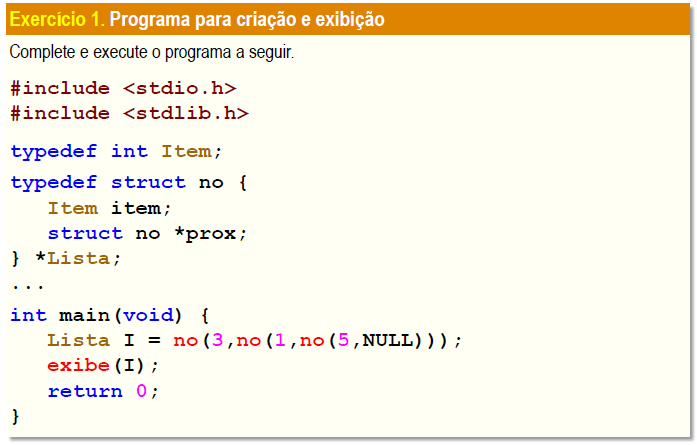
ed-08 – 21/10/2021



#include <stdio.h>

#include <stdlib.h>

typedef int Item;

typedef struct no {

Item item;

struct no \*prox;

} \*Lista;

Lista no(Item x, Lista p) {

Lista n = malloc(sizeof(struct no));

n -> item = x;

n -> prox = p;

return n;

}

void exibe(Lista L) {

while(L != NULL) {

printf("%d\n", L -> item);

L = L -> prox;

}

}

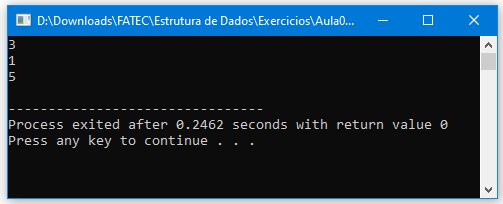
int main(void) {

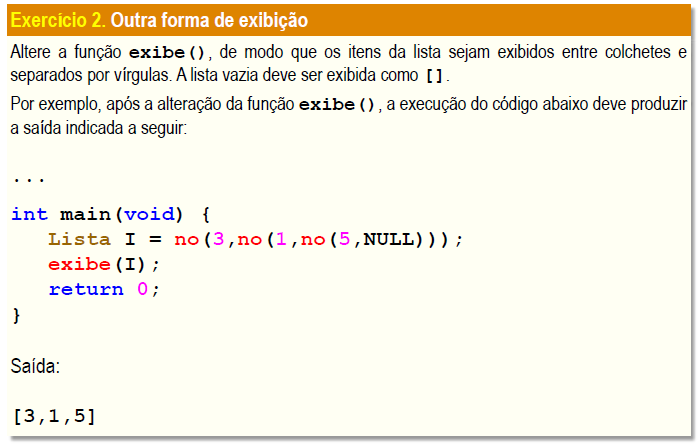
Lista I = no(3, no(1, no(5, NULL)));

exibe(I);

return 0;

}





#include <stdio.h>

#include <stdlib.h>

typedef int Item;

typedef struct no {

Item item;

struct no \*prox;

} \*Lista;

Lista no(Item x, Lista p) {

Lista n = malloc(sizeof(struct no));

n -> item = x;

n -> prox = p;

return n;

}

void exibe(Lista L) {

printf("[");

while(L != NULL) {

printf("%d, ", L -> item);

L = L -> prox;

}

printf("\b\b]");

}

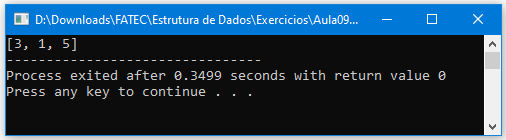
int main(void) {

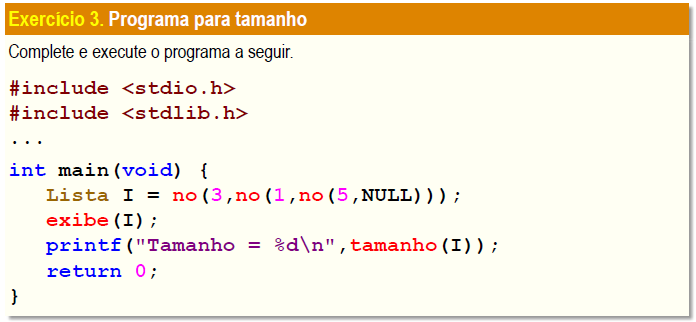
Lista I = no(3, no(1, no(5, NULL)));

exibe(I);

return 0;

