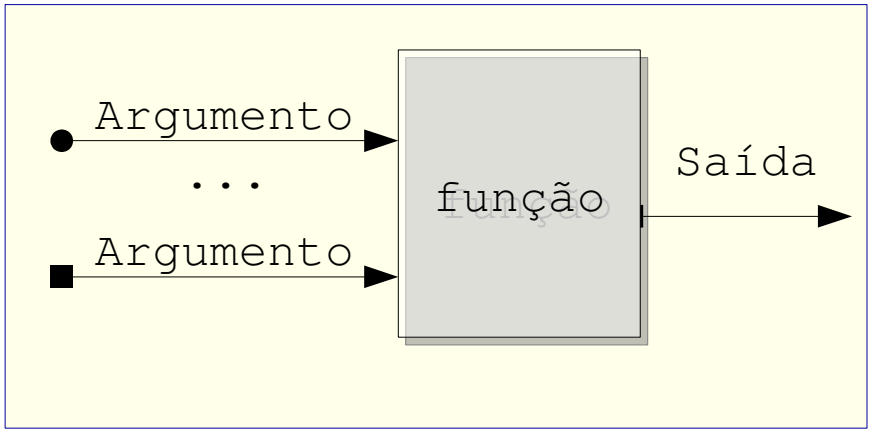
Todo programa em linguagem C é estruturado em uma ou mais **Funções.**

O nome das funções é de escolha do programador, exceto a única função obrigatória, **main.**

Estrutura básica de uma função

main.c

```c
#include <stdio.h>

int main()
{
    printf("Hello World");

    return 0;
}
```



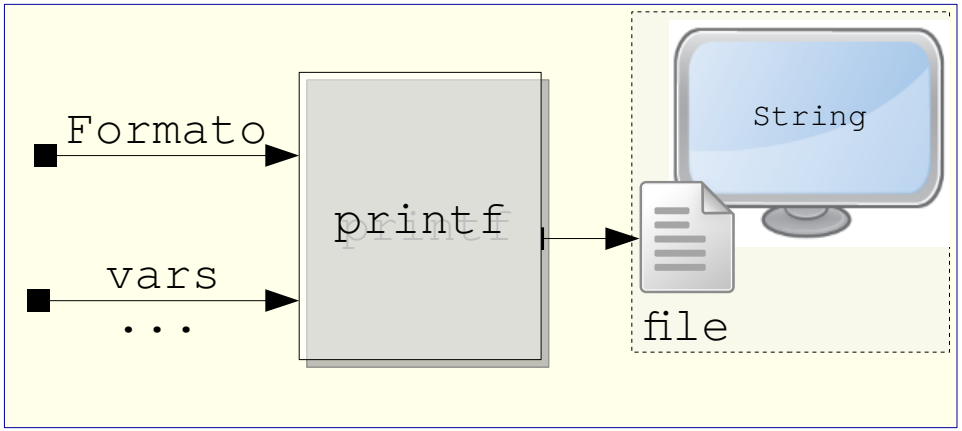
Estrutura de uma função

```
<tipo_saída> nome_da_funcao(<tipo_entrada> argumento,…)
{  início do corpo da função
    corpo da função com retorno de dado
}  fim do corpo da função
```

Função printf()

source code

```
1  #include <stdio.h>
2
3  int main()
4  {
5      printf("String");
6      return 0;
7  }
8
```



printf

Imprime uma cadeia de caracteres no arquivo de saída padrão.
A saída padrão, geralmente, é o monitor.

