

Trabalhos Práticos

Programação II

Curso: Eng^o Electrotécnica - Electrónica e Computadores

1. Objectivos
 2. Calendarização
 3. Normas
 - 3.1 Relatório de Progresso
 - 3.2 Relatório Final
 - 3.3 Avaliação
 4. Propostas
-

1. Objectivos

Análise, projecto e desenvolvimento de uma pequena aplicação que envolva alguns dos conceitos abordados na disciplina, nomeadamente, estruturas e vectores.

O trabalho prático deve ser realizado em grupo (máximo 2 elementos), extra aulas. No final o trabalho será acompanhado de um relatório e será apresentado e demonstrado individualmente. Este trabalho tem um peso de 25% na Nota Final e de 50% na Nota de Frequência (NFREQ).

Notas:

- Para os alunos com dispensa de avaliação contínua é obrigatória a realização do trabalho prático.
- A Nota de Frequência só é válida no semestre em que foi obtida.

2. Calendarização

Lançamento das propostas de trabalhos: até 23 de Março de 2005

Constituição dos grupos e temas escolhidos: até 8 de Abril de 2005

Entrega do relatório de progresso : até 22 de Abril de 2005

Entrega do trabalho: até 27 de Maio de 2005

Apresentação e discussão: de 30 de Maio a 3 de Junho de 2005

A identificação dos grupos e temas escolhidos, assim como o relatório de progresso e o trabalho deve ser entregue ao professor das aulas práticas.

3. Normas

O trabalho pode ser realizado individualmente ou em grupos de dois alunos, no entanto, a apresentação e discussão será sempre realizada individualmente.

1. A data final de ENTREGA do trabalho é 27 de Maio de 2005. No entanto os grupos terão de cumprir as seguintes fases intermédias:

- Semana de 21/0/05 a 23/034/05: Lançamento dos temas de trabalho.
- Semana de 04/04/05 a 08/04/05: Identificação do grupo e descrição informal (mas escrita) do trabalho.
- Semana de 18/04/05 a 22/04/05: Relatório de Progresso, com uma descrição das estruturas de dados que pretendem utilizar e uma descrição geral da organização do programa.
- Até 27 de Maio de 2005: Entrega do trabalho.

Um grupo que não cumpra um prazo terá uma penalização de 2 valores na nota final do trabalho.

Independentemente destes prazos, os grupos deverão ser capazes de, quando o professor o solicitar, reportar o estado de desenvolvimento do trabalho.

A entrega do trabalho consta de um relatório (ver estrutura no ponto seguinte), código fonte do programa e executável.

2. A apresentação e discussão trabalho decorrerá na semana de 3 a 7 de Janeiro, em dia e hora marcar por cada professor das práticas.

- No dia da apresentação, TODOS os elementos do grupo deverão estar presentes. Os elementos ausentes não terão classificação.

3. Cada grupo é responsável por gerir o seu processo de desenvolvimento. Dificuldades e problemas deverão ser comunicadas atempadamente ao professor das aulas práticas.

3.1 Relatório de Progresso

Dever ser realizado um relatório de progresso, onde deverá constar obrigatoriamente:

- a identificação do trabalho,
- a identificação completa dos alunos que o realizaram,
- a definição das estruturas de dados utilizadas,
- uma explicação da estrutura geral do programa

e qualquer outra informação adicional que o aluno julgue conveniente.

3.2 Relatório Final

Do relatório final deverão constar a identificação do trabalho e a identificação completa dos alunos que o realizaram. Este relatório deverá seguir as seguintes recomendações de estrutura:

1. Introdução – Identificação e descrição do trabalho e objectivos propostos.

2. Estruturas de Dados – definição das estruturas de dados utilizadas no programa.

3. Algoritmos - Apresentar o algoritmo de funcionamento global do programa. Para as funcionalidades fundamentais do programa apresentar o(s) algoritmo(s) envolvidos. Os algoritmos devem ser descritos em Português Estruturado ou Fluxogramas.

