

# SSC0501 - Introdução à Ciência de Computação I

## Resposta da 9ª Lista de Exercícios

**Professor:** Claudio Fabiano Motta Toledo (claudio@icmc.usp.br)

**Estagiário PAE:** Jesimar da Silva Arantes (jesimar.arantes@usp.br)

---

### Resposta pergunta 1:

```
1 #include <stdio.h>
2 #include <string.h>
3
4 #define TAM 5
5
6 struct carro{
7     char fabricante[20];
8     char modelo[20];
9     char cor[20];
10    int ano;
11    int preco;
12 };
13
14 void lerDados(struct carro *car);
15 void imprimirDados(struct carro car);
16 void imprimirTodosDados(struct carro car[], int tam);
17 void modificarDados(struct carro car[], int tam);
18
19 int main(){
20     printf("-----Carro-----\n\n");
21     struct carro car[TAM];
22     int i;
23     for (i = 0; i < TAM; i++){
24         printf("    Ler Dados: \n");
25         lerDados(&car[i]);
26     }
27     imprimirTodosDados(car, TAM);
28     modificarDados(car, TAM);
29     imprimirTodosDados(car, TAM);
30     return 0;
31 }
32
33 void lerDados(struct carro *car){
34     char fab[20];
35     scanf("%s", fab);
36     strcpy((*car).fabricante, fab);
37
38     char mod[20];
39     scanf("%s", mod);
40     strcpy((*car).modelo, mod);
41
42     char cor[20];
43     scanf("%s", cor);
44     strcpy((*car).cor, cor);
45
46     int ano;
```

```
47  scanf("%d", &ano);
    (*car).ano = ano;
49
    int preco;
51  scanf("%d", &preco);
    (*car).preco = preco;
53 }

55 void imprimirDados(struct carro car){
    printf("\n");
57  printf("Fab: %s\n", car.fabricante);
    printf("Mod: %s\n", car.modelo);
59  printf("Cor: %s\n", car.cor);
    printf("Ano: %d\n", car.ano);
61  printf("Pre: %d\n", car.preco);
    }
63
    void imprimirTodosDados(struct carro car[], int tam){
65     int i;
        for (i = 0; i < tam; i++){
67         imprimirDados(car[i]);
        }
69     }

71 void modificarDados(struct carro car[], int tam){
    int i;
73     for (i = 0; i < tam; i++){
        if (strcmp(car[i].fabricante, "Chevrolet") == 0){
75         strcpy(car[i].fabricante, "GM");
        }
77     }
    }
```

Listing 1: Resposta do exercício 1 codificado na linguagem C

**Resposta pergunta 2:**

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

4 #define TAM 10

6 struct filme{
    char nome[20];
8    char diretor[20];
    char genero[20];
10   int ano;
    };
12

    void cadastrarFilme(struct filme *movie);
14 void imprimirFilme(struct filme movie);
    void imprimirTodosFilmes(struct filme movie[], int tam);
16 void buscarFilmesGenero(struct filme movie[], int tam);

18 int main(){
```

```
printf("——Filmes——\n\n");
20 struct filme movies[TAM];
int i;
22 for (i = 0; i < TAM; i++){
    printf("    Cadastrar Filme: \n");
24    cadastrarFilme(&movies[i]);
}
26 printf("    Imprimir Filmes: \n");
imprimirTodosFilmes(movies, TAM);
28 printf("    Buscar Filmes Aventura (2001 a 2005): \n");
buscarFilmesGenero(movies, TAM);
30 return 0;
}

32 void cadastrarFilme(struct filme *movie){
34     char nome[20];
    scanf("%s", nome);
36     strcpy((*movie).nome, nome);

38     char diretor[20];
    scanf("%s", diretor);
40     strcpy((*movie).diretor, diretor);

42     char genero[20];
    scanf("%s", genero);
44     strcpy((*movie).genero, genero);

