Princípios de Programação Procedimental 2023/24

Licenciatura em Engenharia Informática

Projeto gestão de doentes

Discente: **Número:**

-Calvin Fernando Manhique Comolo **2021243519**

-Cíntia Dalila Luís Cumbane **2020244607**

1 2 9 0

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D COIMBRA

Coimbra, aos 23 de Maio de 2024

MANUAL DO PROGRAMADOR

Estrutura geral do Programa

O programa contém os seguintes ficheiros:

gestao_pacientes.c- ficheiro contendo as funções que permitem as operações básicas das Listas Ligadas:

- cria- Cria a lista
- destroi- Destroi a lista
- vazia- Verifica se a lista está vazia
- insere- Insere um elemento na lista
- retira- Remove um elemento da lista
- pesquisa- Devolve um ponteiro para um elemento (se existir)
- imprime- Imprime a informação da lista

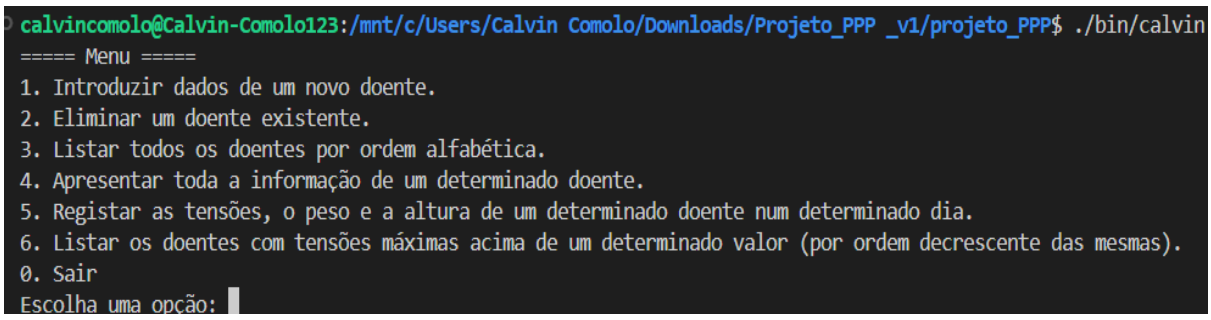
Contem também funções de leitura e escrita nos ficheiros de texto.

funcoes.h- ficheiro contendo o cabeçalho das funções em gestao_pacientes.c

ficheiros.c- ficheiro contendo funções de leitura de inputs de forma protegida.

ficheiros.h- ficheiro contendo o cabeçalho das funções em ficheiros.c.

main.c- ficheiro contendo o main() do programa. Esta função cria um menu que permite a interação do utilizador com o programa.



gestão_pacientes.h- ficheiro com as estruturas de dados e bibliotecas usadas.

O programa contém também dois ficheiros de texto:

dados_doentes.txt- ficheiro de texto com informação dos doentes.

registros_doentes.txt- ficheiro de texto com a informação dos registos.

Principais estruturas de dados usadas

Neste projeto, foi desenvolvida uma aplicação para a gestão de pacientes (doentes), com foco no armazenamento e monitoramento dos dados clínicos, como registos de tensões, peso e altura. Para tal, foram definidas várias estruturas em C que permitem a organização e manipulação dos dados de forma eficiente. Abaixo, são detalhadas as estruturas utilizadas.

Estrutura doente

A estrutura doente armazena informações pessoais e clínicas de um paciente. Ela inclui campos para identificação, nome, data de nascimento, número de cartão de cidadão, telefone, e-mail e um ponteiro para uma lista de registos clínicos (*record_head). Além disso, contém um ponteiro *maior_tensao_max para o registo com a maior tensão máxima.

Estrutura noListaDoente

Esta estrutura é um nó de uma lista ligada que contém pacientes (doente). Cada nó da lista (*noListaDoente) aponta para o próximo nó, permitindo a criação de listas dinâmicas de pacientes.

Estrutura registo

A estrutura registo é usada para armazenar os dados clínicos de um paciente, como tensões, peso e altura. Cada registo também inclui um ponteiro *next_tensao, que aponta para o próximo registro na lista, facilitando a organização dos registos com base na tensão máxima.

Estrutura noListaRegisto

Esta estrutura é um nó de uma lista ligada que contém registos clínicos. Cada nó da lista (noListaRegisto) aponta para o próximo nó, permitindo a criação de listas dinâmicas de registros.

Estrutura data

A estrutura data é utilizada para representar datas, como a data de nascimento do paciente ou a data de um registo clínico. Ela possui três campos: dia, mes e ano.

Funcionalidades

Carregar/guardar dados nos ficheiros

Ao iniciar o programa os dados dos ficheiros são lidos e as informações do doente são colocados na lista de doentes e as informações do registo são colocados na fila de registos do respetivo doente. Função: void lerDadosDoentes(const char *nome_arquivo, dLista lista_pacientes);

Inserir doente

Os doentes são inseridos por ordem de nomes, mantendo a fila sempre ordenada por ordem alfabética. Sempre que um doente é inserido, os dados são acrescentados no ficheiro dados_doentes.txt. Função: void insere(dLista lista, struct doente p1).

Eliminar doente

Ao eliminar um doente da lista liberta-se a memória associada ao nó do doente bem como os nós associados aos registos desse doente. Em seguida a informação na fila de doente é reescrita nos ficheiros. Função: void elimina(dLista lista, int chave).

Listar doentes por ordem alfabética

Para listar os doentes por ordem alfabética basta imprimir da lista (o campo name) uma vez que ela já se encontra ordenada por nomes. Funções: void imprime(dLista lista);

Registar tensões, peso e altura num determinado dia

Os registos dos doentes são inseridos por ordem das datas. Sempre que um registo é inserido, a informação é acrescentada no ficheiro registros_doentes.txt. Função: void insere_registro(dLista lista, int id_doente, struct registo novo_registro);

Apresentar todas informações de um doente.

A procura do doente em questão é feita por ID, em seguida percorre-se a respetiva lista de registos e imprime-se toda informação. Função: dLista pesquisa(dLista lista, int chave);

Listar doentes com tensões máximas acima de um determinado valor (por ordem decrescente das mesmas). – LISTA AXILIAR

Para poder imprimir as tensões por ordem decrescente, a estrutura doente contém um ponteiro extra (*maior_tensao_max) para o registo com a maior tensão máxima e a estrutura registo contem um

ponteiro (*next_tensao) para o próximo registo com tensão máxima menor. Permitindo assim, através desses dois ponteiros, percorrer lista de registos por ordem decrescente das tensões máximas (até encontrar um determinado valor). Função: void listar_doentes_com_tensao_acima(dLista lista, int valor);

Proteções de input

O código está protegido de maneira que que todos inputs sejam lidos como strings. Depois valida-se o input transformando-o para um inteiro se o input tiver apenas dígitos (com exceção do nome, cartão de cidadão e e-mail).

Data: os inputs das datas estão protegidos de maneira que só se possa inserir inteiros válidos para o dia, mês e ano. Quanto às datas de registo, estes não podem ser anteriores à data de nascimento do paciente.

Telefone: os números de telefone estão protegidos de modo que só se possa inserir inteiros entre 910000000 e 939999999.

Tensão: a tensão mínima de um paciente num determinado dia não pode ser maior que a sua tensão máxima.

O nome, número de cartão de cidadão e e-mail são campos sem proteção, podendo misturar dígitos, letras e carateres especiais, mas não podem estar vazios.