Parte A

O exemplo escolhido para a criação de fila foi a utilização de frutas como elementos desta lista. Para a completa exibição das 15 operações solicitadas, a resposta excederá o número de linhas sugeridas.

insert(“Maçã”);

Fila == Inicio: Maçã :Fim

insert(“Laranja”);

Fila == Inicio: Maçã, Laranja :Fim

remove();

Fila == Inicio: Laranja :Fim

remove();

Fila == Inicio: :Fim

insert(“Maçã”);

Fila == Inicio: Maçã :Fim

insert(“Laranja”);

Fila == Inicio: Maçã, Laranja :Fim

insert(“Morango”);

Fila == Inicio: Maçã, Laranja, Morango :Fim

insert(“Pêssego”);

Fila == Inicio: Maçã, Laranja, Morango, Pêssego :Fim

insert(“Limão”);

Fila == Inicio: Maçã, Laranja, Morango, Pêssego, Limão :Fim

insert(“Banana”);

Fila == Inicio: Maçã, Laranja, Morango, Pêssego, Limão, Banana :Fim

remove(2);

Fila == Inicio: Maçã, Laranja, Pêssego, Limão, Banana :Fim

Removed(4);

Fila == Inicio: Maçã, Laranja, Pêssego, Limão :Fim

Removed(1);

Fila == Inicio: Maçã, Pêssego, Limão :Fim

Insert(“Melão”);

Fila == Inicio: Maçã, Pêssego, Limão, Melão :Fim

remove();

Fila == Inicio: Pêssego, Limão, Melão :Fim

Parte B

Para atender a demanda da questão, será necessário exceder o números de linhas sugerido, visto que o código completo contempla mais linhas do que o sugerido pelo enunciado. A struc foi nomeada como “exerciciofael”.

#include <stdio.h>

#include <stdlib.h>

**struct** Exerciciofael{

**int** num;

**struct** Exerciciofael \*prox;

};

**typedef** **struct** Exerciciofael exerciciofael;

**int** tam;

**void** inicia(exerciciofael \*LISTA);

**int** menu(**void**);

**void** opcao(exerciciofael \*LISTA, **int** op);

exerciciofael \*criaNo();

**void** insereFim(exerciciofael \*LISTA);

**void** insereInicio(exerciciofael \*LISTA);

**void** exibe(exerciciofael \*LISTA);

**void** libera(exerciciofael \*LISTA);

**void** insere (exerciciofael \*LISTA);

exerciciofael \*retiraInicio(exerciciofael \*LISTA);

exerciciofael \*retiraFim(exerciciofael \*LISTA);

exerciciofael \*retira(exerciciofael \*LISTA);

**int** main(**void**)

{

exerciciofael \*LISTA = (exerciciofael \*) malloc(**sizeof**(exerciciofael));

**if**(!LISTA){

printf("Sem memoria disponivel!\n");

exit(1);

}**else**{

inicia(LISTA);

**int** opt;

**do**{

opt=menu();

opcao(LISTA,opt);

}**while**(opt);

free(LISTA);

**return** 0;

}

}

**void** inicia(exerciciofael \*LISTA)

{

LISTA->prox = NULL;

tam=0;

}

**int** menu(**void**)

{

**int** opt;

printf("Escolha a opcao\n");

printf("0. Sair\n");

printf("1. Zerar lista\n");

printf("2. Exibir lista\n");

printf("3. Adicionar exerciciofael no inicio\n");

printf("4. Adicionar exerciciofael no final\n");

printf("5. Escolher onde inserir\n");

printf("6. Retirar do inicio\n");

printf("7. Retirar do fim\n");

printf("8. Escolher de onde tirar\n");

printf("Opcao: "); scanf("%d", &opt);

**return** opt;

}

**void** opcao(exerciciofael \*LISTA, **int** op)

{

exerciciofael \*tmp;

**switch**(op){

**case** 0:

libera(LISTA);

**break**;

**case** 1:

libera(LISTA);

inicia(LISTA);

**break**;

**case** 2:

exibe(LISTA);

**break**;

**case** 3:

insereInicio(LISTA);

**break**;

**case** 4:

insereFim(LISTA);

**break**;

**case** 5:

insere(LISTA);

**break**;

**case** 6:

tmp= retiraInicio(LISTA);

printf("Retirado: %3d\n\n", tmp->num);

**break**;

**case** 7:

tmp= retiraFim(LISTA);

printf("Retirado: %3d\n\n", tmp->num);

