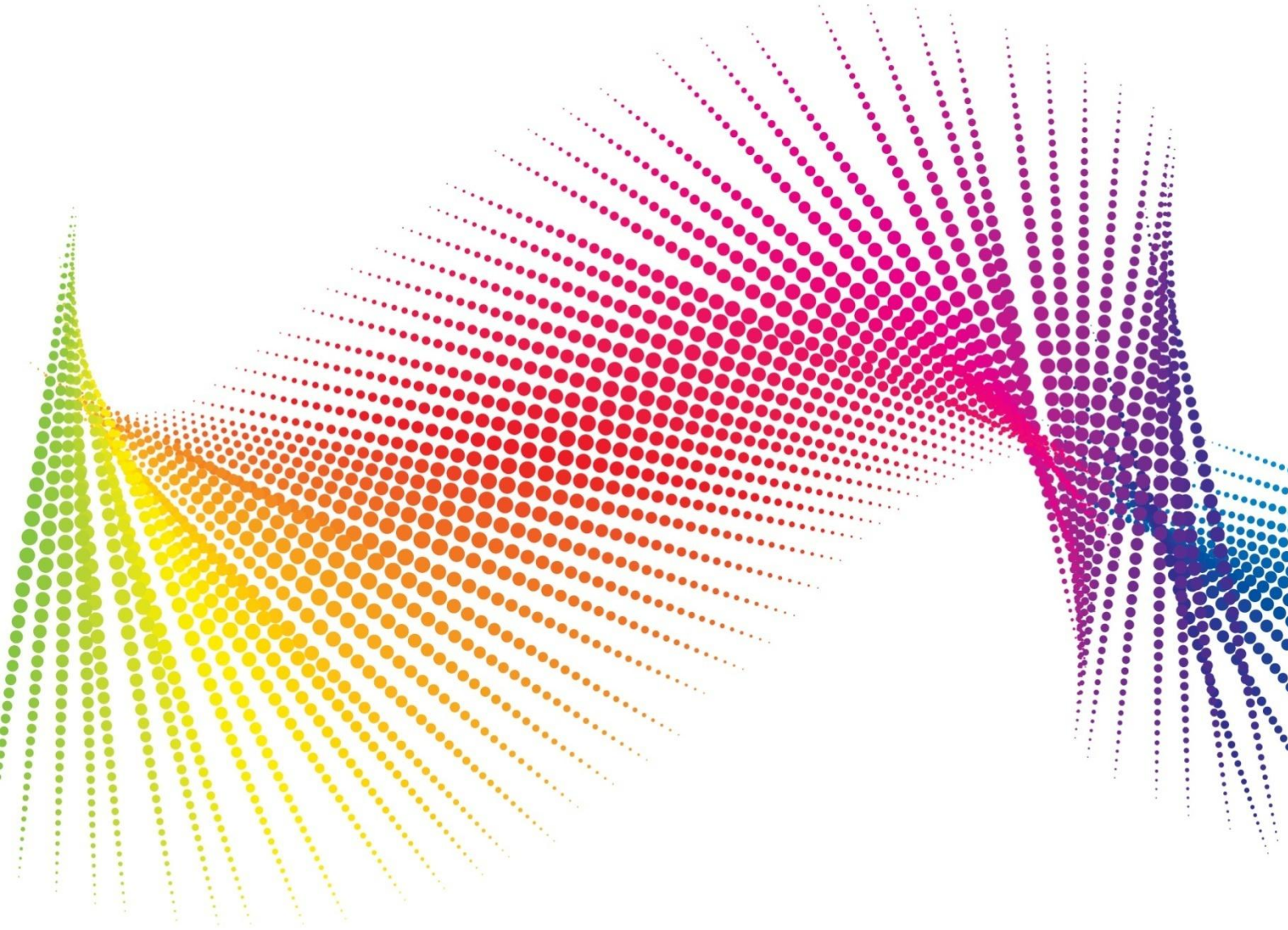
UNINOVE
Universidade Nove de Julho

Estrutura de Dados

Aula 04



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 04: Strings

Objetivo: Estudar as strings, que são cadeias de caracteres.

