1°)

typedef struct NO{

float p;

int qtdEstoque;

char nome[20], desc[100];

struct NO \* prox;

}TNo;

void aumento(TNo \* L, float porc, char nome[]){

int cont = 0;

if(L == NULL){

return;

}else{

while(L != NULL){

if(strcmp(L->nome, nome) == 0){

L->preco += L->preco \* (porc/100);

cont++;

}

}

}

printf("Foram alterados %d produtos", cont);

}

2°)

typedef struct NO(){

int info;

struct NO \* prox, ant;

}TNo;

void buscar\_remover(TNo \*\* L, int num){

TNo \* aux, aux2;

int cont = 0;

if(\*L == NULL){

printf("\nLista vazia");

return;

}else{

if((\*L)->info == num){

aux = \*L;

L->prox->ant = NULL;

\*L = (\*L)->prox;

free(aux);

cont++;

}else{

aux = L;

aux2 = aux->prox;

while(aux2 != NULL){

if(aux2->info > num){

return;

}else if(aux2->info == num){

aux->prox = aux2->prox;

aux2->prox->ant = aux;

free(aux2);

}else{

aux = aux2;

aux2 = aux2->prox;

}

}

}

}

}

3°)

void copiar(TNo \* L, TNo \*\* LCopia){

TNo \* ax1 = \*LCopia, novo, ax2 = L, ax3;

while(ax2 != NULL){

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

novo->info = ax2->info;

if(ax1 == NULL{

ax1 = novo;

novo->ant = NULL;

novo->prox=NULL;

}else{

ax3 = ax1;

while(ax3->prox != NULL)

ax3 = ax3->prox;

ax3->prox = novo;

novo->ant = ax3;

novo->prox = NULL;

}

}

}

4°)

typedef struct NO{

int info;

struct NO \*ant, \*prox;

}TNo;

void inserir(TNo \*\* L, int n){

TNo \* novo, aux, aux2;

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

if(\*L == NULL){

novo->prox = NULL;

novo->ant = NULL;

\*L = novo;

}else if((\*L)->info >= n){

novo->ant = NULL;

novo->prox = L;

L->ant == novo;

\*L = novo;

}else{

aux = L;

aux2 = aux->prox;

while(aux2 != NULL){

if(aux2->info >= n){

novo->prox = aux2;

novo->ant = aux;

aux2->ant = novo;

aux->prox = novo;

aux2->ant = novo;

}else{

aux = aux2;

aux2 = aux2->prox;

}

}

aux->prox = novo;

novo->ant = aux;

novo->prox = NULL;

}

}

5°)

typedef struct NO{

float info;

struct NO \* prox;

}TNo;

void remover(TNo \*\* L, int n){

Tnum \*aux, \*aux2;

if(\*L == NULL){

printf("Lista Vazia!");

return;

}else{

if((\*L)-> info == num){

aux = L;

\*L = \*L->prox;

free(aux);

}else{

aux = \*L;

aux2 = aux->prox;

while(aux2 != NULL){

if(n > aux2->info)

return;

else if(aux2->num == n){

aux->prox = aux2->prox;

free(aux2);

}else{

aux = aux2;

aux2 = aux2->prox;

}

}

}

}

}

6°)

void inverter(TNo \*\* L){

TNo \* ax1, ax2, ax3, ax4, ax5, ax6;

ax1 = (\*L);

ax2 = ax1;

while(ax2->prox != NULL)

ax2 = ax2->prox;

ax4 = ax1->prox;

ax5 = ax2->ant;

ax2->prox = ax4;

ax2->ant = NULL;

ax1->prox= NULL;

while(ax1->ant != ax2){

ax3 = ax1->ant;

ax4 = ax1->prox;

ax5 = ax2->ant;

ax6 = ax2->prox;

ax3->prox = ax1;

ax4->ant = ax1;

ax5->prox = ax2;

ax6->ant = ax2;

ax1 = ax4;

ax2 = ax5;

}

}