#include <stdio.h>

#include <stdlib.h>

typedef struct nodo {

char data;

struct nodo \*link;

} \*ptr;

typedef struct fila {

ptr inicio;

ptr fim;

} Fila;

// Cria a fila (inicializa os ponteiros)

void create(Fila \*f) {

f->inicio = NULL;

f->fim = NULL;

}

// Verifica se a fila está vazia

int isEmpty(Fila \*f) {

return f->inicio == NULL;

}

// Adiciona um elemento no final da fila

void store(Fila \*f, char elem) {

ptr aux = malloc(sizeof(struct nodo));

aux->data = elem;

aux->link = NULL;

if (isEmpty(f)) {

f->inicio = aux;

f->fim = aux;

} else {

f->fim->link = aux;

f->fim = aux;

}

}

// Remove e retorna o elemento do início da fila

char retrieve(Fila \*f) {

if (isEmpty(f)) {

printf("Fila vazia!\n");

return '\0';

}

ptr auxptr = f->inicio;

char aux = f->inicio->data;

f->inicio = auxptr->link;

if (f->inicio == NULL) {

f->fim = NULL;

}

free(auxptr);

return aux;

}

// Libera a memória da fila

void destroy(Fila \*f) {

while (!isEmpty(f)) {

retrieve(f);

}

}

// Exibe os elementos da fila

void exibe(Fila f) {

ptr temp = f.inicio;

printf("Fila: [ ");

while (temp != NULL) {

printf("%c ", temp->data);

temp = temp->link;

}

printf("]\n");

}

int main(void) {

Fila f;

create(&f); // Cria a fila

store(&f, 'A'); // Adiciona 'A'

exibe(f);

store(&f, 'B'); // Adiciona 'B'

exibe(f);

store(&f, 'C'); // Adiciona 'C'

exibe(f);

printf("\nRetira o primeiro da fila: %c\n", retrieve(&f)); // Remove 'A'

exibe(f);

printf("Retira o próximo da fila: %c\n", retrieve(&f)); // Remove 'B'

exibe(f);

printf("Retira o próximo da fila: %c\n", retrieve(&f)); // Remove 'C'

exibe(f);

printf("Fila vazia? %d\n", isEmpty(&f)); // Verifica se a fila está vazia

destroy(&f); // Libera a fila

printf("Fila destruída.\n");

return 0;

}