Ex.1) Usando o programa da semana passada, mostre os valores dos nós acima da média da lista ligada.

#include <stdio.h>

#include <stdlib.h>

struct tipoNo{

int valor;

struct tipoNo \*prox;

};

void mostrarLista (struct tipoNo \*parm);

int somaLista(struct tipoNo \*parm);

int contaLista(struct tipoNo \*parm);

void mostrarAcima(struct tipoNo \*parm, float media);

int main(void){

struct tipoNo \*head = NULL, \*no;

int qtd = 1, pos;

float soma, media;

printf("Digite 5 valores\n");

for(pos = 1; pos < 6; pos++){

printf("Digite o valor: ");

scanf("%d",&qtd);

if(!(no = malloc(sizeof(struct tipoNo)))){

printf("Faltou memoria!\n");

exit(1);

}

no -> prox = head;

head = no;

no -> valor = qtd;

}

printf("\n Fim da criacao da lista\n");

printf("\n Mostra a lista:\n");

no = head;

mostrarLista(no);

printf("Destruindo a lista:\n");

no = head;

while(no != NULL){

head = head -> prox;

free(no);

no = head;

}

soma = somaLista(head);

qtd = contaLista(head);

media = (float) soma / qtd;

mostrarAcima(head , media);

return 0;

}

int somaLista(struct tipoNo \*parm){

int soma = 0;

struct tipoNo \*aux;

aux = parm;

while(aux != NULL){

soma = soma + aux -> valor;

aux = aux -> prox;

}

return soma;

}

int contaLista(struct tipoNo \*parm){

int qtd = 0;

struct tipoNo \*aux;

aux = parm;

while (aux != NULL){

qtd = qtd + 1;

aux = aux -> prox;

}

return qtd;

}

void mostrarAcima(struct tipoNo \*p, float m){

struct tipoNo \*aux;

aux = p;

while(p!=NULL){

if(aux -> valor > m){

printf("Valor = %d\n", aux -> valor);

}

aux = aux -> prox;

}

}

void mostrarLista(struct tipoNo \*parm){

while(parm != NULL){

printf("\n Valor = %d\n", parm -> valor);

parm = parm -> prox;

}

}

Uma lista ligada ou lista encadeada também é chamada de LISTA SIMPLESMENTE LIGADA porque possui apenas o apontador para o próximo elemento. Em algumas situações é necessário acrescentar outro apontador para o elemento anterior.

Exemplo: struct tipoDupla{

int valor;

struct tipoDupla \*prox, \*ant; //proximo e anterior

};

Neste caso, temos uma lista duplamente ligada

Quando o último elemento aponta para o primeiro, nós chamamos de:

Lista circular

As estruturas de dados costuma oferecer métodos de acesso para:

inserir valor ou acrescentar nó;

remover valor ou retirar nó;

consultar valor;

ordenar valor;

Ex.2) Criar função para informar se a lista ligada contém um valor digitado pelo usuário na função main().

#include <stdio.h>

#include <stdlib.h>

struct tipoNo{

int valor;

struct tipoNo \*prox;

};

void mostrarLista (struct tipoNo \*parm);

int somaLista(struct tipoNo \*parm);

int contaLista(struct tipoNo \*parm);

void mostrarAcima(struct tipoNo \*parm, float media);

int contemLista(struct tipoNo \*parm);

int main(void){

struct tipoNo \*head = NULL, \*no;

int qtd = 1, pos;

float soma, media;

printf("Digite 5 valores\n");

for(pos = 1; pos < 6; pos++){

printf("Digite o valor: ");

scanf("%d",&qtd);

if(!(no = malloc(sizeof(struct tipoNo)))){

printf("Faltou memoria!\n");

exit(1);

}

no -> prox = head;

head = no;

no -> valor = qtd;

}

printf("\n Fim da criacao da lista\n");

printf("\n Mostra a lista:\n");

no = head;

mostrarLista(no);

printf("Destruindo a lista:\n");

no = head;

while(no != NULL){

head = head -> prox;

free(no);

no = head;

}

soma = somaLista(head);

qtd = contaLista(head);

media = (float) soma / qtd;

mostrarAcima(head , media);

return 0;

}

int somaLista(struct tipoNo \*parm){

int soma = 0;

struct tipoNo \*aux;

aux = parm;

while(aux != NULL){

soma = soma + aux -> valor;

aux = aux -> prox;

}

return soma;

}

int contaLista(struct tipoNo \*parm){

int qtd = 0;

struct tipoNo \*aux;

aux = parm;

while (aux != NULL){

qtd = qtd + 1;

aux = aux -> prox;

}

return qtd;

}

void mostrarAcima(struct tipoNo \*p, float m){

struct tipoNo \*aux;

aux = p;

while(p!=NULL){

if(aux -> valor > m){

printf("Valor = %d\n", aux -> valor);

}

aux = aux -> prox;

}

}

void mostrarLista(struct tipoNo \*parm){

while(parm != NULL){

printf("\n Valor = %d\n", parm -> valor);

parm = parm -> prox;

}

}

int contemLista(struct tipoNo \*parm){

while(parm != NULL){

}

}