



3a Prova Teórica

Nome: _____ Matrícula: _____ Turma: _____ Data: _____

Instruções Básicas:

- As respostas devem ser marcadas a caneta.
- As questões rasuradas serão anuladas.
- O caderno de código é parte integrante da prova.
- A prova não deve ser desgrampeada.

1. Considere que o código *prova3b.c* e o arquivo texto *B.txt* estejam na mesma pasta. Considere ainda, que este código seja executado por seis vezes, sendo que em cada uma delas a string literal "MODE" é substituída respectivamente por: r, w, a, r+, w+ e a+. Escolha a opção que apresenta a sequência correta, resultante das referidas execuções do software: (2 pontos)

- (a) 39 0 0 39 0 0
- (b) 0 39 0 0 39 0
- (c) 39 39 39 39 39 39
- (d) 0 0 0 0 0 0
- (e) 39 0 0 39 39 39
- ~~(f) 39 0 0 39 0 39~~

2. O código *prova3a.c* apresenta algumas lacunas. Dentre elas está a falta da definição da função de imprimir filmes. Considere as propostas de implementação desta função em *prova3aImprimir* e escolha a correta (2 pontos):

- (a) A
- ~~(b) B~~
- (c) C
- (d) D
- (e) E
- (f) F
- (g) Nenhuma das opções.

3. O código *prova3c.c* possui um falha não evidenciada em tempo de compilação. Escolha a opção que determina essa falha: (2 pontos)

- (a) A variável *myFile* deveria ser declarada como tipo *FILE* e não como ponteiro para *FILE*.
- (b) Mesmo que a função *fopen* não consiga abrir o arquivo, a estrutura de repetição será executada.
- (c) Dependendo do conteúdo do arquivo texto o loop de *while(1)* pode se tornar infinito.
- ~~(d) O código abre o arquivo e não o fecha, podendo incorrer em perda das mudanças realizadas.~~
- (e) Nenhuma afirmação está correta.
- (f) Todas as afirmações estão erradas.

4. O código *prova3a.c* apresenta algumas lacunas. Dentre elas está a falta da definição da função para incluir filme. Considere as propostas de implementação desta função em *prova3aIncluir* e escolha a correta (2 pontos):

- (a) A
- (b) B
- (c) C
- (d) D
- (e) E
- ~~(f) F~~
- (g) Nenhuma das opções.

5. Informe no espaço a seguir a(s) possibilidade(s) de substituição do termo *MODE* da linha 79 do código *prova3a.c*, para que a função *ImprimirFilmes* apresente todos os registros do catálogo. (2 pontos):

r, r+, a+

```

1  #include <stdio.h>
2  #include <string.h>
3
4
5  struct Pessoa
6  {
7      char nome[30];
8      char sobrenome[30];
9  };
10
11 struct Filme
12 {
13     char titulo[200];
14     int ano;
15     struct Pessoa diretor;
16 };
17
18
19 void removeQuebraLinha(char * texto)
20 {
21     unsigned int len;
22
23     len = strlen(texto);
24     if (texto[len - 1] == '\n')
25         texto[len - 1] = '\0';
26 }
27
28 int main ()
29 {
30     int contador, indice;
31     char num_str[10];
32     struct Pessoa diretor1;
33     struct Filme filme1;
34     FILE * catalogo_permanente;
35
36     catalogo_permanente = fopen("B.dat", "a+");
37
38     if (catalogo_permanente == NULL)
39     {
40         perror("Falha na abertura do arquivo.");
41         return 0;
42     }
43
44     puts("Quantos filmes deseja adicionar ao catalogo?");
45
46     ///Trecho de codigo necessario para driblar problemas de leitura de string com scanf
47
48     ///Com funcao fgets obtem-se a string num_str pela entrada padrao
49     fgets(num_str, sizeof(num_str), stdin)
50
51     ///A funcao atoi converte a string num_str para int
52     contador = atoi(num_str);
53
54     for (indice = 0; indice < contador; indice++)
55     {
56         puts("Informe o titulo do filme:");
57         fgets(filme1.titulo, sizeof(filme1.titulo), stdin);
58         removeQuebraLinha(filme1.titulo);
59
60         puts("Informe o ano do filme:");
61         fgets(num_str, sizeof(num_str), stdin);
62         filme1.ano = atoi(num_str);
63
64         puts("Informe o nome do diretor:");
65         fgets(filme1.diretor.nome, sizeof(filme1.diretor.nome), stdin);
66         removeQuebraLinha(filme1.diretor.nome);
67
68         puts("Informe o sobrenome do diretor:");
69         fgets(filme1.diretor.sobrenome, sizeof(filme1.diretor.sobrenome), stdin);
70
71         if (IncluirFilme(&filme1, catalogo_permanente) > 0)
72         {
73             printf("Filme: %s adicionado com sucesso\n", filme1.titulo);
74         }
75     }
76
77     fclose(catalogo_permanente);