4. Manual do Utilizador – Descrição das funcionalidades do programa apresentadas ao utilizador.

5. Conclusões - Apresentação das dificuldades que foram sentidas na execução do trabalho. Limitações do programa desenvolvido e funcionalidades que poderiam/deveriam ser modificadas/melhoradas.

6. Anexos

3.3 Avaliação

Na avaliação do trabalho serão considerados

- a concretização dos objectivos propostos,
- e definição das estruturas de dados,
- a estruturação do programa,
- a robustez,
- o interface.

4. Propostas

Tema 1 – Gestão Pacientes de um Dentista

Tema 2 – Gestão de um Gabinete de Podologia

Tema 3 – Gestão Comercial de uma Loja

Tema 4 – Gestão de um Stand de Automóveis

Tema 5 – Gestão de Apostas de Futebol

Tema 6 – Gestão de um Sistema de Reservatórios de Água

Tema 7 – Informatização da reparadora “O Calhambeque”

Tema 1 – Gestão Pacientes de um Dentista

Pretende-se um programa que permita registar a informação sobre os tratamentos efectuados por um Dentista aos dentes dos seus pacientes.

Assim, para cada paciente deverá ser armazenada a seguinte informação: Nome, Morada, Telefone, Telemóvel, Idade, Dentes (sendo o total 32...). Para cada Dente a designação (por ex: incisivo central superior esquerdo...), se existe ou não (o dente pode já ter sido retirado...), os tratamentos a que foi submetido, e para cada tratamento a Data do Tratamento, o Tipo de Tratamento e o seu Custo

O programa deve permitir ao Dentista efectuar as seguintes operações:

- Inserir, mostrar, alterar e eliminar informação de um paciente
- Inserir tratamento
- Indicar quais os dentes que o paciente já não possui
- Mostrar todos os tratamentos efectuados a um determinado dente de um paciente
- Mostrar todos os clientes com idades compreendidas entre os limites especificados pelo utilizador
- Mostrar o custo total dos tratamentos efectuados a um determinado dente de um paciente

Trabalho

Desenvolvimento das funções de criação e manipulação de arrays de estruturas para armazenar a informação necessária. Deverão ser utilizadas funções de escrita (inserção e alteração e eliminação), leitura (consultas), pesquisa de dados no array e ordenação se assim entender necessário.

Utilização de ficheiros para guardar os dados. (Deverá gravar/ler um registo de cada vez). Permitir ordenação dos dados por nome de paciente.

Tema 2 – Gestão de um Gabinete de Podologia

Um gabinete de Podologia pretende uma aplicação para a gestão dos tratamentos aí efectuados. Para tal, devem ser criadas as estruturas necessárias para guardar as informações sobre:

- Pacientes (Código do paciente, nome, morada, telefone,...,sistema de saúde);
- Tratamentos(Código do tratamento, tipo de tratamento, data de início e data de fim);

Os tratamentos podem ser diversos e os preços variam consoante o Sistema de Saúde de cada paciente, bem como alguns clientes têm um preço especial devido ao facto de serem pacientes permanentes.

O programa deve permitir ao utilizador:

- A gestão de todas as tabelas utilizadas (Inserção, remoção ou actualização).
- Efectuar a Listagem discriminada dos valores a pagar mensalmente por paciente ou por sistema de saúde.
- Efectuar a listagem discriminada do valor a pagar por um paciente no final do tratamento (por tipo de tratamento e respectivo total) .

Para a marcação dos pacientes deve ser criada uma lista ordenada por código ou por nome do paciente, com o intuito de facilitar a sua marcação, que deve ser confirmada no dia.

Trabalho

Desenvolvimento das funções de criação e manipulação de arrays de estruturas para armazenar a informação necessária. Deverão ser utilizadas funções de escrita (inserção e alteração e eliminação), leitura (consultas), pesquisa de dados no array e ordenação se assim entender necessário.