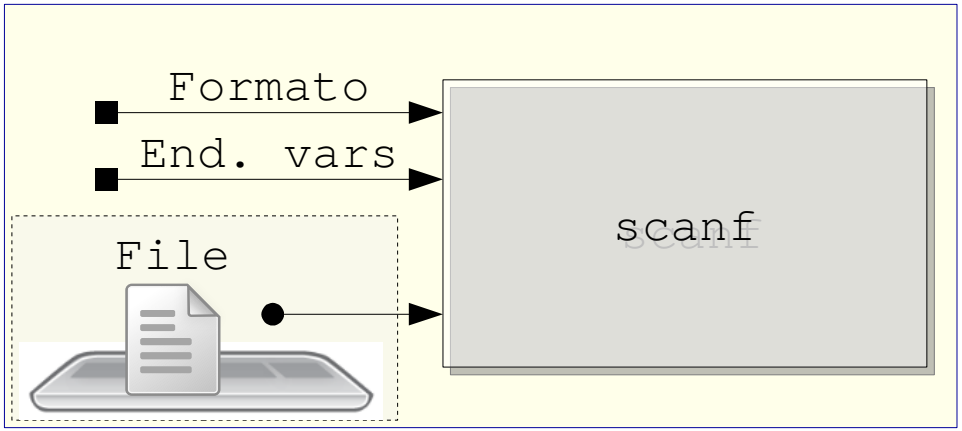
Memória

...	'S'	't'	'r'	'i'	'n'	'g'	\0	...

Função scanf()

main.c

```c
#include <stdio.h>

int main()
{
    int num = 0;
    printf("Numero: %d\n", num );
    scanf("%d", &num );
    printf("Numero: %d\n", num );
    return 0;
}
```

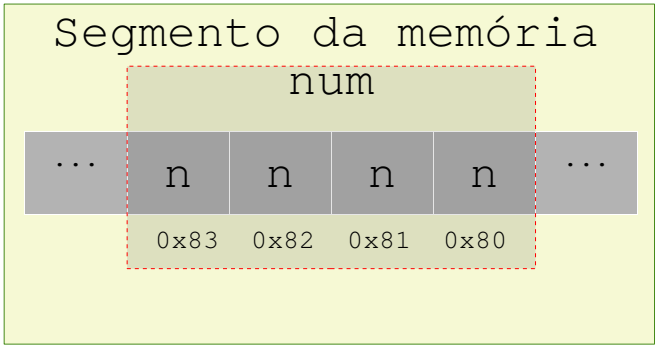


scanf()

Lê a entrada padrão, que geralmente, é o teclado

%d: tipo de dado de entrada: inteiro

&<var>: endereço da variável que receberá o dado lido.



74 = 0b01001010 = 0x4A

Função soma()

main.c

```c
        // Declaração de bibliotecas
#include <stdio.h>
        // Declaração da função soma()
int soma( int a, int b )
{
    return(a+b);
}
        // Programa principal
int main()
{
        // Declaração de variáveis
    int parcela1;
    int parcela2;
    int total;

        // Entrada de dados
    printf("Digite dois numeros: ");
    scanf("%d %d", &parcela1, &parcela2 );

        // Execução
    total = soma( parcela1, parcela2 );

        // Saída de dados/resultados
    printf("%d + %d = %d\n", parcela1, parcela2, total );

    return 0;
}
```

Declaração

Execução

função

Todas as funções procisam ser **declaradas** antes dos pontos de 'chamada' (**execução**).

Funções de associação de resistores

main.c

```c
        // Declaração de bibliotecas
#include <stdio.h>
        // Declaração das funções
int serie( int rx, int ry )
{
    return(rx + ry);
}
int paralelo( int rx, int ry )
{
    int prod;
    int soma;
    prod = rx * ry;
    soma = rx + ry;
    return( prod / soma );
}

        // Programa principal
int main()
{
        // Declaração de variáveis
    int resistor1;
    int resistor2;
    int reqSerie;
    int reqParalelo;

        // Entrada de dados
    printf("Digite o valor de dois resistores[em ohms]: ");
    scanf("%d %d", &resistor1, &resistor2 );

        // Execução
    reqSerie = serie( resistor1, resistor2 );
    reqParalelo = paralelo( resistor1, resistor2 );

        // Saída de dados/resultados
    printf("%d + %d = %d\n", resistor1, resistor2, reqSerie );
    printf("%d // %d = %d\n", resistor1, resistor2, reqParalelo );

    return 0;
}
```

Declaração

Execução

função

Todas as funções procisam ser **declaradas** antes dos pontos de 'chamada' (**execução**).

