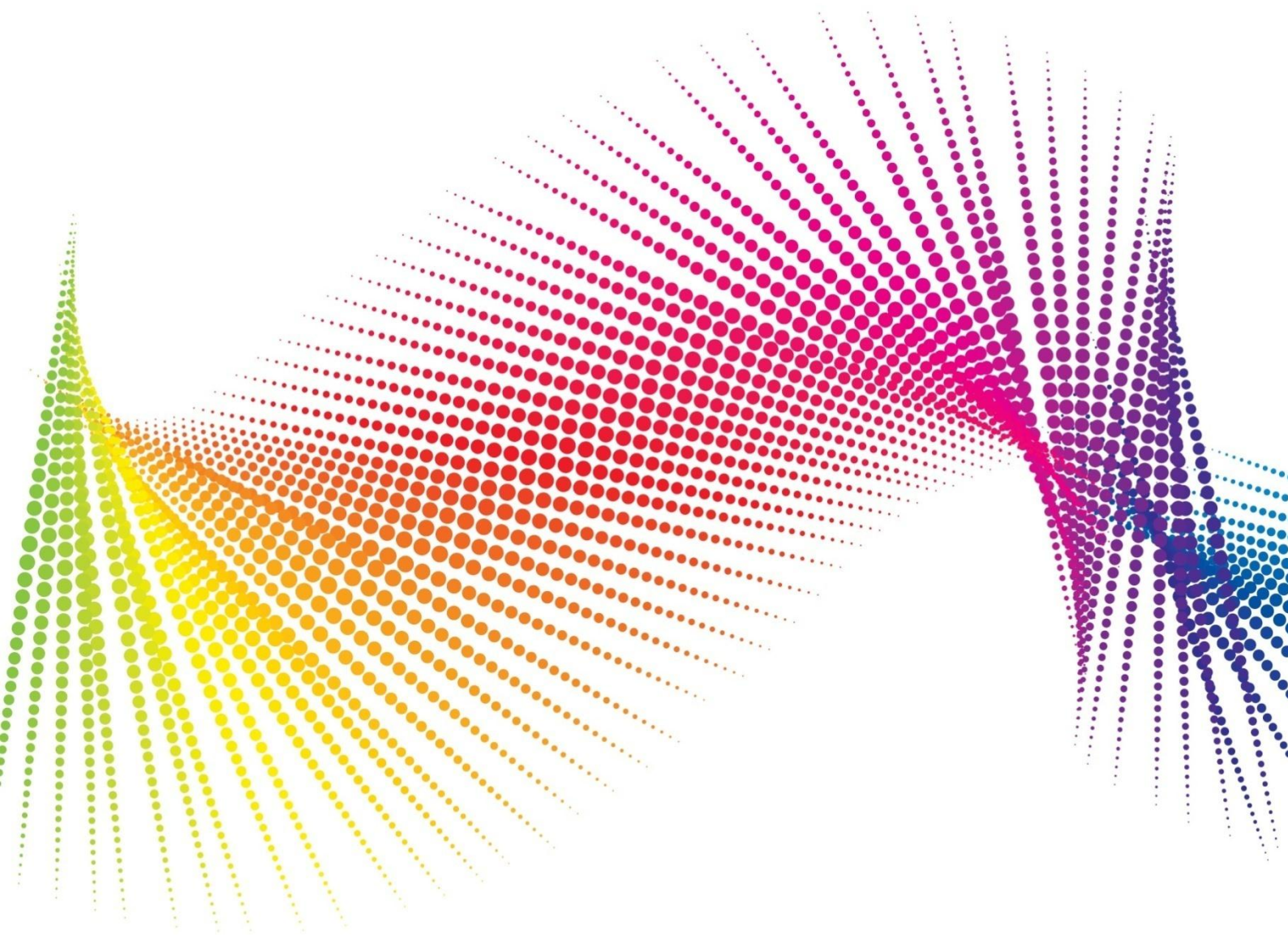


Estrutura de Dados

Aula 07



Este material é parte integrante da disciplina oferecida pela UNINOVE.

O acesso às atividades, conteúdos multimídia e interativo, encontros virtuais, fóruns de discussão e a comunicação com o professor devem ser feitos diretamente no ambiente virtual de aprendizagem UNINOVE.

Uso consciente do papel.

Cause boa impressão, imprima menos.

Aula 7: Funções

Objetivo: Dar início aos estudos das funções na linguagem C.

Estrutura de um programa em C

Caro aluno, um programa C é dividido em três partes: componentes iniciais, função `main()`, e funções.

