

# Trabalhos Finais

Data de Apresentação: 11 e 12 de abril

Nº máximo de Componentes: 3

Valor: 3 pontos

## Trabalho 1

Crie uma estrutura para representar os dados de um aluno, com matrícula de tipo inteiro, nome de tipo string, endereço do tipo string e um vetor de disciplinas com as disciplinas. O tipo disciplina deve ser definido como uma estrutura com nome do tipo string, professor do tipo string e notas como sendo um vetor de 3 floats para armazenar as notas do aluno naquela disciplina. Com essa estrutura faça as seguintes operações:

(a) Preencha os dados de 5 alunos automaticamente via comandos, para que seja possível testar todas as funções mesmo sem cadastrar manualmente alguns alunos.

(b) Faça com que o programa apresente um menu através de uma função onde o usuário possa escolher entre as funções de cadastrar aluno, excluir aluno, alterar aluno, buscar aluno, listar todos os alunos, listar aluno de determinada disciplina, listar alunos aprovados e sair. Cada funcionalidade deve ser feita como uma função em separado (o usuário deverá poder escolher qualquer das opções qualquer quantidade de vezes. Somente ao escolher a opção sair o programa deve ser finalizado).

(c) Quando a opção cadastrar aluno for escolhida, leia todos os dados de um aluno e o inclua na primeira posição livre do vetor.

(d) Quando a opção alterar aluno for escolhida, o usuário deverá escolher a posição a alterar. O programa deverá mostrar os dados atuais e pedir para o usuário digitar novamente todos os dados para aquele aluno.

(e) Quando a opção buscar aluno for escolhida, o programa deverá solicitar ao usuário o nome a procurar. Após isso, o programa irá buscar, entre os alunos cadastrados, um com nome procurado (ou parte dele). Caso encontre, deverá mostrar todos os dados desse aluno. Caso contrário, deverá exibir a mensagem "Aluno não encontrado".

(f) Quando a opção listar todos os alunos for selecionada, o programa deve exibir todos os dados de todos os alunos cadastrados até o momento. Se nenhum aluno estiver cadastrado, deverá exibir a mensagem "Nenhum funcionário cadastrado".

(g) Quando a opção listar alunos de determinada disciplina for selecionada, o programa deve solicitar ao usuário o nome de uma disciplina e procurar e mostrar, entre os alunos cadastrados, os que estejam naquela disciplina. Liste todos os alunos com as respectivas notas, médias e resultados (AM para média  $\geq 6$ , RN para média  $< 6$ ). Caso não encontre nenhum mostrar a mensagem "Nenhum aluno encontrado para a disciplina procurada".

(h) Quando a opção listar alunos aprovados, o programa deverá procurar os alunos cadastrados que possuem como média (soma das 3 notas na disciplina dividida por 3) um valor maior ou igual a 6. Caso não seja encontrado nenhum aluno, mostrar a seguinte mensagem: "Nenhum aluno aprovado encontrado".

(i) Quando a opção sair for escolhida, o programa deverá finalizar

## Trabalho 2

Crie a estrutura para representar os produtos de um supermercado com o código do produto, nome, preço do fornecedor, preço de venda, quantidade. Gere outra estrutura para representar um carrinho de compras com o código do carrinho, vetor dos produtos comprados composto pelo código do produto, quantidade e valor individual de venda do produto.

(a) Preencha os dados de 5 produtos automaticamente via comandos, para que seja possível testar todas as funções mesmo sem cadastrar manualmente alguns produtos.

(b) Faça com que o programa apresente um menu através de uma função onde o usuário possa escolher entre as funções de cadastrar produto, atualizar produto, comprar produto, buscar produto, exibir faturamento do dia, exibir uma compra específica e sair. Cada funcionalidade deve ser feita como uma função em separado (o usuário deverá poder escolher qualquer das opções qualquer quantidade de vezes. Somente ao escolher a opção sair o programa deve ser finalizado).

(c) Para cadastrar o produto o usuário deverá fornecer todos os dados necessários, sendo eles: código, nome, preço do fornecedor, preço para venda, quantidade no estoque.

(d) Para atualizar um produto, deverá fornecer o código do produto a atualizado e fornecer todos os dados novamente. Caso não haja o código cadastrado deverá exibir: "Código do produto inválido".

(e) Quando a opção comprar produto for escolhida, leia o código de cada produto e a quantidade. Salve os dados na estrutura do carrinho e altere a quantidade do produto no estoque.

(f) Quando a opção buscar produto for escolhida, o programa deverá solicitar ao usuário o nome a procurar. Após isso, o programa irá buscar, entre os produtos cadastrados, um com nome procurado (ou parte dele). Caso encontre, deverá mostrar todos os dados desse produto. Caso contrário, deverá exibir a mensagem "Produto não encontrado".

(g) Quando a opção exibir faturamento do dia for selecionado, deverá exibir todos os códigos das compras junto ao valor seu valor total. No final deverá exibir a soma de todas as compras e o lucro (valor da venda do produto menos o valor do fornecedor) diário.

(h) Quando selecionar a opção de exibir uma compra específica, deverá exibir todos os dados da compra (produtos, quantidade, valor unitário, valor final da compra).

(i) Quando a opção sair for escolhida, o programa deverá finalizar

### Trabalho 3

Crie a estrutura para representar um calendário com os compromissos do usuário com a data e hora, local e descrição do compromisso. A data e a hora devem ser representados com o tipo `date` do python.

(a) Preencha os dados de 5 compromissos automaticamente via comandos, para que seja possível testar todas as funções.

(b) Faça com que o programa apresente um menu através de uma função onde o usuário possa escolher entre as funções de cadastrar compromisso, buscar compromisso por data, alterar compromisso, exibir todos os compromissos, excluir compromisso por posição, excluir compromissos por dia. Cada funcionalidade deve ser feita como uma função em separado (o usuário deverá poder escolher qualquer das opções qualquer quantidade de vezes. Somente ao escolher a opção sair o programa deve ser finalizado).

(c) Para cadastrar um novo compromisso, deverá ser solicitado ao usuário a data, hora, local e descrição do compromisso. Ao receber os dados sobre a data e hora deverá ser convertido no tipo `date` para armazenamento do mesmo.

(d) Dado uma data o mesmo poderá exibir todos os compromissos dessa data mostrando todos os dados cadastrados.

(e) Dada a posição de um compromisso, o mesmo poderá alterar todos os dados relacionados ao compromisso.

(f) Todos os compromissos deverão ser exibidos de forma organizada na tela mostrando a sua posição na lista e o tempo que falta em horas comparando a data atual.

(g) Para excluir um compromisso, o mesmo deverá fornecer a posição na lista de compromissos.

(h) Para excluir um compromisso pelo dia, o usuário deverá fornecer a data e todos os compromissos desta data deverão ser excluídos.

(i) Quando a opção sair for escolhida, o programa deverá finalizar