```

```

78
79     catalogo_permanente = fopen("/tmp/test/A.dat",MODE);
80
81     ImprimirFilmes(catalogo_permanente);
82
83     fclose(catalogo_permanente);
84
85     return 0;
86 }

```

(prova3b.c)

```

1  #include <stdio.h>
2
3  int main ()
4  {
5      FILE * myFile;
6      int loops = 0;
7      char ch;
8
9      myFile = fopen("B.txt", "MODE");
10
11      if (myFile == NULL)
12      {
13          puts("falha_a_tentar_abrir_o_arquivo");
14      } else
15
16      while (1)
17      {
18          ch = fgetc(myFile);
19          if (ch == EOF) break;
20          else loops+=1;
21      }
22
23      printf("%d",loops);
24
25      fclose(myFile);
26
27      return 0;
28 }

```

(B.txt)

```

1  He's_making_a_list,_Checking_it_twice;

```

(prova3aImprimir)

A	<pre> void ImprimirFilmes(FILE * fileMovies) { struct Filme filme; fscanf(fileMovies, "%s%d%s", filme.titulo, filme.ano, filme.diretor.nome, filme.diretor.sobrenome) != EOF) printf("%s%d%s\n", filme.titulo, filme.ano, filme.diretor.nome, filme.diretor.sobrenome); } </pre>
B	<pre> void ImprimirFilmes(FILE * fileMovies) { struct Filme filme; while (fscanf(fileMovies, "%s%d%s", filme.titulo, &filme.ano, filme.diretor.nome, filme.diretor.sobrenome) != EOF) { printf("%s%d%s\n", filme.titulo, filme.ano, filme.diretor.nome, filme.diretor.sobrenome); } } </pre>
C	<pre> void ImprimirFilmes(FILE * fileMovies) { struct Filme filme; fscanf(fileMovies, "%s%d%s", filme.titulo, &filme.ano, filme.diretor.nome, filme.diretor.sobrenome) != EOF) printf("%s%d%s\n", filme.titulo, filme.ano, filme.diretor.nome, filme.diretor.sobrenome); } </pre>
D	<pre> void ImprimirFilmes(FILE * fileMovies) { struct Filme filme; while (fscanf(&fileMovies, "%s%d%s", filme.titulo, &filme.ano, filme.diretor.nome, filme.diretor.sobrenome) != EOF) { printf("%s%d%s\n", filme.titulo, filme.ano, filme.diretor.nome, filme.diretor.sobrenome); } } </pre>
E	<pre> void ImprimirFilmes(FILE * fileMovies) { struct Filme filme; while (fscanf(fileMovies, "%s%d%s", filme->titulo, &filme->ano, filme->diretor.nome, filme->diretor.sobrenome) != EOF) { printf("%s%d%s\n", filme.titulo, filme.ano, filme.diretor.nome, filme.diretor.sobrenome); } } </pre>
F	<pre> void ImprimirFilmes(FILE * fileMovies) { struct Filme filme; while (fscanf(&fileMovies, "%s%d%s", filme->titulo, &filme->ano, filme->diretor.nome, filme->diretor.sobrenome) != EOF) { printf("%s%d%s\n", filme.titulo, filme.ano, filme.diretor.nome, filme.diretor.sobrenome); } } </pre>