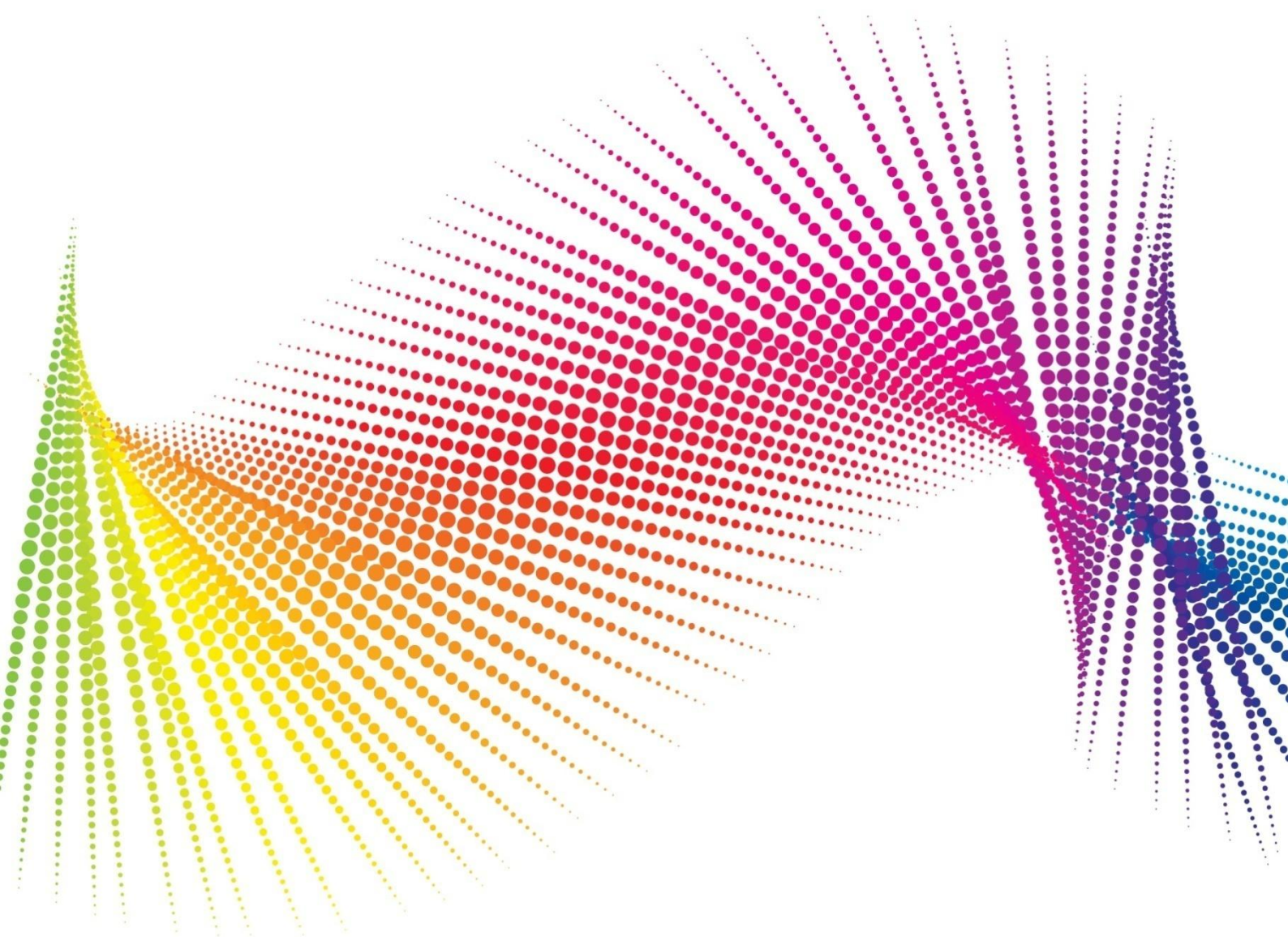


Estrutura de Dados

Aula 01



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 01: Revisão

Objetivo: Um breve resumo sobre a lógica de programação.

Introdução

Caro aluno, seja muito bem-vindo à disciplina de Estrutura de Dados. Começamos hoje uma jornada muito interessante e produtiva que o levará a conhecer mais sobre como organizar e estruturar dados no computador, bem como aprofundar mais seus conhecimentos na instigante e inovadora linguagem de programação C. Lembre-se de que aprender a programar em C, sabendo usar bem as estrutura de dados, é alicerçar uma promissora carreira de desenvolvedor! “C retém a filosofia básica de que os programadores sabem o que estão fazendo” (KERNIGHAN; RITCHE, 1989).

