#include <stdio.h>

#include <stdlib.h>

#include <string.h>

// Definição da estrutura do nó da lista

typedef struct Node {

char strValue[100];

float floatValue;

struct Node \*prev;

struct Node \*next;

} Node;

// Função para criar um novo nó

Node \*createNode(char \*strValue, float floatValue) {

Node \*newNode = (Node \*)malloc(sizeof(Node));

if (newNode == NULL) {

printf("Erro: Falha ao alocar memória para o novo nó.\n");

exit(1);

}

strcpy(newNode->strValue, strValue);

newNode->floatValue = floatValue;

newNode->prev = NULL;

newNode->next = NULL;

return newNode;

}

// Função para inserir um novo nó no final da lista

void insertEnd(Node \*\*head, char \*strValue, float floatValue) {

Node \*newNode = createNode(strValue, floatValue);

if (\*head == NULL) {

\*head = newNode;

} else {

Node \*temp = \*head;

while (temp->next != NULL) {

temp = temp->next;

}

temp->next = newNode;

newNode->prev = temp;

}

}

// Função para remover um nó da lista

void removeNode(Node \*\*head, char \*strValue, float floatValue) {

if (\*head == NULL) {

printf("A lista está vazia.\n");

return;

}

Node \*temp = \*head;

while (temp != NULL) {

if (strcmp(temp->strValue, strValue) == 0 &&

temp->floatValue == floatValue) {

if (temp->prev != NULL) {

temp->prev->next = temp->next;

} else {

\*head = temp->next;

}

if (temp->next != NULL) {

temp->next->prev = temp->prev;

}

free(temp);

printf("Nó com valor \"%s\" e %.2f removido com sucesso.\n", strValue,

floatValue);

return;

}

temp = temp->next;

}

printf("Nó com valor \"%s\" e %.2f não encontrado na lista.\n", strValue,

floatValue);

}

// Função para reinicializar a lista

void clearList(Node \*\*head) {

Node \*current = \*head;

Node \*next;

while (current != NULL) {

next = current->next;

free(current);

current = next;

}

\*head = NULL;

printf("Lista reinicializada com sucesso.\n");

}

// Função para imprimir a lista

void printList(Node \*head) {

printf("Lista:\n");

while (head != NULL) {

printf("Produto: %s, Quantidade(Unidade/Kg): %.2f\n", head->strValue,

head->floatValue);

head = head->next;

}

printf("\n");

}

int main() {

Node \*head = NULL;

int choice;

char strValue[50];

char floatValue[50];

do {

printf("\n--- Menu ---\n");

printf("1. Inserir\n");

printf("2. Remover\n");

printf("3. Reinicializar lista\n");

printf("4. Imprimir lista\n");

printf("0. Sair\n");

printf("Escolha: ");

fgets(strValue, sizeof(strValue), stdin);

choice = atoi(strValue);

switch (choice) {

case 1:

printf("Digite o nome do produto: ");

fgets(strValue, sizeof(strValue), stdin);

printf("Digite a quantidade(unid ou Kg) do produto: ");

fgets(floatValue, sizeof(floatValue), stdin);

insertEnd(&head, strValue, atof(floatValue));

break;

case 2:

printf("Digite o nome do produto a ser removido: ");

fgets(strValue, sizeof(strValue), stdin);

printf("Digite a quantidade(ou Kg) do produto que está sendo removido: ");

fgets(floatValue, sizeof(floatValue), stdin);

removeNode(&head, strValue, atof(floatValue));

break;

case 3:

clearList(&head);

break;

case 4:

printList(head);

break;

case 0:

clearList(&head);

printf("Encerrando o programa.\n");

break;

default:

printf("Escolha inválida. Tente novamente.\n");

}

} while (choice != 0);

return 0;

}