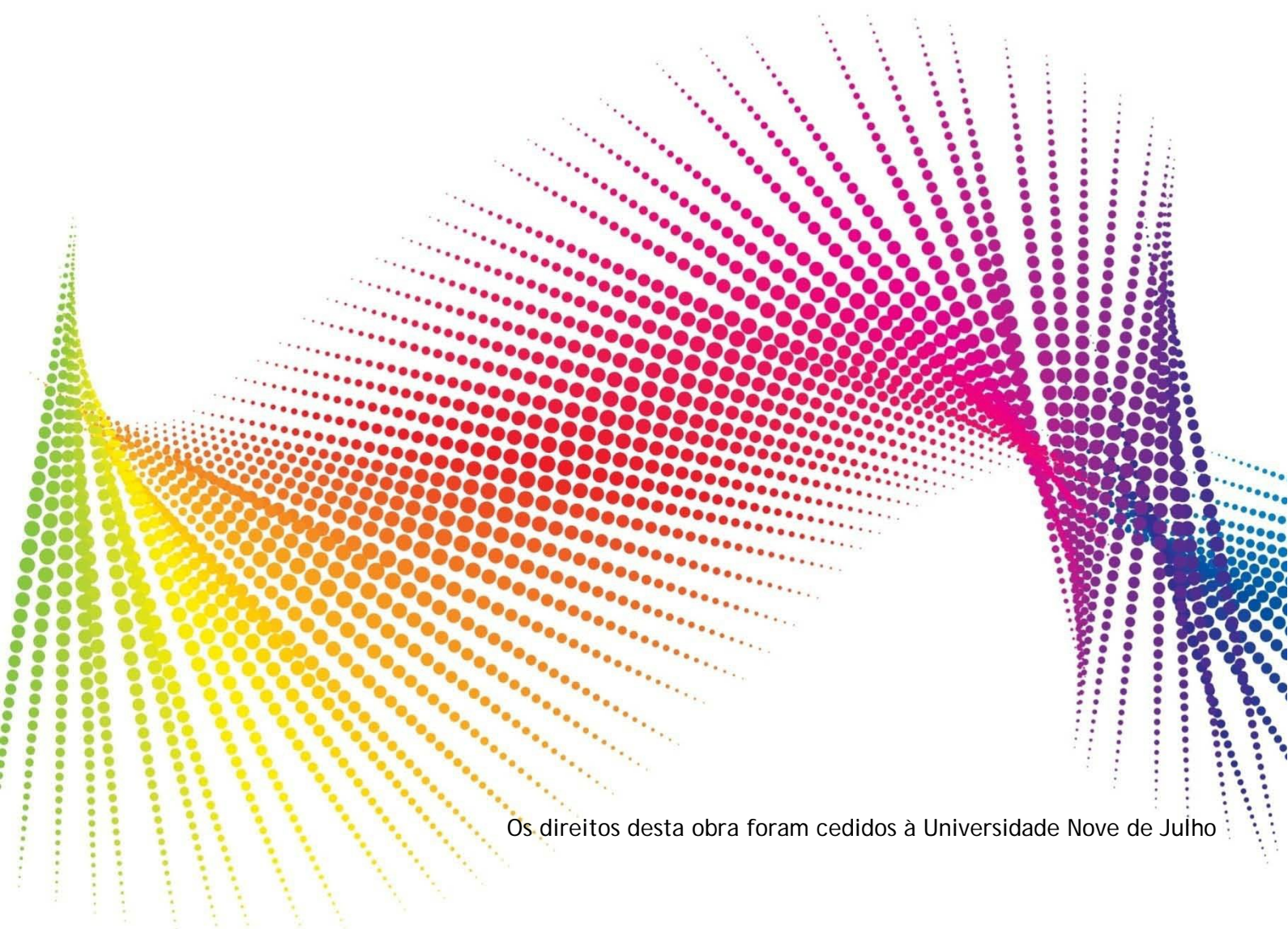


Aula 14

Vetores de estruturas



Os direitos desta obra foram cedidos à Universidade Nove de Julho

Aula 14 – Vetores de estruturas

Caro aluno,

Se você for recortar as soluções dos programas deste arquivo e colá-las no compilador DevC++, tome cuidado pois as aspas e o sinal de menos precisam ser substituídos, senão vai dar erro.

Outra dica: não use acentos nas palavras, pois não funcionam nos programas.

Lista de exercícios da aula 14

Vamos praticar um pouco mais usando vetores de estruturas. Você deve tentar fazer os exercícios sozinho e somente depois ver as respostas, ok? Boa sorte!

1. Desenhe a estrutura declarada:

```
struct reg {    char nome [30];  
  
               long int cpf;  
  
               float salario;  
  
               char cargo[30];  
  
} vetreg[10];
```

- 2.** Modifique o método da bolha e crie uma nova versão para que ele leia nomes e ordene os nomes. Use um vetor de estrutura.
- 3.** Um pequeno município deseja cadastrar seus contribuintes proprietários de imóveis, verificar se seu IPTU está em dia e calcular a multa daqueles que estão em débito. Faça um programa para resolver o problema da prefeitura, considerando que:
- Serão fornecidos o nome do proprietário, o número do imóvel, o valor do IPTU e os meses em atraso (último nome = “fim”).