}





#include <stdio.h>

#include <stdlib.h>

typedef int Item;

typedef struct no {

Item item;

struct no \*prox;

} \*Lista;

Lista no(Item x, Lista p) {

Lista n = malloc(sizeof(struct no));

n -> item = x;

n -> prox = p;

return n;

}

void exibe(Lista L) {

printf("[");

while(L != NULL) {

printf("%d, ", L -> item);

L = L -> prox;

}

printf("\b\b]");

}

int tamanho(Lista L) {

int t = 0;

while(L) {

t++;

L = L -> prox;

}

return t;

}

int main(void) {

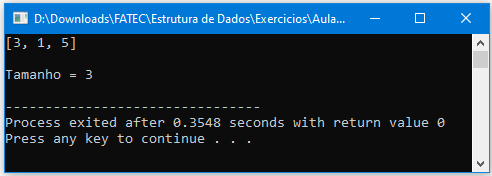
Lista I = no(3, no(1, no(5, NULL)));

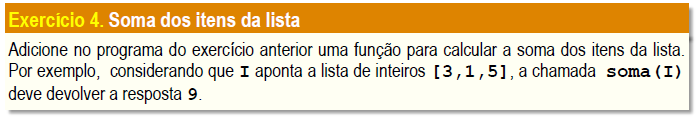
exibe(I);

printf("\n\nTamanho = %d\n", tamanho(I));

return 0;

}





#include <stdio.h>

#include <stdlib.h>

typedef int Item;

typedef struct no {

Item item;

struct no \*prox;

} \*Lista;

Lista no(Item x, Lista p) {

Lista n = malloc(sizeof(struct no));

n -> item = x;

n -> prox = p;

return n;

}

void exibe(Lista L) {

printf("[");

while(L != NULL) {

printf("%d, ", L -> item);

L = L -> prox;

}

printf("\b\b]");

}

int tamanho(Lista L) {

int t = 0;

while(L) {

t++;

L = L -> prox;

}

return t;

}

int soma(Lista L) {

int l = 0, s = 0;

while(L != NULL){

l = L -> item;

s = s + l;

L = L -> prox;

}

return s;

}

int main(void) {

Lista I = no(3, no(1, no(5, NULL)));

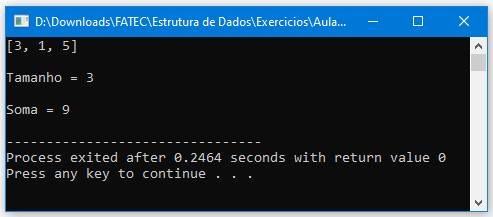
exibe(I);

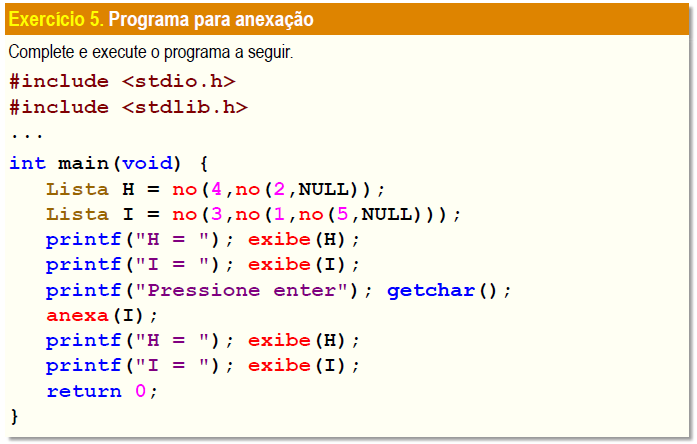
printf("\n\nTamanho = %d\n", tamanho(I));

printf("\nSoma = %d\n", soma(I));

return 0;

}





#include <stdio.h>

#include <stdlib.h>

typedef int Item;

typedef struct no {

Item item;

struct no \*prox;

} \*Lista;

Lista no(Item x, Lista p) {

Lista n = malloc(sizeof(struct no));

n -> item = x;

n -> prox = p;

return n;

}

void exibe(Lista L) {

printf("[");

while(L != NULL) {

printf("%d, ", L -> item);