46     int ano;
    scanf("%d", &ano);
48     (*movie).ano = ano;
}

50 void imprimirFilme(struct filme movie){
52     printf("\n");
    printf("Nome: %s\n", movie.nome);
54     printf("Diretor: %s\n", movie.diretor);
    printf("Genero: %s\n", movie.genero);
56     printf("Ano: %d\n", movie.ano);
}

58 void imprimirTodosFilmes(struct filme movies[], int tam){
60     int i;
    for (i = 0; i < tam; i++){
62         imprimirFilme(movies[i]);
    }
64 }

66 void buscarFilmesGenero(struct filme movies[], int tam){
    int i;
68     for (i = 0; i < tam; i++){
        if (strcmp(movies[i].genero, "aventura") == 0){
70             if (movies[i].ano >= 2001 && movies[i].ano <= 2005){
                imprimirFilme(movies[i]);
72             }
        }
    }
}
```

```
74     }  
    }  
}
```

Listing 2: Resposta do exercício 2 codificado na linguagem C

**Resposta pergunta 3:**

```
1  #include <stdio.h>  
2  #include <stdlib.h>  
3  #include <string.h>  
  
5  #define TAM 100  
  
7  typedef struct {  
8      char nome[30];  
9      char cargo[30];  
10     int numero;  
11     int idade;  
12     int telefone;  
13     float salario;  
14     short usado;  
15 } regFuncionario;  
  
17 void imprimeMenu() {  
18     printf("0 - Sair;\n");  
19     printf("1 - Cadastrar funcionario;\n");  
20     printf("2 - Imprimir todos funcionarios;\n");  
21     printf("3 - Buscar pelo nome;\n");  
22     printf("4 - Buscar pelo numero;\n");  
23     printf("5 - Remover funcionario;\n");  
24     printf("6 - Editar funcionario;\n");  
25 }  
  
27 void iniciaCadastro(regFuncionario vet[]) {  
28     for(int i = 0; i < TAM; i++) {  
29         vet[i].usado = 0;  
30     }  
31 }  
  
33 void lefuncionario(regFuncionario *a) {  
34     printf("Nome:\n");  
35     scanf("%s", (*a).nome);  
36     getchar();  
37     printf("Cargo:\n");  
38     scanf("%s", (*a).cargo);  
39     getchar();  
40     printf("Numero:\n");  
41     scanf("%d", &(*a).numero);  
42     getchar();  
43     printf("Idade:\n");  
44     scanf("%d", &(*a).idade);  
45     getchar();  
46     printf("Telefone:\n");  
47     scanf("%d", &(*a).telefone);
```

```
49     getchar();
    printf("Salario\n");
    scanf("%f", &(*a).salario);
51     getchar();
}

53
int cadastrafuncionario(regFuncionario vet[], regFuncionario a) {
55     int i;
    for(i = 0; i < TAM; i++) {
57         if(vet[i].usado == 0) {
            vet[i] = a;
59             vet[i].usado = 1;
            return 1;
61         }
    }
63     return 0;
}

65
void edit(regFuncionario vet[], regFuncionario a, int i) {
67     vet[i] = a;
}

69
int buscarfuncionario_cod(regFuncionario vet[], int numero) {
71     for(int i=0; i<TAM; i++) {
        if(vet[i].numero == numero) {
73             return i;
        }
75     }
    return -1;
77 }

79
int buscarfuncionario_nome(regFuncionario vet[], char *nome) {
    for(int i=0; i<TAM; i++) {
81         if(!strcmp(vet[i].nome, nome)) {
            return i;
83         }
    }
85     return -1;
}

87
void removefuncionario(regFuncionario vet[], int posicao) {
89     for(int i = posicao; i < (TAM - 1); i++) {
        vet[i] = vet[i+1];
91     }
}