Utilização de ficheiros para guardar os dados. (Deverá gravar/ler um registo de cada vez). Permitir ordenação dos dados por nome de paciente.

Tema 3 – Gestão Comercial de uma loja

O proprietário de uma loja pretende ter informatizada informação sobre os produtos que vende habitualmente e respectivos fornecedores. Terá que ser guardada informação sobre o nome do produto, a referência do produto, o nome do fornecedor, preço de compra do produto, preço de venda do produto e stock existente.

Note que cada produto pode ter mais do que um fornecedor (máximo 5), tendo para cada fornecedor um determinado preço de compra e prazo de entrega.

Desenvolva um programa em Linguagem C que implemente uma aplicação para ajudar o proprietário da loja. A aplicação deverá manter num vector os dados dos produtos ordenados alfabeticamente, e conter as seguintes funcionalidades:

- Acrescentar produtos;
- Acrescentar fornecedor a um produto;
- Listar produtos;
- Retirar produtos;
- Alterar preço de compra de um produto a um determinado fornecedor;
- Alterar preço de venda de um produto;
- Pesquisar para um dado produto qual o fornecedor aconselhável;

Trabalho

Desenvolvimento das funções de criação e manipulação de arrays de estruturas para armazenar a informação necessária. Deverão ser utilizadas funções de escrita (inserção e alteração e eliminação), leitura (consultas), pesquisa de dados no array e ordenação se assim entender necessário.

Utilização de ficheiros para guardar os dados. (Deverá gravar/ler um registo de cada vez). Permitir ordenação dos dados por produto.

Tema 4 – Gestão de um Stand de Automóveis

Pretende-se uma aplicação que permita a gestão de um *stand* de automóveis. A aplicação, desenvolvida em C deve permitir a gestão de vendedores, automóveis e das vendas realizadas.

Vendedores	Automóveis	Vendas
Inserir novo vendedor	Inserir automóvel	Inserir venda
Eliminar vendedor	Listar automóvel	Alterar dados da venda
Alterar vendedor	Eliminar automóvel (através de	Consultar venda
Consultar vendedor	venda ou por sinistro)	Anular venda
	Alterar automóvel	

Trabalho

A aplicação deve permitir armazenar e ler os dados dos vendedores, automóveis e vendas em três ficheiros binários. Para além disso deve estar disponível uma opção que crie um ficheiro de texto com as vendas realizadas por um vendedor introduzido pelo utilizador.

Tema 5 – Gestão de Apostas de Futebol

Durante os primeiros meses do corrente ano fizeram-se inquéritos por todo o país a adeptos de futebol, com o objectivo de se conhecer as suas opiniões em relação à equipa que pensam que vai ganhar o campeonato deste ano.

A informação recolhida por entrevistado foi o nome da cidade onde vive, a data em que foi feita a entrevista e o código da equipa que pensa ter mais possibilidades de vir a ser campeã, devendo servir de base para actualizar um outro ficheiro, binário, das apostas obtidas até ao momento.

Trabalho

Construa uma aplicação em C que para além de fazer o processamento das apostas anteriormente descrito, responda também às seguintes funcionalidades:

- Permitir a entrada de apostas *on-line*;
- Manter actualizado um ficheiro com os códigos de clubes e respectivos nomes;
- Eliminar todas as apostas para uma determinada cidade;
- Visualizar todos os inquéritos realizados num determinado dia;
- Listar por ordem alfabética as cidades e por cada uma, o nome do clube e a respectiva quantidade de apostas contabilizadas até ao momento.