Componentes iniciais:

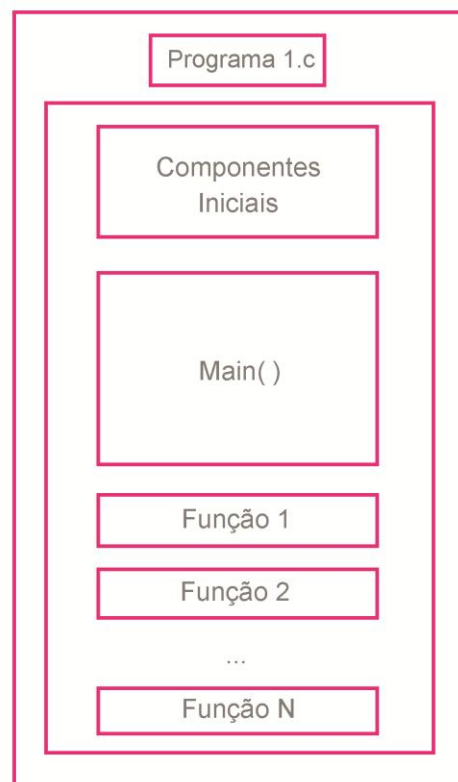
- Comandos de pré-processamento:
- Inclusão de arquivos de cabeçalho (da biblioteca);
- Definição de constantes.
- Declaração de variáveis, estruturas e ponteiros globais.
- Inicialização de variáveis, constantes, estruturas e ponteiros globais.
- Declaração de protótipos de funções: fornece ao compilador o nome e os parâmetros das funções contidas no programa.

Função `main()`:

- Função especial cujo programa inicia sua execução.
- Contém as instruções que comandam toda a execução do programa.
- Pode ter variáveis, estruturas e ponteiros locais.

Funções:

- Desempenham tarefas específicas no programa.
- São criadas pelo programador e são opcionais, podem ou não existir.



A figura a seguir ilustra um programa C completo em que esses três componentes estão destacados. O programa possui uma função `media`. O protótipo

da função `media` é declarado nos componentes iniciais. A função `media` é chamada dentro da função `main()`. O corpo da função `media` é declarado após a função `main()`.

```
// Media.c - calcula a média aritmética de dois números inteiros

#include <stdio.h>

int a,b;    // declara a e b como variáveis inteiras globais
float m;    // declara m como variável de ponto flutuante global

float media(int x, int y); /* declara o protótipo da função media
                           como uma função que recebe 2
                           números inteiros
                           e retorna um float */
```

```
int main(void)
{
// Solicita o primeiro valor usando a função de biblioteca printf
printf("Qual o primeiro valor ? ");

// Lê o primeiro valor da tela usando a função de biblioteca scanf
scanf("%i",&a);

// Solicita o segundo valor usando a função de biblioteca printf
printf("Qual o segundo valor ? ");

// Lê o segundo valor da tela usando a função de biblioteca scanf
scanf("%i",&b);

m=media(a,b); /*Chama a função média para calcular a
               média entre a e b e atribui o resultado a m*/

// Mostra na tela o resultado
printf("\nA média é %.2f: ",m);
return 0;
}
```

```
float media(int x, int y)
{ float result;

  result = (float) (x+y) /2;
  return (result);
}
```

Acesse ao AVA e assista à animação para conhecer mais sobre o conceito de funções. Esta animação faz parte da sequência desta aula e, portanto, é importante para a sua aprendizagem.

Conceito de funções

As funções são trechos de códigos ou subprogramas que realizam tarefas específicas, recebem ou não informações por meio de argumentos ou parâmetros, e são designadas por um nome, podendo ou não retornar um valor à função que as chamou. As vantagens de se usar funções é que elas possibilitam a realização de uma mesma tarefa a partir de vários pontos do programa, sem a necessidade de se repetir código, permitem que se isole o código e as variáveis do restante do programa e tornam mais fácil a depuração do programa.

O protótipo de uma função

Na linguagem C, é usual que o programa contenha o **protótipo** e a **definição** da função. O protótipo é declarado nos componentes iniciais e fornece ao compilador uma descrição da função que será definida posteriormente, após a função `main()`.

O protótipo deve ser colocado antes da primeira chamada à função, de preferência antes da função `main ()`. Ele deve possuir o tipo de variável de retorno, nome da função, os tipos de variáveis que são os parâmetros de entrada (ou parâmetros formais) e, opcionalmente, nomes associados a cada um destes tipos.

Tipos de função

Existem funções com e sem retorno de valor. Resumidamente, as funções com retorno de valor são declaradas **como tipo int, float ou char**. As funções sem retorno de valor são declaradas como tipo **void**.

Exemplos

Função com retorno de valor. Somar 2 números

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

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

```
void somar( float , float);
```

```
main()
```

```
{
```

```
float a,b;
```

```
printf("\n Entre com 2 numeros:");
```

```
scanf("%f %f",&a,&b);
```

```
somar(a,b);
```

```
system("pause >> log");
```

```
}
```

```
void somar(float a, float b)
```

```
{
```

```
float soma;
```

```
soma=a+b;
```

```
printf("\n %.2f + %.2f = %.2f\n",a,b,soma);
```

```
}
```

Protótipo: estamos informando ao compilador que este programa trabalha com uma função sem retorno de valor. Ela tem o nome **somar** e ela possui dois parâmetros reais para a entrada de dados.