- A multa deve ser calculada de acordo com a tabela a seguir, que deve estar armazenada em memória:

valor do iptu	% por mês em atraso
até 2000,00	1%
de 2000,01 a 4000,00	3%
de 4000,01 a 7000,00	5%
de 7000,01 a 10000,00	8%
acima de 10000,00	10%

Respostas dos exercícios

```
1. struct reg { char nome [30];  
                long int cpf;  
  
                float salario;  
  
                char cargo[30];  
            } vetreg[10];
```

vetreg

0	nome		cpf		salario		cargo	
1	nome		cpf		salario		cargo	
2	nome		cpf		salario		cargo	
3	nome		cpf		salario		cargo	
4	nome		cpf		salario		cargo	
5	nome		cpf		salario		cargo	
6	nome		cpf		salario		cargo	
7	nome		cpf		salario		cargo	
8	nome		cpf		salario		cargo	
9	nome		cpf		salario		cargo	

2.

/* bolhaAlfa.c: implementa o metodo da bolha para ordenacao de vetor - bubble sort

vetor de nomes */

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

```
#include <conio.h>
```

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

```
#define N 10
```

```
struct info { char nome[20];
```

```
};
```

```
struct info vet[N+1], aux;
```

```

int  bolha, lsup, j, i;

int  cont=0; ///conta numero de comparacoes


int main()
{
    printf("\nMetodo da Bolha - ordenacao de nomes");

    printf("\n\nVetor a ser ordenado \n");
    for (i=1; i<=N; i++)
    {   printf("\n Digite o nome %d ",i);
        scanf("%s",vet[i].nome);
    }

    lsup=N;

    while (lsup > 1)
    { bolha = 0;
        for (j=1; j <= (lsup-1); j++)
        { if(strcmp(vet[j].nome,vet[j+1].nome)> 0)
            { aux = vet[j];
              vet[j] = vet[j+1];
              vet[j+1] = aux;

              bolha = j;//bolha guarda a posicao onde ocorreu a ultima troca
            }

            ++cont;
        }

        lsup = bolha;
    }

    printf("\nVetor ordenado pelo metodo da Bolha");

    for (j=1; j<=N; j++)
    { printf("\n Nome = %s", vet[j].nome);

```

```
    getch();  
}
```

```
printf("\n\n Numero de comparacoes no BOLHA: %d ", cont);  
printf("\n\n\nFim do programa");  
getch();  
return 0;  
}
```

3.

/*iptu.c: calcula a multa sobre o IPTU de contribuintes -
ilustra o uso de struct */

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

```
#include <conio.h>
```

```
#include <string.h>
```

```
struct tab {float de, ate, porc;  
            } tabela[5] = {           0.0,  2000.00,  0.01,  
                                2000.01, 4000.00,  0.03,  
                                4000.01, 7000.00,  0.05,  
                                7000.01, 10000.00, 0.08,  
                                10000.01, 1000000.00, 0.1,  
                                };
```

```
struct d {  char nome[30];  
            int imovel;  
            float iptu;  
            int atraso;  
        } dados;
```

```
float multa;
```

```
int i, achou;
```

```
int main(void)
```

```
{
```

```
    while (1)
```

```
    { printf("\nDados do contribuinte");  
      printf("\n=====");  
      printf("\n\nNome (fim para terminar): ");  
      gets(dados.nome);
```

```
      if (strcmp(dados.nome,"fim") == 0)  
          break;
```

```
      printf("\nNumero do imovel: ");  
      scanf("%d",&dados.imovel);  
      printf("\nValor do IPTU: ");  
      scanf("%f",&dados.iptu);  
      printf("\nMeses atrasados: ");  
      scanf("%d",&dados.atraso);
```

```
      achou = 0;
```

```
      if (dados.atraso == 0)
```

```
      { multa = 0;
```

```
        achou = 1;
```

```
        printf("\nO contribuinte %s nao esta em atraso e nao tem multa a pagar.",
```

```

        dados.nome);
    }

    /* localiza a faixa de iptu do contribuinte */

    i = 0;
    while (achou == 0)
    {
        if (dados.iptu >= tabela[i].de && dados.iptu <= tabela[i].ate)
        {
            multa = dados.iptu*dados.atraso*tabela[i].porc;
            achou = 1;
            printf("\nO contribuinte %s tem %d meses em atraso ",
                    dados.nome,dados.atraso);
            printf("\n e pagara uma multa de R$%.2f calculada com %.0f %% de juros ao mes
",
                    multa,tabela[i].porc*100);
        }
        else ++i;
    }
    printf("\n");
    getch();
    fflush(stdin);

}

printf("\nFim do programa... pressione <enter>");
getch();
return 0;
}_

```