



Pontifícia Universidade Católica de Minas Gerais
Curso de Ciência da Computação
Disciplina: Algoritmos e Estruturas de Dados I
Prof. Felipe Cunha

Laboratório 10

```
1  /* Escreva um programa que leia um vetor G de 20 caracteres que representa o gabarito de uma
2     prova. A seguir, para cada um dos alunos de uma turma, leia o vetor de respostas (R) e
3     conte o número de acertos. Mostre a nota do aluno e uma mensagem APROVADO, se a nota for
4     maior ou igual a 6 ou REPROVADO, caso contrário. Cada questão correta vale 0,5 ponto. */
5
6  #include <stdio.h>
7  #include <stdlib.h>
8
9  #define TAM 20
10
11 int main(void)
12 {
13     char gabarito[TAM];
14     char respostas[5][TAM];
15     int nota = 0;
16     int resultado[5];
17
18     for (int i = 0; i < TAM; i++){
19         ??printf("Entre com o gabarito da questao [%d]:", i + 1);
20         scanf(" %c", &gabarito[i]);
21         system("cls");
22     }
23
24     for (int i = 0; i < 5; i++){
25         ??printf("\n*****\n");
26         printf("                RESPOSTA ALUNO %d\n", i + 1);
27         nota = 0;
28         for (int j = 0; j < TAM; j++){
29             ??printf("Entre com a alternativa da questao [%d]:", j + 1);
30             scanf(" %c", &respostas[i][j]);
31             if (gabarito[j] == respostas[i][j])
32                 nota++;
33         }
34         resultado[i] = nota;
35     }
36
37     printf("\n*****");
38     printf("\n*                Resultados");
39     printf("\n*****");
40     for (int i = 0; i < 5; i++){
41         if (resultado[i] * 0.5 >= 6){
42             printf("\nCandidato %d Aprovado!!", i + 1);
43         } else {
44             printf("\nCandidato %d Reprovado!!", i + 1);
45         }
46     }
47
48     printf("\n\n*****");
49     printf("\n*                GABARITO");
50     printf("\n*****");
51     for (int i = 0; i < TAM; i++)
52         printf("\n[%d]:%c", i + 1, gabarito[i]);
53
54     return 0;
55 }
```

```

1  /*
2     Faça um programa em C para ler e manipular informações de um cadastro de automóveis. São
3     necessárias duas estruturas: carro e proprietario. Todos os dados de um carro devem ser
4     lidos e depois impressos. Os campos das estruturas são:
5     carro: modelo(char), ano(int), placa (char), proprietario (proprietario);
6     proprietario: cpf(long int), nome(char).
7  */
8
9  #include<stdlib.h>
10 #include<stdio.h>
11
12 typedef struct Proprietario{
13     char nome[20];
14     long int cpf;
15 }Proprietario;
16
17 typedef struct Carro{
18     char modelo[15];
19     int ano;
20     char placa[10];
21     Proprietario proprietario;
22 }Carro;
23
24 int main(){
25     Carro c1;
26     printf("Cadastro de Automoveis\n");
27     printf("Digite o nome do proprietario:\n");
28     scanf("%s", c1.proprietario.nome);
29     printf("Digite o CPF do proprietario:\n");
30     scanf(" %ld", &c1.proprietario.cpf);
31     printf("Digite o modelo do carro:\n");
32     scanf("%s", c1.modelo);
33     printf("Digite o ano do carro:\n");
34     scanf(" %d", &c1.ano);
35     printf("Digite a placa do carro:\n");
36     scanf("%s", c1.placa);
37
38     printf("Cadastro de Automoveis\n");
39     printf("Nome do proprietario: %s\n", c1.proprietario.nome);
40     printf("CPF do proprietario: %ld\n", c1.proprietario.cpf);
41     printf("Modelo do carro: %s\n", c1.modelo);
42     printf("Ano do carro: %d\n", c1.ano);
43     printf("Placa do carro: %s\n", c1.placa);
44
45     return 0;
46 }

```

```
1 /*Faça um programa que receba dois arquivos do usuário, e crie um terceiro arquivo com o
   conteúdo dos dois primeiros juntos (o conteúdo do primeiro seguido do conteúdo do segundo
   ).*/
2
3 #include <stdio.h>
4 #include <stdlib.h>
5 #include <string.h>
6
7 int main(){
8     char facilitador[200], nomeArquivo01[20],
9         nomeArquivo02[20];
10
11     printf("\n Insira o nome do primeiro arquivo: ");
12     scanf("%s", nomeArquivo01);
13
14     printf("\n Insira o nome do segundo arquivo: ");
15     scanf("%s", nomeArquivo02);
16
17     FILE *arquivo;
18     FILE *junto;
19
20     arquivo = fopen(nomeArquivo01, "r");
21     junto = fopen("junto.txt", "w");
22     while(!feof(arquivo)){
23         fgets(facilitador, 200, arquivo);
24         fprintf(junto,"%s", facilitador);
25     }
26     fclose(arquivo);
27
28     arquivo = fopen(nomeArquivo02, "r");
29     while(!feof(arquivo)){
30         fgets(facilitador, 200, arquivo);
31         fprintf(junto,"%s\n", facilitador);
32     }
33     fclose(arquivo);
34     fclose(junto);
35
36     return 0;
37 }
38
```

```
1  /*
2  Faça um programa que leia um vetor com os dados de 5 carros: marca (maximo 15 letras), ano e
   preço. Leia um valor p e mostre as informações de todos os carros com preço menor que p.
   Repita este processo ate que seja lido um valor p = 0.
3  */
4
5
6  typedef struct Carro{
7      char marca[15];
8      int ano;
9      float preco;
10 } Carro;
11
12 int main(){
13     Carro vetor[5];
14
15     for(int i = 0; i < 5; i++){
16         printf("Entre com a marca: ");
17         scanf("%s", vetor[i].marca);
18         printf("Entre com o ano: ");
19         scanf(" %d", &vetor[i].ano);
20         printf("Entre com o preço: ");
21         scanf(" %f", &vetor[i].preco);
22     }
23
24     float preco = 0;
25     printf("Entre com o preço a ser pesquisado: ");
26     scanf(" %f", &preco);
27
28     while(preco != 0){
29         system("clear");
30         for (int i = 0; i < 5; i++){
31             if(vetor[i].preco < preco){
32                 printf("Marca %s \t", vetor[i].marca);
33                 printf("Entre com o ano %d \t", vetor[i].ano);
34                 printf("Entre com o preço: %f \n", vetor[i].preco);
35             }
36         }
37         printf("Entre com o preço a ser pesquisado: ");
38         scanf(" %f", &preco);
39     }
40
41     return 0;
42 }
43
```