As variáveis da função main só existem aqui dentro da função main e são chamadas variáveis.

Aqui chamamos a função somar e passamos os 2 valores que queremos somar. Estas 2 variáveis têm que ser passadas porque elas só existem aqui dentro da função principal.

Aqui é o código da função propriamente dito, ou seja, é o que ela faz. Ela recebe os 2 valores passados e os soma. **Observe que foi printado o resultado na tela dentro da função somar** que senão não seria possível ver o resultado em outro local porque a soma só existe aqui dentro.

Função com retorno de valor. Somar 2 números

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

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

Protótipo: estamos informando ao compilador que este programa trabalha com uma função com retorno de valor. Ela possui o nome **somar** e ela possui dois parâmetros reais para a entrada de dados. O seu valor de retorno também é tipo real.

```
float somar( float , float);
```

```
main()
```

```
{  
float a,b,soma;
```

Criando 3 variáveis reais. Agora temos também a variável soma, Todas elas só existirão dentro da função main.

```
printf("\n Entre com 2 numeros:");
```

```
scanf("%f %f",&a,&b);
```

```
soma=somar(a,b);
```

Chamando a função passando 2 valores e recebendo o resultado.

```
printf("\n %.2f + %.2f = %.2f\n",a,b,soma);
```

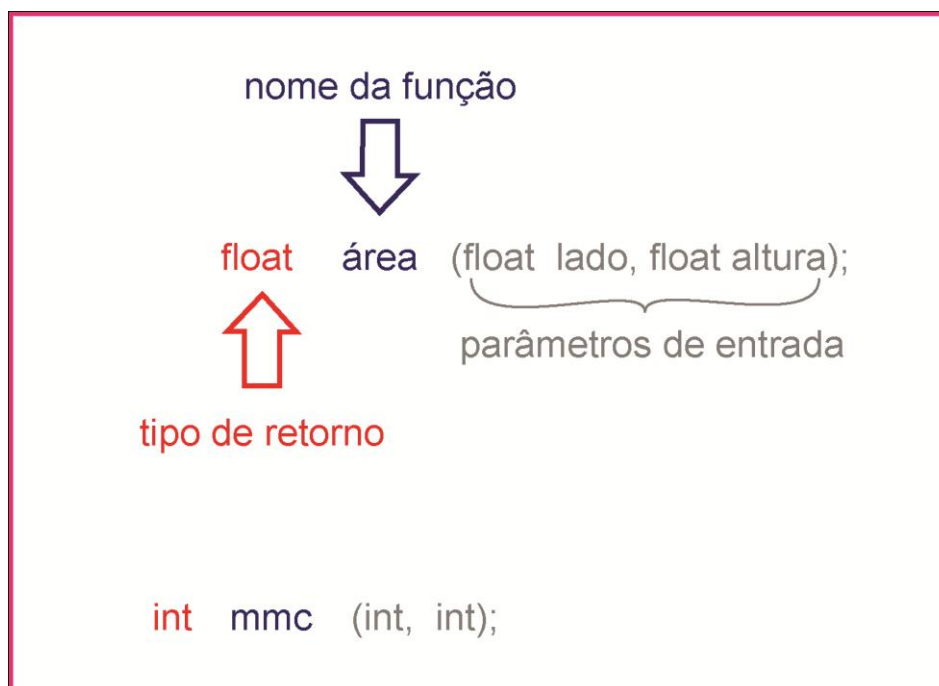
```
system("pause >> log");
```

```
}
```

```
float somar(float a, float b)
```

```
{  
float soma;  
soma=a+b;  
return soma;  
}
```

Aqui é o código da função propriamente dito, ou seja, é o que ela faz. Ela recebe os 2 valores passados e os soma. Observe o comando RETURN. Este comando retorna o valor da soma para a função principal e lá este valor poderá ser printado na tela.



Agora, caro aluno, vamos praticar resolvendo os exercícios propostos. **Leia a lista, resolva os exercícios e verifique seu conhecimento.** Caso fique alguma dúvida, leve a questão ao Fórum e divida com seus colegas e professor.

REFERÊNCIAS

MIZRAHI, V.V. *Treinamento em linguagem C. Módulo 1.* 2 ed. São Paulo: Makron Books, 2006.

_____. *Treinamento em linguagem C. Módulo 2.* 2 ed. São Paulo: Makron Books, 2006.

SCHILDT, H. *C Completo e total.* 3. ed. São Paulo: Makron Books, 1997.