L = L -> prox;

}

printf("\b\b]");

}

void anexa(Lista \*A, Lista B) {

if(!B) return;

while(\*A)

A = &(\*A) -> prox;

\*A = B;

}

int main(void) {

Lista H = no(4, no(2, NULL));

Lista I = no(3, no(1, no(5, NULL)));

printf("H = "); exibe(H);

printf("\nI = "); exibe(I);

printf("\nPressione enter..."); getchar();

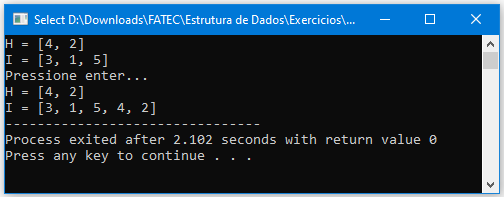
anexa(&I, H);

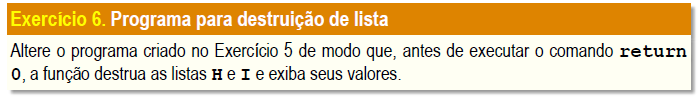
printf("H = "); exibe(H);

printf("\nI = "); exibe(I);

return 0;

}





#include <stdio.h>

#include <stdlib.h>

typedef int Item;

typedef struct no {

Item item;

struct no \*prox;

} \*Lista;

Lista no(Item x, Lista p) {

Lista n = malloc(sizeof(struct no));

n -> item = x;

n -> prox = p;

return n;

}

void exibe(Lista L) {

printf("[");

while(L != NULL) {

printf("%d, ", L -> item);

L = L -> prox;

}

printf("\b\b]");

}

int tamanho(Lista L) {

int t = 0;

while(L) {

t++;

L = L -> prox;

}

return t;

}

void anexa(Lista \*A, Lista B) {

if(!B) return;

while(\*A)

A = &(\*A) -> prox;

\*A = B;

}

void destroi(Lista \*L) {

while(\*L) {

Lista n = \*L;

\*L = n -> prox;

free(n);

}

}

int main(void) {

Lista H = no(4, no(2, NULL));

Lista I = no(3, no(1, no(5, NULL)));

printf("H = "); exibe(H);

printf("\nI = "); exibe(I);

printf("\nPressione enter..."); getchar();

anexa(&I, H);

printf("H = "); exibe(H);

printf("\nI = "); exibe(I);

destroi(&H);

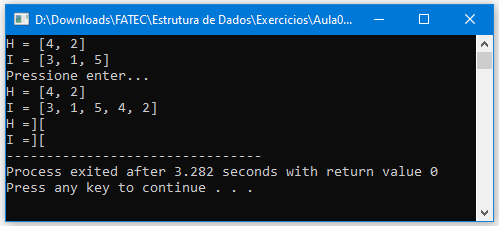
destroi(&I);

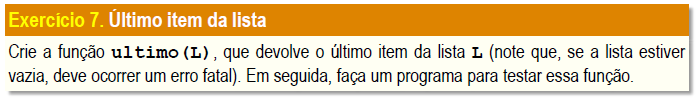
printf("\nH = "); exibe(H);

printf("\nI = "); exibe(I);

return 0;

}





#include <stdio.h>

#include <stdlib.h>

typedef int Item;

typedef struct no {

Item item;

struct no \*prox;

} \*Lista;

Lista no(Item x, Lista p) {

Lista n = malloc(sizeof(struct no));

n -> item = x;

n -> prox = p;

return n;

}

void exibe(Lista L) {

printf("[");

while(L != NULL) {

printf("%d, ", L -> item);

L = L -> prox;

}

printf("\b\b]");

}

int tamanho(Lista L) {

int t = 0;

while(L) {

t++;

L = L -> prox;

}

return t;

}

void anexa(Lista \*A, Lista B) {

if(!B) return;

while(\*A)

A = &(\*A) -> prox;

\*A = B;

}

void destroi(Lista \*L) {

while(\*L) {

Lista n = \*L;

\*L = n -> prox;

free(n);

}

}

int ultimo(Lista L) {

int last = 0;

if(L == NULL) puts("\b Lista vazia.");

while(L != NULL) {

last = L -> item;

L = L -> prox;

}

return last;

}

int main(void) {

Lista Z = NULL;