**break**;

**case** 8:

tmp= retira(LISTA);

printf("Retirado: %3d\n\n", tmp->num);

**break**;

**default**:

printf("Comando invalido\n\n");

}

}

**int** vazia(exerciciofael \*LISTA)

{

**if**(LISTA->prox == NULL)

**return** 1;

**else**

**return** 0;

}

exerciciofael \*aloca()

{

exerciciofael \*novo=(exerciciofael \*) malloc(**sizeof**(exerciciofael));

**if**(!novo){

printf("Sem memoria disponivel!\n");

exit(1);

}**else**{

printf("Novo elemento: "); scanf("%d", &novo->num);

**return** novo;

}

}

**void** insereFim(exerciciofael \*LISTA)

{

exerciciofael \*novo=aloca();

novo->prox = NULL;

**if**(vazia(LISTA))

LISTA->prox=novo;

**else**{

exerciciofael \*tmp = LISTA->prox;

**while**(tmp->prox != NULL)

tmp = tmp->prox;

tmp->prox = novo;

}

tam++;

}

**void** insereInicio(exerciciofael \*LISTA)

{

exerciciofael \*novo=aloca();

exerciciofael \*oldHead = LISTA->prox;

LISTA->prox = novo;

novo->prox = oldHead;

tam++;

}

**void** exibe(exerciciofael \*LISTA)

{

system("clear");

**if**(vazia(LISTA)){

printf("Lista vazia!\n\n");

**return** ;

}

exerciciofael \*tmp;

tmp = LISTA->prox;

printf("Lista:");

**while**( tmp != NULL){

printf("%5d", tmp->num);

tmp = tmp->prox;

}

printf("\n ");

**int** count;

**for**(count=0 ; count < tam ; count++)

printf(" ^ ");

printf("\nOrdem:");

**for**(count=0 ; count < tam ; count++)

printf("%5d", count+1);

printf("\n\n");

}

**void** libera(exerciciofael \*LISTA)

{

**if**(!vazia(LISTA)){

exerciciofael \*proxExerciciofael,

\*atual;

atual = LISTA->prox;

**while**(atual != NULL){

proxExerciciofael = atual->prox;

free(atual);

atual = proxExerciciofael;

}

}

}

**void** insere(exerciciofael \*LISTA)

{

**int** pos,

count;

printf("Em que posicao, [de 1 ate %d] voce deseja inserir: ", tam);

scanf("%d", &pos);

**if**(pos>0 && pos <= tam){

**if**(pos==1)

insereInicio(LISTA);

**else**{

exerciciofael \*atual = LISTA->prox,

\*anterior=LISTA;

exerciciofael \*novo=aloca();

**for**(count=1 ; count < pos ; count++){

anterior=atual;

atual=atual->prox;

}

anterior->prox=novo;

novo->prox = atual;

tam++;

}

}**else**

printf("Elemento invalido\n\n");

}

exerciciofael \*retiraInicio(exerciciofael \*LISTA)

{

**if**(LISTA->prox == NULL){

printf("Lista ja esta vazia\n");

**return** NULL;

}**else**{

exerciciofael \*tmp = LISTA->prox;

LISTA->prox = tmp->prox;

tam--;

**return** tmp;

}

}

exerciciofael \*retiraFim(exerciciofael \*LISTA)

{

**if**(LISTA->prox == NULL){

printf("Lista ja vazia\n\n");

**return** NULL;

}**else**{

exerciciofael \*ultimo = LISTA->prox,

\*penultimo = LISTA;

**while**(ultimo->prox != NULL){

penultimo = ultimo;

ultimo = ultimo->prox;

}

penultimo->prox = NULL;

tam--;

**return** ultimo;

}

}

exerciciofael \*retira(exerciciofael \*LISTA)

{

**int** opt,

count;

printf("Que posicao, [de 1 ate %d] voce deseja retirar: ", tam);

scanf("%d", &opt);

**if**(opt>0 && opt <= tam){

**if**(opt==1)

**return** retiraInicio(LISTA);

**else**{

exerciciofael \*atual = LISTA->prox,

\*anterior=LISTA;

**for**(count=1 ; count < opt ; count++){

anterior=atual;

atual=atual->prox;

}

anterior->prox=atual->prox;

tam--;

**return** atual;

}

}**else**{

printf("Elemento invalido\n\n");

**return** NULL;

}

}