

1ª Trabalho

Curso: Engenharia de Computação
Disciplina: Estruturas de Dados
Prof. Dr. Jarbas Joaci de Mesquita Sá Junior
Universidade Federal do Ceará – UFC/Sobral

Entrega: 10/12/2015 via e-mail para jarbas_joaci@yahoo.com.br – **Obs 1:** Não receberei o trabalho após a data mencionada. **Obs 2:** o trabalho deverá ser feito em **dupla** e apresentado em 14/12/2015 (cada membro da banca **obrigatoriamente** deverá participar da apresentação para obtenção de nota)

Nome: _____ Data ____/____/2015

1ª) Implemente o Tipo Abstrato de Dados (TAD) “lista.h” (ver slides sobre Listas Encadeadas) e acrescente as seguintes funções:

a) função que tenha como valor de retorno o comprimento de uma lista encadeada, isto é, que calcule o número de nós de uma lista. Essa função deve obedecer ao protótipo:

```
int comprimento(Lista* l);
```

b) função para retornar o número de nós da lista que possuem o campo `info` com valor maior que `n`. Essa função deve obedecer ao protótipo:

```
int maiores(Lista* l, int n);
```

c) função para somar os valores do campo `info` de todos os nós. Essa função deve obedecer ao protótipo:

```
int soma(Lista* l);
```

d) função para retornar o número de nós da lista que possuem o campo `info` com número primo. Essa função deve obedecer ao protótipo:

```
int primos(Lista* l);
```

A seguir, execute o seguinte programa.

```
#include <stdio.h>
#include<stdlib.h>
#include "lista.h"

int main(void){
    int maior;
    Lista* l = lst_cria();
    l = lst_insere(l,10);
    l = lst_insere(l,13);
    l = lst_insere(l,25);
    l = lst_insere(l,29);
```

```

l = lst_insere(l,40);
l = lst_remove(l,10);
lst_imprime(l);
lst_imprime_invertida_rec(l);
maior = maiores(l,23);
printf("Número de nós com info maior que 23 é %d\n",maior);
printf("O comprimento da lista é %d\n",comprimento(l));
printf("Soma dos valores dos nós %d\n",soma(l));
printf("Número de nós com valores primos é %d\n",primos(l));
lst_libera(l);

system("PAUSE");
return 0;
}

```

2ª) Implemente o Tipo Abstrato de Dados (TAD) “pilha.h” usando Listas Encadeadas (ver slides sobre Pilhas) e acrescente as seguintes funções:

a) função que receba uma pilha como argumento e retorne o valor armazenado em seu topo. Essa função deve obedecer ao protótipo:

```
int topo(Pilha* p);
```

Obs: Essa função não altera a pilha, apenas retorna uma cópia do valor armazenado no seu topo.

b) função para retornar o número de elementos da pilha que possuem o campo info com valor ímpar. Essa função deve obedecer ao protótipo:

```
int impares(Pilha* p);
```

A seguir, execute o seguinte programa.

```

#include <stdio.h>
#include <stdlib.h>
#include "pilha.h"

int main(void){
    int a;
    Pilha* p1 = pilha_cria();
    pilha_push(p1,10);
    pilha_push(p1,20);
    pilha_push(p1,25);
    pilha_push(p1,30);
    a = pilha_pop(p1);
    pilha_imprime(p1);
    printf("Elemento no topo da pilha p1 é %d\n",topo(p1));
    printf("Qde elems impares na pilha é %d\n",impares(p1));
    pilha_libera(p1);
}

```

```

    system("PAUSE");
    return 0;
}

```

3ª) Implemente o Tipo Abstrato de Dados (TAD) “fila1.h”(implementação com vetor) e “fila2.h” (implementação com listas encadeadas) e acrescente as seguintes funções:

a) função para retornar o número de elementos da fila com valor maior que n. Essa função deve obedecer ao protótipo:

```
int qtd_maior(Fila* f, int n);
```

b) função que inverta a ordem dos elementos da fila. Essa função deve obedecer ao protótipo:

```
void inverte(Fila* f);
```

c) função para retornar o número de elementos da fila que possuem o campo info com valor par. Essa função deve obedecer ao protótipo:

```
int pares(Fila* f);
```

A seguir, execute o seguinte programa com as TAD's “fila1.h” e “fila2.h”

```

#include <stdio.h>
#include<stdlib.h>
#include "fila1.h" //executar também com "fila2.h"

int main(void){
    int a, qtd;
    Fila* f = fila_cria();
    fila_insere(f,1);
    fila_insere(f,2);
    fila_insere(f,3);
    fila_insere(f,4);
    fila_insere(f,5);
    inverte(f);
    a = fila_remove(f);
    fila_imprime(f);
    qtd=qtd_maior(f,2);
    printf("'Núm. de elem. maiores que 2: %d\n'",qtd);
    printf("'Qtd. elementos pares na fila: %d\n'",pares(f));
    fila_libera(f);

    system("'PAUSE'");
    return 0;
}

```

4ª) Implemente os Tipos Abstratos de Dados (TAD) “lista_circular.h” e “lista_dupla.h” (ver slides sobre Listas Circulares e Duplamente Encadeadas).

A seguir, execute os seguintes programas.

```
#include <stdio.h>
#include<stdlib.h>
#include "lista_circular.h"

int main(void){
    int maior;
    Lista* l = lst_circ_cria();
    l = lst_circ_insere(l,10);
    l = lst_circ_insere(l,13);
    l = lst_circ_insere(l,25);
    l = lst_circ_insere(l,29);
    l = lst_circ_insere(l,40);
    l = lst_circ_remove(l,10);
    lst_circ_imprime(l);
    lst_circ_libera(l);

    system("PAUSE");
    return 0;
}
```

```
#include <stdio.h>
#include<stdlib.h>
#include "lista_dupla.h"

int main(void){
    int maior;
    Lista* l = lst_dupl_cria();
    l = lst_dupl_insere(l,10);
    l = lst_dupl_insere(l,13);
    l = lst_dupl_insere(l,25);
    l = lst_dupl_insere(l,29);
    l = lst_dupl_insere(l,40);
    l = lst_dupl_remove(l,10);
    lst_dupl_imprime(l);
    lst_dupl_libera(l);

    system("PAUSE");
    return 0;
}
```

Boa Diversão!!!