Declaração de uma string

Strings nada mais são que vetores de caracteres cujo último elemento é igual a um nulo ('\0'). Elas são cadeias de caracteres e se diferenciam dos caracteres únicos pelos colchetes –[ ] – usados na sua declaração. Veja a diferença entre uma string (`palavra`) e um caracter único (`letra`):

```
char letra = "A", palavra[10]= "exemplo";
```

Toda string tem 0 (zero) como índice do seu primeiro caracter e termina com o caracter nulo \0.

Sintaxe

```
char  nome_da_string[tamanho];
```

Exemplos

```
char nome[100];
//declaramos a variável nome que pode armazenar até 100 caracteres
char str[9]= "Bom dia!";
//pode-se inicializar a string str na declaração
char str[]= "Bom dia!";
//caso se inicialize a string na declaração o tamanho fica opcional,
  mas os colchetes [] devem ser mantidos
char str[]= {'B','o','m',' ','d','i','a','!','\0'};
//pode-se usar uma string como um vetor de caracteres, mas
//deve-se colocar o \0 para finalizá-la
char *vetdias[]={"segunda","terça","quarta","quinta",
                 "sexta","sabado","domingo"};
```

```
      //aqui estamos criando um vetor em que cada posição armazena uma
      // palavra ou frase. Depois entenderemos o que o * significa, ok?
```

Exemplo de uso de strings

```
 /* string1.c: Mostra o uso de strings     */
 #include <stdlib.h>
#include <stdio.h>

main()
{
  char nome[50];//criamos aqui um vetor de caracteres

  printf("\n Entre com se nome para guardar no variavel nome:");
  fflush(stdin);//limpa o buffer (memória da entrada de dados
  gets(nome);//comando que lê a string digitada no teclado

  printf("\n O seu nome: %s foi armazenado na variavel
nome\n",nome);

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

Vamos entender alguns trechos usados neste programa:

1) Comando `gets(`: serve para ler apenas strings ou frases. Com `gets` podemos ler várias palavras separadas por espaço. O `gets` para de ler apenas quando o ENTER é pressionado. Poderíamos ainda ler com o comando `scanf`, mas é um pouco mais sofisticado o seu uso, senão ele consegue ler somente uma palavra, pois o espaço finaliza a leitura.

2) O comando system (“pause >> log”) é para pausar a tela. O símbolo >> redireciona uma mensagem automática (“pressione qualquer tecla para continuar”) para o arquivo log.

Funções especiais para tratamento de strings: biblioteca <string.h>

Todos que programam em C costumam reclamar do uso de strings, que é muito chato, pois precisa do uso de funções. Não podemos nem fazer uma atribuição a uma string, pois até a atribuição de um valor tem que ser feita usando a função `strcpy()`. Prossigamos para entender melhor isso.

Toda operação que envolve strings deve ser feita por meio de funções especiais que se encontram na biblioteca <string.h>.

As principais funções da biblioteca <string.h> são:

strcat (destino, fonte)

anexa o string <fonte> ao string <destino>

strcpy (destino, fonte)

copia o string <fonte> para o string <destino>

strcmp (string1,string2)

compara o <string1> com o <string 2> pela ordem alfabética (conforme tabela ASCii), resultando em:

menor que 0 .... <string1> menor que <string2>
igual a 0 .... <string1> igual a <string2>
maior que 0 .... <string1> maior que <string2>

strchr(string, caracter)

verifica se o <caracter> se encontra na <string> e retorna a posição da primeira ocorrência do <caracter> no <string>;
se o <caracter> não for encontrado, retorna NULL

tamanho = strlen(string);

retorna o <tamanho> de uma <string> em número de caracteres.

strrev(string)

inverte a ordem das letras de <string>.

strupr(string)

converte <string> para letras maiúsculas.

strlwr(string)

converte <string> para letras minúsculas.

Caro aluno, acesse ao AVA e teste seus conhecimentos entretendo-se com as palavras cruzadas sobre strings.

Agora vamos apresentar vários exemplos para você entender como usar estas funções, ok? Teste-as e, caso tenha alguma de dúvida, tire-as nos fóruns.

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*. São Paulo: Makron Books, 1997.