Código	RunCodes	Observação	Código	Nota
13692400	10			0
13692417	10		10	10
4818232	10	Escrita ok (não entendi o strlen()+1 no fwrite da string de autor) mas para a impressão utilizou os dados que estavam em memória principal ao invés de ler do arquivo.	6	8
13692309	10		10	10
12563690	10		10	10
10677290	10	Sugestão: os byteoffsets poderiam ser pegos com o ftell no começo da execução de cada chamada de carregarLivro(), ao invés de calcular todos de uma vez com a função getPositions() e depois ler os livros.	10	10
13692362	10	Leitura do arquivo utiliza informações sobre comprimento das strings salvas em memória principal, ao invés de utilizar delimitador e indicador de comprimento do arquivo.	7	8,5
13750770	-		-	0
13733579	10		10	10
4822220	0	Escrita em arquivo fugiu da organização proposta (fwrite da struct inteira), impressão utilizou os dados que estavam em memória principal ao invés de ler do arquivo.	2	1
12682435	10	Sugestão: No geral, utilizamos os arquivos .h mais para definições como assinaturas de funções, classes etc. e suas implementações ficam em um respectivo .c. Não é regra (inclusive existem as chamadas header-only libraries) mas é o mais usual.	10	10
12681733	10		10	10
11816098	10		10	10
13692289	10	O ideal seria saber o tamanho das strings através do arquivo, ao invés de salvar em um vetor memória principal. (esse é o objetivo de escrever o delimitador e o indicador de comprimento no arquivo)	7	8,5
10346940	-		-	0

12725515	10	O ideal seria saber o tamanho das strings através do arquivo, ao invés de salvar em vetor em memória principal. (esse é o objetivo de escrever o delimitador e o indicador de comprimento no arquivo)	7	8,5
13696662	10	Escrita em arquivo ok. Leitura do arquivo utiliza informações sobre comprimento das strings salvos em memória principal, ao invés de utilizar delimitador e indicador de comprimento do arquivo.	7	8,5
13750784	-		-	0
13783972	0	Leitura das strings do stdin (scanf("%s")) para de ler no primeiro " ". Funções de escrita e leitura em arquivo ok. Calculo de byteoffset do primeiro registro a ser impresso assume que as strings tem tamanho 100 (tamanho máximo do buffer para ler do stdin, mas no arquivo é escrito o número certo de caracteres da string), Não calcula byteoffset para o resto dos registros que deveriam ser impressos. Aparentemente não compilou por causa do makefile. Rodando localmente, os resultados não batem com os casos de teste.	7	3,5
13692355	0			0
12676192	10	Sugestão: como a string de autor é a última e há um delimitador de registro, ele serve de delimitador pra string também, mas como temos o tamanho da string através do indicador de comprimento, daria pra ler somente com um fwrite, ao invés de usar a mesma lógica da string de titulo.	10	10
12725025	10	Leitura do arquivo utiliza informações sobre comprimento da string de titulo salvas em memória principal, ao invés de utilizar delimitador escrito no arquivo.	8	9
12543287	10	Escrita em arquivo ignora indicador de comprimento da string autor. Impressão dos registros, utiliza dados salvos na memória principal durante a leitura do stdin, não há leitura do arquivo.	6	8
13861176	0	Escrita em arquivo ok. Leitura do arquivo não lê os dados corretamente (não entendi os fseeks).	6	3