Lista R = no(3, no(1, no(5, no(7, no(9, NULL)))));

printf("Z = "); exibe(Z);

printf("Ultimo item Z = %d", ultimo(Z));

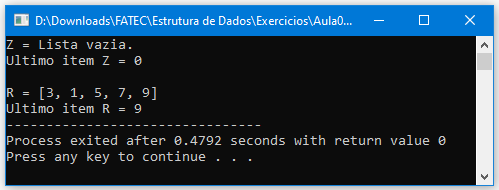
printf("\n\nR = "); exibe(R);

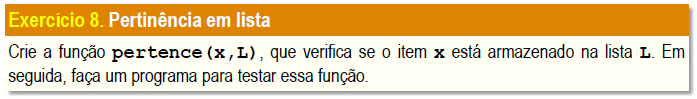
printf("\nUltimo item R = %d", ultimo(R));

destroi(&R);

return 0;

}





#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

typedef int Item;

typedef struct no {

Item item;

struct no \*prox;

} \*Lista;

Lista no(Item x, Lista p) {

Lista n = malloc(sizeof(struct no));

n -> item = x;

n -> prox = p;

return n;

}

void exibe(Lista L) {

printf("[");

while(L != NULL) {

printf("%d, ", L -> item);

L = L -> prox;

}

printf("\b\b]");

}

void destroi(Lista \*L) {

while(\*L) {

Lista n = \*L;

\*L = n -> prox;

free(n);

}

}

bool pertence(int x, Lista L) {

int i = 0;

while(L != NULL){

i = L -> item;

if(x == i){

return 1;

}

L = L -> prox;

}

return 0;

}

int main(void) {

Lista R = no(3, no(1, no(5, no(7, no(9, NULL)))));

int x, ask = 0;

do{

printf("Elemento a procurar: ...");

scanf("%d", &x);

if(pertence(x, R)==1){

printf("O elemento esta armazenado na lista.");

} else {

printf("O elemento NAO esta armazenado na lista.");

}

printf("\n\nDeseja continuar procurando?... \n1 - sim\n2 - nao\n");

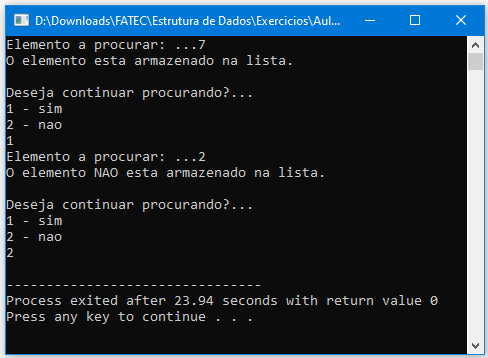
scanf("%d", &ask);

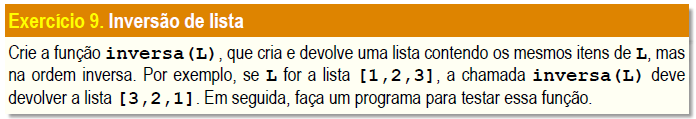
}while(ask==1);

destroi(&R);

return 0;

}





#include <stdio.h>

#include <stdlib.h>

#include "pilhaInt.h"

typedef int Item;

typedef struct no {

Item item;

struct no \*prox;

} \*Lista;

Lista no(Item x, Lista p) {

Lista n = malloc(sizeof(struct no));

n -> item = x;

n -> prox = p;

return n;

}

void exibe(Lista L) {

printf("[");

while(L != NULL) {

printf("%d, ", L -> item);

L = L -> prox;

}

printf("\b\b]");

}

void destroi(Lista \*L) {

while(\*L) {

Lista n = \*L;

\*L = n -> prox;

free(n);

}

}

void inversa(Lista L) {

int last = 0;

Pilha P = pilha(5);

if(L == NULL) puts("\b Lista vazia.");

while(L != NULL) {

last = L -> item;

empilha(last, P);

L = L -> prox;

}

printf("[");

while(!vaziap(P)) {

printf("%d, ", desempilha(P));

}

printf("\b\b]");

}

int main(void) {

Lista R = no(3, no(1, no(5, no(7, no(9, NULL)))));

printf("Lista R = "); exibe(R);

printf("\nLista inversa R = "); inversa(R);

destroi(&R);

return 0;

}