Exemplo de uma listagem

CIDADE	CLUBE	APOSTAS
Braga	F.C. Braga	100
	S.L. Benfica	30
	F.C. Porto	20
	Sporting C. P.	25
Coimbra	F.C. Porto	40
	Sporting C. P.	60
	S.L. Benfica	30
	Académica	45
Lisboa

Tema 6 – Gestão de um Sistema de Reservatórios de Água

Pretende-se desenvolver uma aplicação que simule o comportamento de um sistema de reservatórios de água. O sistema a analisar não terá mais do que dez tanques.

Os reservatórios estão ligados em série, isto é, a saída do primeiro reservatório encontra-se ligada à entrada do segundo reservatório, e assim sucessivamente. A ligação entre cada par de reservatórios é realizada por uma canalização com uma válvula unidireccional que permite variar a resistência ao caudal.

Os reservatórios poderão estar cobertos (sem entrada de água da chuva) ou descobertos (com entrada de chuva pelo topo do reservatório). As dimensões dos reservatórios (Largura x Profundidade x Altura), em metros, são totalmente independentes de reservatório para reservatório.

As equações que permitem aproximar o comportamento dinâmico da água entre reservatórios são as seguintes:

1. $\text{Caudal} = (h_1 - h_2) / \text{Resistência da válvula (m}^3/\text{s)}$

O caudal é a quantidade de água que passa pela canalização, do tanque 1 para o tanque 2, por unidade de tempo. h_1 é o nível (altura) da água no reservatório de origem, e h_2 é o nível da água no reservatório de destino.

2. $\text{Volume deslocado} = \text{Caudal} * T \text{ (m}^3\text{)}$

O volume deslocado é a quantidade de água que passa de um tanque para o seguinte, em T segundos. T representa o tempo de um passo (iteração) da simulação.

3. $\text{Variação da altura} = \text{Volume deslocado} / \text{Superfície do tanque (m)}$

A variação da altura do nível da água é o aumento ou diminuição do nível devida ao volume de água que entrou ou saiu do tanque.

Se o tanque não tiver cobertura, é necessário também contar com a entrada de chuva pelo topo, sendo necessária a seguinte equação:

4. $\text{Volume pluviosidade} = \text{Fluxo de chuva} * \text{Superfície do tanque (m}^3\text{)}$

Trabalho

Objectivo 1

A aplicação deverá permitir a definição do número de tanques, as características de cada um deles (dimensões, cobertura, resistência da válvula de saída, ...) bem como os parâmetros da pluviosidade, o tempo que representa cada passo da simulação e ainda o número de iterações da simulação.

Objectivo 2

O progresso da simulação deverá ser guardado num ficheiro, cujo nome é dado pelo utilizador. O estado do sistema após cada iteração deverá ser registado no referido ficheiro. O programa a desenvolver deverá apresentar o progresso individual de um tanque durante uma simulação registada.

Tema 7 – Informatização da reparadora “O Calhambeque”

A empresa de reparação de automóveis “O Calhambeque” pretende informatizar a área de intervenção para com os clientes. O sistema de informação deverá estar organizado de forma a permitir emitir uma factura para os consertos das viaturas dos vários clientes.

Note que cada cliente pode ser proprietário de mais que uma viatura, uma factura refere-se sempre a um conserto de uma única viatura. Nas reparações podem ser contabilizados preços para peças e mão-de-obra.

Desenvolva um programa em Linguagem C que implemente uma aplicação para ajudar o proprietário da reparadora. A aplicação deverá permitir as seguintes funcionalidades:

- Gestão de clientes;
- Gestão de viaturas dos clientes;
- Gestão de peças;
- Gestão de reparações;
- Emissão da factura de uma reparação.

Trabalho

Desenvolvimento das funções de criação e manipulação de arrays de estruturas para armazenar a informação necessária. Deverão ser utilizadas funções de escrita (inserção e alteração e eliminação), leitura (consultas), pesquisa de dados no array e ordenação se assim entender necessário.

Utilização de ficheiros para guardar os dados. (Deverá gravar/ler um registo de cada vez). Permitir ordenação dos dados por produto.