Antes de começar, você já deve ter instalado um compilador C em sua máquina porque foi tema da disciplina anterior, Lógica de Programação, (usamos o compilador DEV C++ que pode ser baixado pela internet).

Vamos fazer um “jogo rápido” revisando os principais pontos da lógica de programação.

Variáveis

Todos os dados que entram via teclado no computador obrigatoriamente ficam armazenados em variáveis. As variáveis são tipos abstratos de dados organizados e classificados em tipo inteiro, real, caracter e lógico, lembra-se?

Um programa exemplo envolvendo variáveis.

```
#include <stdio.h>
#include <stdlib.h>
main()
```

```

{
    int a,b,temp;
    printf("\nEntre com um valor em A:");
    scanf("%d",&a);
    printf("\nEntre com um valor em B:");
    scanf("%d",&b);
    //troca
    temp = a;
    a = b;
    b= temp;
    printf("\n Agora A=%d e B=%d\n");
    system("pause");
}

```

Criando 3 variáveis inteiras.

Só aceita números. Um valor não numérico não é aceito e dará erro.

// indica um comentário que só pode ser feito em 1 única linha. Para fazer comentários que possuem mais de 1 linha, deve-se usar o símbolo /*... */

Troca o conteúdo entre as variáveis.

Programa 1.2.1. Trocar o conteúdo entre 2 variáveis.

Condição

Dados 2 números quaisquer, a lógica a seguir vai colocá-los em ordem crescente de valor:

```

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

main()
{
    float a, b;

    printf("\n Entre com 2 numeros:");
    scanf("%f %f",&a,&b);
    printf("\n Os numeros em ordem crescente de valor:\n");
    if(a < b)
        printf("\n %.2f \n %.2f", a,b);
    else printf("\n %.2f \n %.2f", b,a);
}

```

Criando 2 variáveis reais. Isto quer dizer que posso entrar com valores que tem parte decimal como 2.5.

Pegando do teclado os números digitados.

Comparando os dois números

Mostrando sempre em ordem crescente.

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

Programa 1.2.2. Colocar em ordem crescente 2 números.

Para conhecer um pouco mais sobre os comandos de repetição, acesse o material complementar que contém exemplos válidos e inválidos dos comandos de repetição e exemplos de programas que os utilizam. Esse material faz parte da sequência desta aula e, portanto, é essencial para a aprendizagem.

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.

Exercícios propostos

- 1) Fazer um programa que verifica se um número é divisível por 5.
- 2) Fazer um programa que verifica se um número entrado via teclado está entre 0 e 100.

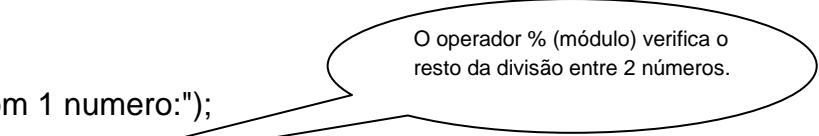
Respostas dos exercícios propostos

Exercício 1.

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

```
main()  
{  
    int x;
```

```
    printf("\n Entre com 1 numero:");  
    scanf("%d",&x);  
    if(x % 5 == 0)  
        printf("\n %d e divisivel por 5", x);  
    else printf("\n %d e nao e divisivel por 5", x);
```



O operador % (módulo) verifica o resto da divisão entre 2 números.

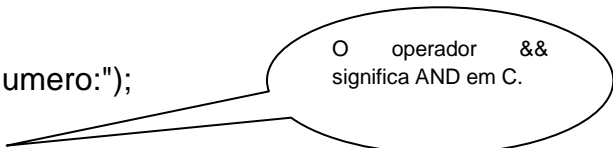
```
system("pause >> log");  
}
```

Exercício 2.

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

```
main()  
{  
    int x;
```

```
    printf("\n Entre com 1 numero:");  
    scanf("%d",&x);  
    if(x >=0 && x <= 100)  
        printf("\n %d esta entre 0 e 100", x);  
    else printf("\n %d nao esta entre 0 e 100", x);
```



O operador &&
significa AND em C.

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

REFERÊNCIAS

KERNIGHAN, B. W.; RITCHIE, D. M. *C- A Linguagem de Programação Padrão ANSI*. Rio de Janeiro: Campus, 1989.

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*. São Paulo: Makron Books, 1997.