
	<b>SERVIÇO PÚBLICO FEDERAL</b> <b>MINISTÉRIO DA EDUCAÇÃO</b> <b>CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS</b> <b>UNIDADE DE ENSINO DESCENTRALIZADA DE DIVINÓPOLIS</b>				
	Curso:	<i>Engenharia Mecatrônica</i>	Disciplina:		<i>Programação de Computadores I</i>
Professor(a):				Valor:	
Aluno(a):				Nº:	
Data de Entrega:		Lista:		Nota:	

Lista de exercicios

1. Faça um programa que receba o nome de oito clientes e armazene-os em um vetor. Em um segundo vetor armazene a quantidade de DVDs locados em 2010 por cada um dos oito clientes. Sabe-se que para cada dez locações o cliente tem direito a uma locação grátis. O programa deve mostrar o nome de todos os clientes e a quantidade de locações grátis que ele tem direito.

*Dica 1:* O vetor para armazenar os nomes (strings) pode ser declarado da seguinte forma: `char nomes[TAM_VETOR][TAM_NOME];`

sendo `nomes` o nome da variável vetor, `TAM_VETOR` o tamanho do vetor (quantidade de clientes), e `TAM_NOME` o tamanho máximo da string (contando o caractere nulo) que representa o nome.

2. Faça um programa para corrigir provas de múltipla escolha que foram aplicadas em uma turma de doze alunos. Cada prova tem oito questões e cada questão vale um ponto. O programa deve primeiramente ler o gabarito da prova e em seguida os números de matrícula dos alunos e as respostas que deram às questões. Calcule e mostre:

a) o número e a nota de cada aluno

b) a taxa (em porcentagem) de aprovação, sabendo-se que a nota mínima para aprovação é cinco.

*Dica 1:* Use um vetor de tamanho oito para armazenar o gabarito da prova.

*Dica 2:* Use outro vetor de tamanho oito para armazenar as respostas dos alunos.

*Dica 3:* O tipo utilizado para uma resposta deve ser `char`.

*Dica 4:* Use uma variável simples para calcular a pontuação dos alunos.

*Dica 5:* Use uma variável simples para contar os alunos que foram aprovados.

*Dica 6:* Para corrigir uma prova, compare a resposta dada pelo aluno com a resposta que se encontra no gabarito, para cada uma das oito questões. Se forem iguais some um ponto à pontuação do aluno.

*Dica 7:* Sempre que a pontuação final de um aluno for maior ou igual à pontuação mínima para aprovação (cinco), incremente o contador de alunos aprovados.

3. Desenvolva uma aplicação que simule um caixa automático. Para tanto devem ser lidos os códigos de dez contas e seus respectivos saldos. Não pode haver mais de uma conta com o mesmo código. O programa deverá repetidamente exibir um menu de opções, deixar o usuário escolher uma opção, e executar a operação correspondente, até que ele escolha terminar. As opções do menu são:

Consultar o saldo de uma conta . O usuário deve informar o código da conta. Se o código não estiver cadastrado o usuário deverá ser informado. Se a conta existir, o seu saldo deverá ser exibido.

Efetuar depósito em uma conta . Para efetuar depósito, deve-se solicitar o código da conta e o valor a ser depositado. Se a conta não estiver cadastrada, o usuário deverá ser informado. Se a conta existir, deverá atualizar o seu saldo com o valor do depósito.

Efetuar saque em uma conta . Para efetuar saque, deve-se solicitar o código da conta e o valor a ser sacado. Se conta não estiver cadastrada o usuário deverá ser informado. Se a conta existir e houver saldo suficiente, o saldo deverá ser atualizado com o valor do saque.

Consultar o ativo bancário , ou seja, a soma dos saldos de todas as contas  
Terminar a aplicação .

*Dica 1:* Os códigos das contas devem ser armazenados em um vetor de números inteiros, e o saldo em um vetor de números reais, na mesma posição (índice) do código. Por exemplo, se a conta 504 foi armazenada na quinta posição do vetor de códigos, seu saldo deverá ficar na quinta posição do vetor de saldos.

*Dica 1:* Os códigos das contas devem ser armazenados em um vetor de números inteiros, e o saldo em um vetor de números reais, na mesma posição (índice) do código. Por exemplo, se a conta 504 foi armazenada na quinta posição do vetor de códigos, seu saldo deverá ficar na quinta posição do vetor de saldos.

4. Uma empresa de transporte de passageiros possui ônibus com 48 lugares (24 nas janelas e 24 no corredor). Faça um programa para controlar a ocupação dos lugares enquanto as passagens são vendidas. Inicialmente todas as poltronas estão livres. O programa deverá repetidamente exibir um menu de opções, deixar o usuário escolher uma opção, e executar a operação correspondente, até que ele escolha terminar. As opções do menu são:
- vender passagem
  - mostrar quantidade de lugares livres
  - mostrar mapa de ocupação do ônibus
  - encerrar

Quando a opção escolhida for vender passagem, o usuário deverá informar o número da poltrona. O programa então deverá dar uma das seguintes informações:

- venda efetivada – se a poltrona solicitada estiver livre, marcando-a como ocupada
- poltrona ocupada – se a poltrona não estiver disponível para venda
- ônibus lotado – quando todas as poltronas já estiverem ocupadas.

O mapa de ocupação do ônibus deverá ser formatado como mostro o exemplo:

LADO ESQUERDO

JANELA          CORREDOR

1.ocupado	2.ocupado
5.ocupado	6.ocupado
9.livre	10.ocupado
13.ocupado	14.livre
17.ocupado	18.livre
21.ocupado	22.ocupado
25.ocupado	26.livre
29.ocupado	30.livre
33.livre	34.ocupado
37.livre	38.livre
41.livre	42.livre
45.livre	46.livre

LADO DIREITO

CORREDOR JANELA

3.livre	4.ocupado
7.ocupado	8.livre
11.ocupado	12.ocupado
15.livre	16.livre
19.livre	20.livre
23.ocupado	24.ocupado
27.livre	28.ocupado
31.livre	32.livre
35.livre	36.ocupado
39.ocupado	40.livre
43.livre	44.livre
47.livre	48.livre

*Dica 1:* Utilize um vetor de inteiros para indicar se uma poltrona está ocupada ou não. O valor 1 (verdadeiro) deve ser usado para indicar que a poltrona está ocupada, e o valor 0 (falso) para indicar que a poltrona não está ocupada.

*Dica 2:* O corpo do comando de repetição que irá exibir a tabela de ocupação dos lugares vai exibir uma linha completa (isto é, quatro lugares).