1º Trabalho

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

Entrega: 03/05/2023 via e-mail para jarbas joaci@yahoo.com.br

Obs. 1: Não receberei o trabalho após a data mencionada.

- O trabalho é **opcional**, deverá ser feito **individualmente** e valerá, no máximo, **1,0** ponto.
- Preferencialmente fazer o trabalho usando a IDE Dev-C++.;
- Enviar **todos** os arquivos do projeto, exceto os executáveis (.exe). Organizar os arquivos nas pastas q1, q2 e q3.
- O uso da diretiva #include sem um header file (.h) implicará nota zero no código. Por exemplo, não usar #include "nomearquivo.c"
- 1^a) Implemente o Tipo Abstrato de Dados (TAD) "lista.h" (ver slides sobre Lista Encadeada) e acrescente as seguintes funções:
- a) função que calcule o número de nós de uma lista. Essa função deve obedecer ao protótipo:

```
int comprimento(Lista* 1);
```

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

```
int menores(Lista* 1, int n);
```

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

```
int soma(Lista* 1);
```

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* 1);
```

e) função para criar uma lista que é a concatenação de uma lista 12 no final de uma lista 11. Essa função deve obedecer ao protótipo:

```
Lista* lst conc(Lista* 11, Lista* 12);
```

Obs. as listas 11 e 12 permanecem inalteradas após a execução da função.

f) função que faça a diferença de duas listas L_1 e L_2 (ou seja, que retire de L_1 os elementos que estão em L_2). Por exemplo, se lista $L_1 \rightarrow 3 \rightarrow 7 \rightarrow 2 \rightarrow 4 \rightarrow //$ e lista L_2

 \rightarrow 7 \rightarrow 9 \rightarrow //, a lista L₁ modificada deve ser L₁ \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow //. Essa função deve obedecer ao protótipo:

```
Lista* lst diferenca (Lista* 11, Lista* 12);
A seguir, execute o seguinte programa.
#include <stdio.h>
#include <stdlib.h>
#include "lista.h"
int main(void) {
  Lista* 11 = 1st cria();
  l1 = lst insere(l1, 6);
  11 = 1st insere(11,13);
  11 = 1st insere(11, 45);
  11 = 1st insere(11,28);
  11 = 1st insere(11,41);
  11 = 1st remove(11, 25);
  11 = 1st remove rec(11,41);
  lst imprime(l1);
  lst imprime_invertida_rec(11);
  printf("Num. nós c/ info < que 30: %d\n", menores(11,30));</pre>
  printf("O comprimento da lista é %d\n", comprimento(l1));
  printf("Soma dos valores dos nós %d\n", soma(11));
  printf("Num. nós com val. primos é %d\n",primos(l1));
  Lista* 12 = 1st cria();
  12 = 1st insere(1,28);
  12 = 1st insere(1, 45);
  12 = 1st insere(1, 130);
  Lista* 13=1st conc(11,12);
  lst imprime(13);
  11=1st diferenca(11,12);
  lst imprime(l1);
  lst libera(11);
  lst libera(12);
  lst libera(13);
  system("PAUSE");
  return 0;
}
```

- 2^a) Implemente o Tipo Abstrato de Dados (TAD) "pilha.h" usando Vetor (ver slides sobre Pilha) 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 que retorne o número de elementos da pilha que possuem o campo info com valor ímpar. Essa função deve obedecer ao protótipo:

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

c) função que verifique quais são os elementos pares de uma pilha p1 e que os empilhe em **ordem crescente** em uma pilha p2. Essa função deve obedecer ao protótipo:

```
Pilha* empilha pares(Pilha* p1, Pilha* p2);
```

Obs. a pilha p1 permanece inalterada após a execução da função.

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: %d\n",topo(p1));
 printf("Qde elems impares na pilha p1: %d\n",impares(p1));
 Pilha* p2 = pilha cria();
 pilha push (p2,3);
 pilha push (p2,4);
 p2 = empilha pares(p1, p2);
 pilha imprime(p2);
 pilha libera(p1);
 pilha libera(p2);
 system("PAUSE");
 return 0;
}
```

- 3ª) Implemente o Tipo Abstrato de Dados (TAD) "fila.h" usando Lista Encadeada (ver slides sobre Fila) 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 crie uma fila com os elementos da fila f na ordem inversa. Essa função deve obedecer ao protótipo:

```
Fila* 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.

```
#include <stdio.h>
#include<stdlib.h>
#include "fila.h"
int main(void) {
 int a, qtd;
 Fila* f1 = fila cria();
 fila insere(f1,11);
 fila insere(f1,12);
 fila insere(f1,13);
 fila insere(f1,14);
 fila insere(f1,15);
 a = fila remove(f1);
 printf(''Valor removido da fila f1: %d\n'',a);
 fila imprime(f1);
 Fila* f2=inverte(f1);
 fila imprime (f2);
 qtd=qtd maior(f1,13);
 printf(''Núm. de elem. maiores que 13 em f1: %d\n'',qtd);
 printf(''Qtd. elem. pares na fila f1: %d\n'',pares(f1));
 fila libera(f1);
 fila libera(f2);
 system("PAUSE");
 return 0;
}
```