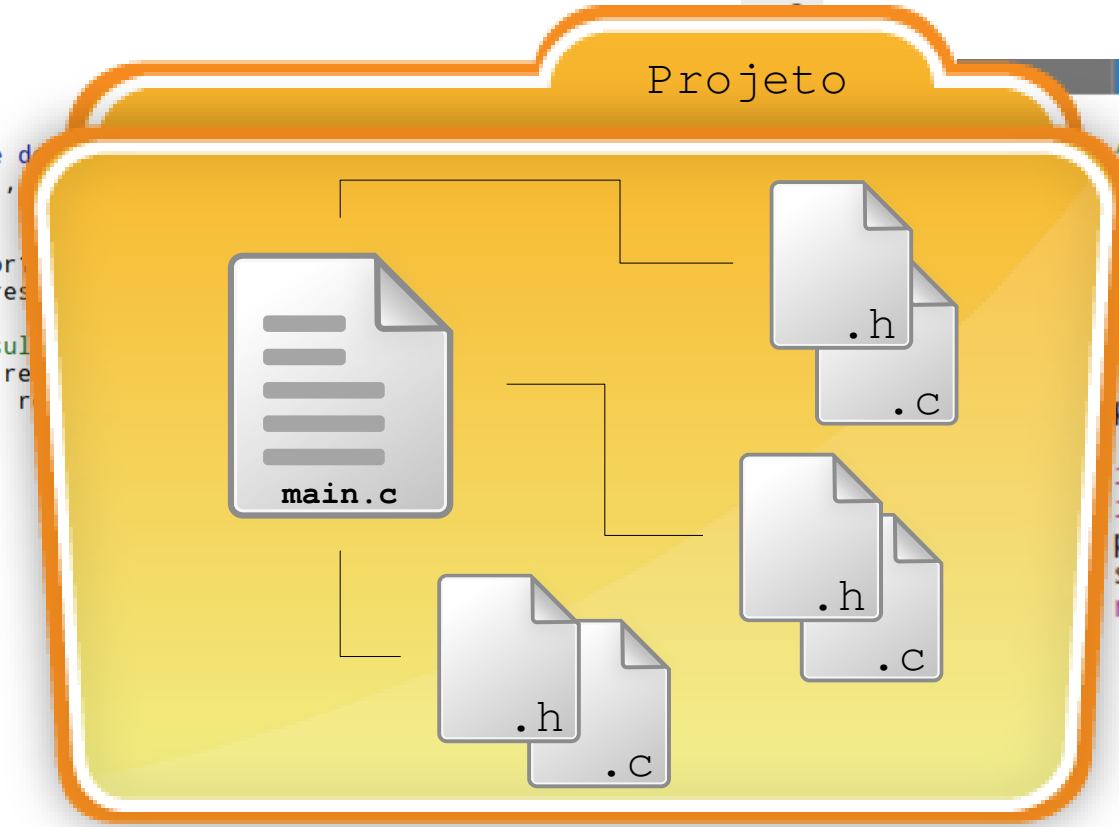
Organização do Projeto

main.c

```c
        // Declaração de bibliotecas
#include <stdio.h>
#include "req.h"

        // Programa principal
int main()
{
        // Declaração de variáveis
    int resistor1;
    int resistor2;
    int reqSerie;
    int reqParalelo;

        // Entrada de dados
    printf("Digite o valor de d
    scanf("%d %d", &resistor1,

        // Execução
    reqSerie = serie( resistor
    reqParalelo = paralelo( res

        // Saída de dados/resul
    printf("%d + %d = %d\n", re
    printf("%d // %d = %d\n", r

    return 0;
}
```

req.h

```c
#ifndef REQ_H
#define REQ_H

int serie( int rx, int ry );
int paralelo( int rx, int ry );

#endif
```



req.c

```c
// Declaração das funções de
// calculo para
// associação de resistores

erie( int rx, int ry )

return(rx + ry);

paralelo( int rx, int ry )

int prod;
int soma;
prod = rx * ry;
soma = rx + ry;
return( prod / soma );
```