11781587	10		10	10
13727485	10		10	10
13692380	10	Sugestões: Normalmente, para os nomes das funções também são utilizadas convenções como camelCase e snake_case, para facilitar a leitura. Função de leitura do arquivo daria pra ser mais simples (usando uma lógica pra posicionar o cursor do arquivo no primeiro registro que deveria ser impresso e a partir daí fazer algo parecido com a escrita (mas com freads), só tomando cuidado com a leitura de titulo que tem que fazer a lógica do delimitador).	10	10
13717560	10		10	10
11372883	0	Código não compila e não está completo, apesar de estar bem modularizado, escrita em arquivo aparentemente ok, falta parte de leitura de arquivo, estava no caminho correto (mas lógica de imprimir os registros iria imprimir os M últimos, mas na ordem inversa aos casos de teste).	6	3
12547830	0	Código da falha de segmentação. Escrita ok (a escrita do delimitador de registro fwrite("-1", sizeof (char), 1), só escreve o "-", se fosse utilizar o delimitador sugerido no enunciado, deveria usar o -1 e não "-1"). Leitura de arquivo lê 1000 bytes e tenta copiar os dados a partir desse buffer para os campos do livro, daria pra só ir lendo os dados do registro direto do arquivo (separadamente).	6	3
11858651	10		10	10
13828592	10	O ideal seria saber o tamanho das strings através do arquivo, ao invés de salvar em vetor em memória principal. (esse é o objetivo de escrever o delimitador e o indicador de comprimento no arquivo)	8	9
13696641	10	Leitura de titulo utilizou o tamanho da string salvo em memória principal e não o delimitador, leitura de autor também, chegou a ler o tamanho da string do arquivo mas utilizou o vetor de tamanhos (salvo em memória principal durante a leitura do stdin) no fread.	7	8,5

10727952	10		10	10
13672922	10		10	10
11838777	8	Resultado aparentemente corretos, não deve estar passando nos casos de teste por casusa de alguma formatação ('\n'). Sugestão: nome das variáveis podem ser mais significativos.	10	9
13692334	10		10	10
13838346	10		10	10
13717786	10		10	10
13730541	10		10	10
13692272	10	A escrita do delimitador de registro fwrite("-1", sizeof (char), 1), só escreve o "-", se fosse utilizar o delimitador sugerido no enunciado, deveria usar o -1 e não "-1"	10	10
11295810	10		10	10
11234061	10		10	10
12675020	10		10	10
13692341	10	Sugestão: como a string de autor é a última e há um delimitador de registro, ele serve de delimitador pra string também, mas como temos o tamanho da string através do indicador de comprimento, daria pra ler somente com um fwrite, ao invés de usar a mesma lógica da string de titulo.	10	10
12563814	-		-	0
11918539	0	Código não compila. Escrita ok. Na leitura do arquivo lê o autor duas vezes (só o primeiro fscanf já era suficiente). O fseek funciona com bytes, então se você der um fseek passando a quantidade de registros que você quer passar, não vai funcionar, ele não sabe quantos bytes há cada registro, você teria que ter alguma lógica para passar o número certo de bytes que o registro ocupa. Não compilou por causa de warning, rodando localmente o resultado não bate com os casos de teste.	7	3,5
11821282	-		-	0
13696658	10		10	10
9793502	0	Não utilizou arquivos.	0	0

13692421	10	Sugestão: como a string de autor é a última e há um delimitador de registro, ele serve de delimitador pra string também, mas como temos o tamanho da string através do indicador de comprimento, daria pra ler somente com um fwrite, ao invés de usar a mesma lógica da string de titulo.	10	10
13732352	10		10	10
13750791	10		10	10
13716301	10	Leitura do arquivo aparentemente não utilzia das técnicas de registro de tamanho variável (delimitador e indicador de comprimento).	7	8,5
10492012	10	Sugestão: como a string de autor é a última e há um delimitador de registro, ele serve de delimitador pra string também, mas como temos o tamanho da string através do indicador de comprimento, daria pra ler somente com um fwrite, ao invés de usar a mesma lógica da string de titulo.	10	10
11833236	10		10	10
13783714	10		10	10
12567502	-		-	0
13692438	10		10	10
11871181	10	O ideal seria saber o tamanho das strings através do arquivo, ao invés de salvar em vetor em memória principal. (esse é o objetivo de escrever o delimitador e o indicador de comprimento no arquivo)	7	8,5
13731066	10	Leitura do arquivo utiliza informações sobre comprimento das strings salvas em memória principal, ao invés de utilizar delimitador e indicador de comprimento do arquivo.	7	8,5
11794824	-		-	0
12544570	10		10	10
11819240	0	Escrita no arquivo aparentemente ok. Não há leitura dos campos dos registros seguindo a organização proposta.	5	2,5
12691710	-		-	0
13673242	10		10	10
13692313	10		10	10