93
void imprimefuncionarios(regFuncionario vet[]) {
95     int cont=0;
    for(int i = 0; i < TAM; i++) {
97         if(vet[i].usado == 1) {
            printf("%d: %-6d %s ", i, vet[i].numero, vet[i].nome);
99             printf("%-6d %s ", vet[i].idade, vet[i].cargo);
            printf("%-6d %.2lf \n", vet[i].telefone, vet[i].salario);
101         } else if(vet[i].usado == 0) {
```

```

    cont = cont + 1;
103     }
    }
105     if (cont==TAM) {
        printf("Erro: nenhum funcionario cadastrado.\n");
107     }
}

109 void imprimeUm(regFuncionario vet[], int i){
111     if (i > -1) {
        if(vet[i].usado == 1) {
113             printf("%d: %-6d %-30s ", i, vet[i].numero, vet[i].nome);
            printf("%d: %-6d %-30s ", i, vet[i].idade, vet[i].cargo);
115             printf("%d: %-6d %lf\n", i, vet[i].telefone, vet[i].salario);
        }
117     } else {
        printf("Nao encontrado\n");
119     }
}

121

123 int main() {
    int op=1;
125     regFuncionario nfuncionario[TAM];
    regFuncionario a;
127     iniciaCadastro(nfuncionario);
    do {
129         imprimeMenu();
        scanf("%d", &op);
131         getchar();
        if(op == 1) {
133             lefuncionario(&a);
            cadastrafuncionario(nfuncionario, a);
135         } else if (op == 2) {
            imprimefuncionarios(nfuncionario);
137         } else if (op == 3) {
            char nome[30];
139             printf("Digite o nome a buscar:\n");
            scanf("%[^\\n]", nome);
141             getchar();
            int pos = buscarfuncionario_nome(nfuncionario, nome);
143             imprimeUm(nfuncionario, pos);
        } else if (op == 4) {
145             int num;
            printf("Digite o numero a buscar:\n");
147             scanf("%d", &num);
            getchar();
149             int pos = buscarfuncionario_cod(nfuncionario, num);
            imprimeUm(nfuncionario, pos);
151         } else if (op == 5) {
            char nome[30];
153             printf("Digite o nome a buscar para remover:\n");
            scanf("%[^\\n]", nome);
155             getchar();
        }
    } while (op != 0);
}
```

```
157         int pos = buscarfuncionario_nome(nfuncionario , nome);
        imprimeUm(nfuncionario , pos);
        removefuncionario(nfuncionario , pos);
159         printf("Registro acima removido\n");
    } else if (op == 6) {
161         char nome[30];
        printf("Digite o nome a buscar para editar:\n");
163         scanf("%s", nome);
        getchar();
165         int pos = buscarfuncionario_nome(nfuncionario , nome);
        if (pos > -1) {
167             imprimeUm(nfuncionario , pos);
            printf("Iniciando edicao\n");
169             lefuncionario(&a);
            edit(nfuncionario , a, pos);
171             printf("Registro acima editado. Atual: \n");
            imprimeUm(nfuncionario , pos);
173         }
    }
175 } while (op != 0);
    return 0;
177 }
```

Listing 3: Resposta do exercício 3 codificado na linguagem C

**Resposta pergunta 8:**

```
1 #include <stdio.h>

3 #define principal main
  #define inicio {
5  #define fim }
  #define imprima printf
7  #define retorno return
  #define se if
9  #define senao }else{
  #define igual ==
11 #define mais +
  #define menos -
13 #define vezes *
  #define dividido /

15 typedef int inteiro;

17 inteiro fatorial(inteiro n);

19 inteiro principal()
21 inicio
    inteiro n = 5;
23     inteiro fat = fatorial(n);
    imprima("Fatorial(%d) = %d", n, fat);
25     retorno 0;
fim
27 inteiro fatorial(inteiro n)
```

```
29 inicio
   se (n igual 1) inicio
31     retorno 1;
   senao
33     retorno n vezes fatorial(n menos 1);
   fim
35 fim
```

Listing 4: Resposta do exercício 8 codificado na linguagem C