Organização do projeto

main.c

```c
        // Declaração de bibliotecas
#include <stdio.h>
#include "req.h"

        // Programa principal
int main()
{
        // Declaração de variáveis
    int resistor1;
    int resistor2;
    int reqSerie;
    int reqParalelo;

        // Entrada de dados
    printf("Digite o valor de dois resistores[em ohms]: ");
    scanf("%d %d", &resistor1, &resistor2 );

        // Execução
    reqSerie = serie( resistor1, resistor2 );
    reqParalelo = paralelo( resistor1, resistor2 );

        // Saída de dados/resultados
    printf("%d + %d = %d\n", resistor1, resistor2, reqSerie );
    printf("%d // %d = %d\n", resistor1, resistor2, reqParalelo );

    return 0;
}
```

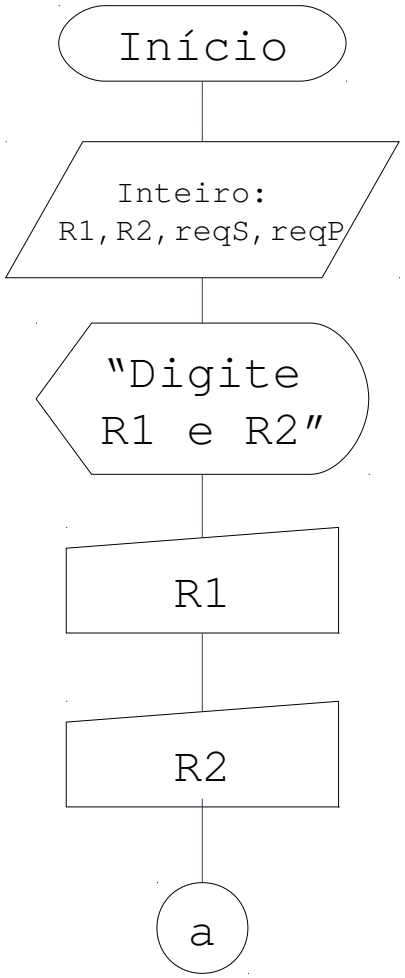
req.h — .h

```c
#ifndef REQ_H
#define REQ_H

int serie( int rx, int ry );
int paralelo( int rx, int ry );

#endif
```

req.c — .c

```c

    // Declaração das funções de
    // calculo para
    // associação de resistores

int serie( int rx, int ry )
{
    return(rx + ry);
}
int paralelo( int rx, int ry )
{
    int prod;
    int soma;
    prod = rx * ry;
    soma = rx + ry;
    return( prod / soma );
}
```

