

Projeto avaliação P1

Você foi contratado por um banco para desenvolver um programa de controle de contas correntes e contas poupança. Para tal, defina em seu programa uma estrutura que contenha os dados para o controle dessas contas bancárias. Esta estrutura deverá se chamar “Conta”, e deverá possuir os seguintes campos para os dados:

- Nº conta (id);
- Nome do Cliente;
- Tipo da conta;
- Saldo da conta.

Você deve utilizar o comando para criar um **novo tipo**, e com este **novo tipo** que se chamará “Conta”, declarar as novas variáveis estruturas desse novo tipo.

Na função main(), colete via teclado um valor “**n**” passado pelo usuário, **que será a quantidade de contas bancárias gerenciadas pelo seu software, ou seja, a quantidade de elementos que estarão no vetor**. A partir deste valor coletado, crie uma função que **aloque dinamicamente** um bloco de memória para abarcar estes “**n**” elementos, e devolva para o main() o endereço deste bloco alocado. Este bloco será usado como o vetor dinâmico e conterá a quantidade de estruturas “Conta”, definido no passo anterior. **(NÃO SERÁ ACEITO DECLARAÇÃO DE VETOR CONVENCIONAL, SOMENTE ALOCAÇÃO DE MEMÓRIA)**

Crie uma função específica e independente para coletar dados das contas bancárias e alimentar o vetor de estruturas dinâmico. Esta função **não receberá nada como parâmetro**, mas devolverá **uma estrutura completa e preenchida (devolverá apenas uma por vez)**, para o vetor de estruturas alocado dinamicamente. **Toda a inserção de dados dos clientes no vetor alocado dinamicamente, deve ser feita somente por esta função. Os dados pertencentes aos Clientes não poderão ser inseridos por qualquer outra função do seu programa. Isso deve ser feito exclusivamente por esta função de coleta.**

Após o preenchimento de todo o vetor dinâmico, crie uma função ou procedimento que receba este vetor e a quantidade de elementos que ele possui. Esta função imprimirá todas as “n” estruturas deste vetor alocado dinamicamente. Lembre-se de deixar visível (pausa) para o usuário visualizar o resultado da impressão.

Salve seu vetor de dados em um arquivo **binário (NÃO SERÃO ACEITOS DADOS SALVOS EM ARQUIVO DE TEXTO PURO)** e **tome o devido cuidado para não perder dados** quando encerrar seu programa.

Crie mais uma **função** ou procedimento, que receba um ponteiro para o vetor dinâmico de estruturas, e libere este bloco de memória alocado.

Obs.: Utilize apenas uma folha de codificação (*new->empty file*), ou seja, apenas um arquivo “**.C**”. **Não crie um projeto!**

Em resumo, a função principal *main()* gerenciará o programa, e então deverá:

- Disparar a função que cria o vetor dinâmico de estruturas alocando memória para isso;
- Chamar a função para a inserção de dados no vetor dinâmico, e esta devolverá, **sem receber nenhum parâmetro**, uma estrutura pronta e preenchida para ser inserida no vetor alocado dinamicamente;
- Chamar a função de impressão para exibição dos elementos, **passando como referência um ponteiro para o bloco de memória (o vetor)**, alocado dinamicamente e a **quantidade de elementos** que este possui;
- Salvar em arquivo, **em modo binário**, os dados inseridos no vetor alocado dinamicamente, não é necessário a recuperação e visualização dos dados. Se você quiser confirmar se os dados estão salvos, utilize a opção **Open with hex editor** no menu **File** da ide Code::Blocks.
- Chamar a função de liberação de memória liberando o vetor alocado, e encerrar o programa.

Este projeto será realizado em duplas, no máximo, e poderá ser entregue por apenas um dos integrantes da dupla.

Serão avaliados a implementação de todos os requisitos solicitados, clareza na codificação e comentários sobre o funcionamento do código (não exagere neste quesito).

Entregue na plataforma Moodle/AVA zipado, todo o conteúdo (todos os arquivos), gerados pela IDE CodeBlocks, juntamente com o arquivo de dados gerados pela sua aplicação.

Na folha de codificação antes do seu código (na primeira linha), coloquem seus nomes e prontuários, como comentário usando este formato:

```
//nome: xxxxxxxxxxxxxx – prontuário: guxxxxxx – Turma: xx.
```

```
//nome: xxxxxxxxxxxxxx – prontuário: guxxxxxx – Turma: xx.
```

Para as cópias de projeto será atribuída a nota 0, tanto a cópia como o trabalho que serviu como fonte.

Tenha em mente o seguinte: Se você for bom o suficiente para copiar, modificando e maquiando o código de outra pessoa, isso é sinal que você tem uma boa capacidade de manipular o código e domínio sobre a linguagem C. Se você tem essa capacidade, não precisa copiar de ninguém, você tem total condições de criar o seu código por você mesmo!!

Mas se você não tem esse domínio, e ainda assim insistir nessa ideia, faça por sua conta e risco. Saiba que todos os códigos serão submetidos ao detector de plágio, que além de verificar possíveis cópias em todos os projetos enviados, também verificará códigos oriundos da internet.