

Importante: construa o seu código exatamente de acordo com os enunciados.

Regras para a prova: * utilize apenas os comandos ANSI C vistos em sala de aula; * para manipulação de arquivos utilizar apenas fopen, fclose, fread, fgets, fscanf; * na construção da prova, utilizar apenas os arquivos de dados fornecidos pelo Professor, SEM modificações; * para construir a resposta, utilize somente a interface do TAD Lista; * todas as figuras exibem apenas exemplos para auxiliar na interpretação do enunciado * Professor pode solicitar que o aluno explique a sua solução da avaliação 2.

Questão Única

Crie um arquivo chamado **resposta.c** contendo o seu código resposta para a questão. Neste arquivo, construa as funções e tipos de acordo com os enunciados a seguir.

a) **Criar os tipos t_projeto, t_membro, t_relacao:** usam structs para modelar uma linha de cada uma das tabelas fornecidas como material de prova (respectivamente tabprojetos, tabmembros, tabrelacao, figuras 1.a, 1.b, 1.c). As identificações de cada coluna de cada tabela estão descritos nas primeiras linhas de cada arquivo. Todos os campos são strings de no máximo 128 chars.

b) **Lista load_projetos(char *fileName):** lê arquivos (nome qualquer) do tipo bdprojetos.txt e transforma cada linha em um dado do tipo t_projeto, em seguida armazena esse dado em uma Lista. Ao final, retorna a Lista para o chamador da função.

c) **Lista load_membros(char *fileName):** lê arquivos (nome qualquer) do tipo bdmembross.txt e transforma cada linha em um dado do tipo t_membro, em seguida armazena esse dado em uma Lista. Ao final, retorna a Lista para o chamador da função.

d) **Lista load_relacoes(char *fileName):** lê arquivos (nome qualquer) do tipo bdrelacoes.txt e transforma cada linha em um dado do tipo t_relacao, em seguida armazena esse dado em uma Lista. Ao final, retorna a Lista para o chamador da função.

e) **função Lista membros_projeto(char *id_proj, Lista lst_relacoes, Lista lst_membros):** recebe 2 Listas contendo as relações e os membros, respectivamente, mais um id de um projeto qualquer. Retorna uma Lista com todos os membros (do tipo t_membro) do projeto de id igual a id_proj. Utilize a lst_relacoes para associar projetos a membros.

f) **função Lista projetos_membro(char *id_mem, Lista lst_relacoes, Lista lst_projetos):** recebe 2 Listas contendo as relações e os projetos, respectivamente, mais um id de um membro qualquer. Retorna uma Lista com todos os projetos (do tipo t_projeto) que o membro de id igual a id_mem participou. Utilize a lst_relacoes para associar projetos a membros..

d) **int main():** A função deve:

- criar as Listas relacionadas aos conteúdos dos arquivos bdmembross.txt, bdprojetos.txt e bdrelacoes.txt.
- Exibir o menu da figura 1.d. A escolha 1 pede o id do membro e lista todos os projetos que ele participou (1 projeto por linha). A escolha 2 pede o id do projeto e lista todos os membros que participaram do projeto (1 por linha).
- O programa deve reexibir o menu e continuar nessa dinâmica até ser escolhida a opção 3-Sair.

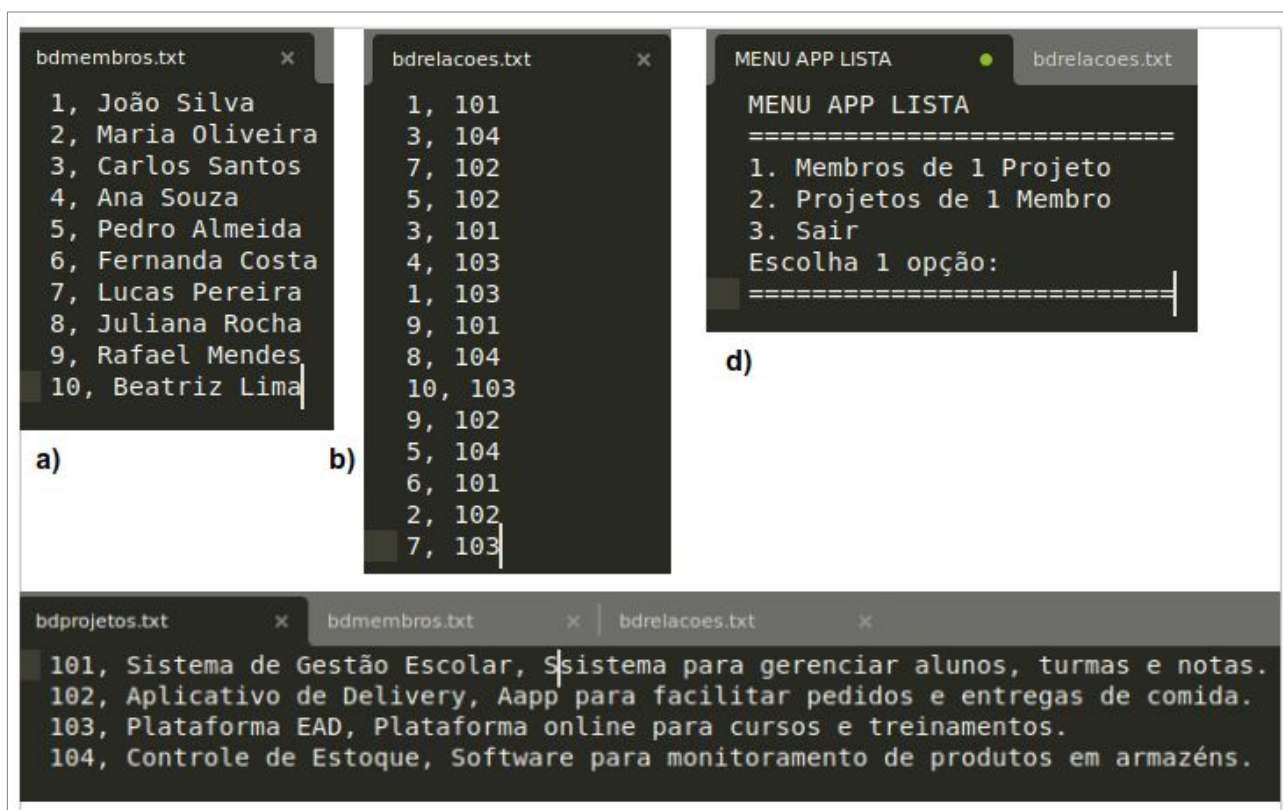


Figura 1

Entrega

Comprima o arquivo **resposta.c**, mais todos os arquivos de dados **bdxxxxx.txt**, em um arquivo zip chamado **<sua matricula>.zip**. Em seguida envie este arquivo zip para a tarefa associada a avaliação 2, no AVA.

Boa prova!