Fluxograma


Início
Inteiro:
R1,R2,reqS,reqP
"Digite
R1 e R2"
R1
R2
a

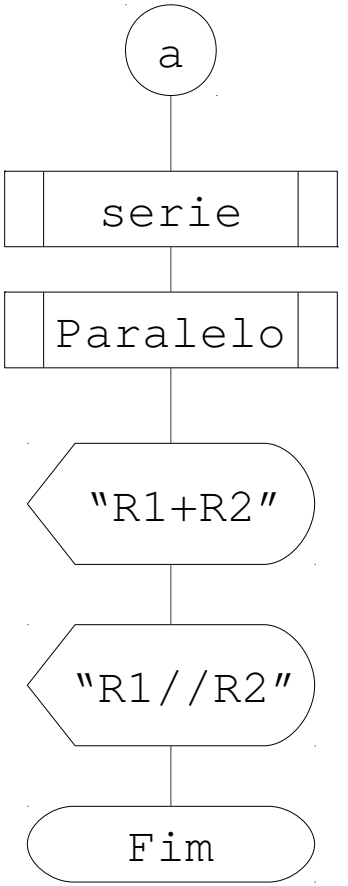
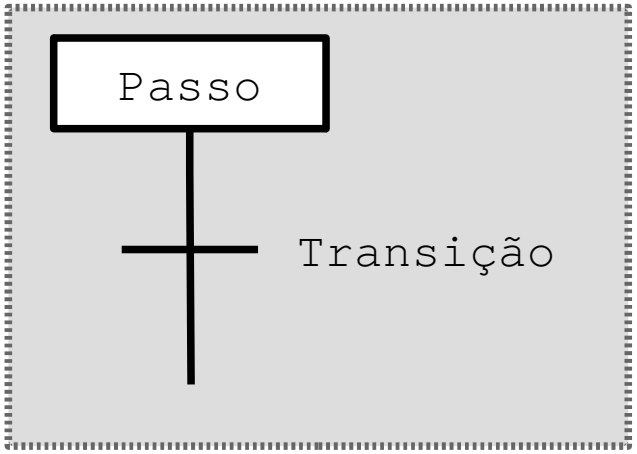
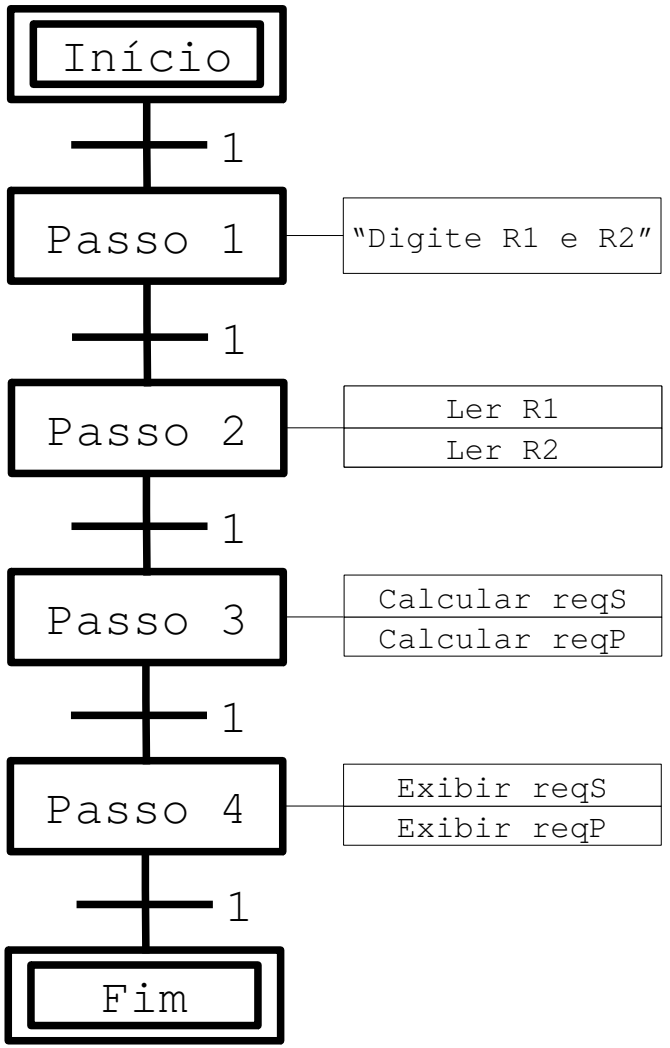

a
serie
Paralelo
"R1+R2"
"R1//R2"
Fim

Grafico de Funções Sequenciais

SFC – Sequential Function Chart





Site: **josewrpereira**.github.io/ddp/
E-mail: **josewrpereira**@outlook.com
jose.wpereira@senaisp.edu.br