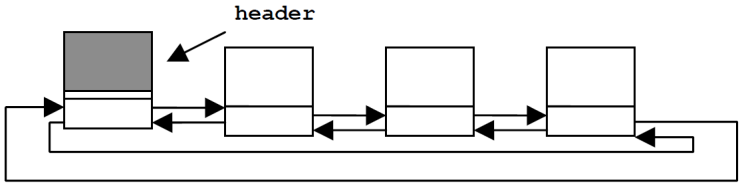
UNIVERSIDADE DE SÃO PAULO - ICMC
Departamento de Ciências de Computação
SCC0202 – Algoritmos e Estruturas de Dados I - 2º Sem /2022
Prof. Rudinei Goularte
Sala 4-229.

1. Defina Lista Linear. Discuta os diferentes tipos de implementação possíveis em termos de organização e alocação de memória.

2. Desenvolva uma função void `inverter (pilha *P);` para inverter a posição dos elementos de uma pilha dinâmica encadeada P. Você pode criar pilhas auxiliares, se necessário, mas o resultado precisa ser dado na pilha P. Considere que a pilha guarda elementos do tipo *int*.

3. Você acaba de ser contratado pela Serasa - São Carlos (Experian). Sua missão atual: desenvolver o cadastro de inadimplentes. A estrutura de dados, por razões históricas, deve ser uma **Lista**. Sabe-se que: desempenho é crítico; memória é crítica; o volume de consultas é alto; o volume diário de pessoas inseridas e removidas do cadastro é alto. Dado esse contexto, como você implementaria essa Estrutura de Dados para que seja o mais eficaz e eficiente possível? Justifique sua resposta em termos de eficiência, eficácia, complexidade algorítmica, estratégias de implementação da estrutura de dados e das operações do TAD lista.

4. Uma lista circular duplamente encadeada com nó de cabeçalho (header) pode ser esquematizada como na figura abaixo. Suponha que o header guarde o número de elementos da lista.



a) Defina o TAD e implemente uma lista assim, com as seguintes operações:
- cria (L)
- IsEmpty(L)
- IsFull(L)

b) Em uma lista dinâmica duplamente encadeada, a operação de remoção possui exceções, isto é, não possui tratamento igual dos nós para todos os casos. Contudo, se a lista for também circular, como no caso deste exercício, essas exceções são evitadas. Explique!

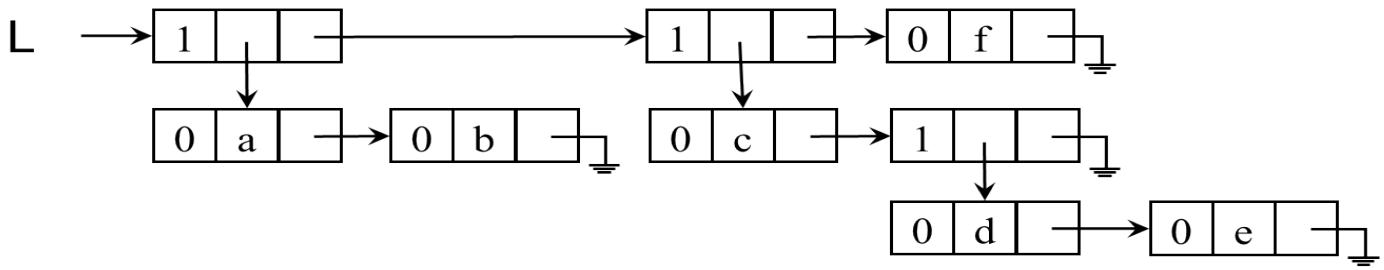
5. Explique que exceções na operação de remoção são evitadas ao se implementar uma lista circular em lugar de uma lista sequencial.

6. Seja uma lista generalizada como definido abaixo:

```
(generalizada.h)
#ifndef LISTAGEL_H
#define LISTAGEL_H
  #define TRUE 1
  #define FALSE 0
  #define boolean int

  #include "item.h"

  typedef struct no_ NO;
  typedef struct lista_ LISTA;
  ...
#endif
```

```
(generalizada.c)
...
#include "generalizada.h"

struct lista_{
       NO *inicio;
};

struct no_{
              union {
                     ITEM *atomo;
                     struct no_ *sublista;
              } info;
              int tipo;
              NO *prox;
};
...
```

Desenvolva uma função com o seguinte protótipo int lista_imprime (LISTA *lista); que imprime todos os elementos (átomos) da lista. Exemplo: L = ((a,b), (c, (d, e)), f):



7. Sejam os TADs **Ponto** e **Circulo** como definido abaixo.

a) Desenvolva um programa cliente (main.c) que, pela ordem: crie um ponto **p** e um círculo **c** definidos pelo usuário (stdin); chame a função `void distancia(Ponto *p, Circulo *r);` para calcular se **p** é interior ou não a **c** e imprimir o resultado.

b) Desenvolva a função *distancia* justificando onde ela deve ser implementada (em que arquivo .c?). Se necessário, desenvolva (implemente) as alterações que julgar pertinentes nos TADs, **sempre justificando**.
Obs.: Um ponto **p** é interior a um círculo **c** se a distância entre **p** e o ponto que define o círculo (`ponto_c`) é menor que o raio de **c** (`raio`). A distância entre dois pontos é dada pela equação (1):

$$(1) d = \sqrt{(x2-x1)^2+(y2-y1)^2}$$

```
(ponto.h)
#ifndef PONTO_H
  #define PONTO_H
  typedef struct ponto_ PONTO;
  PONTO *ponto_criar (float x, float y);
  void ponto_apagar (PONTO *p);
  boolean ponto_set (PONTO *p, float x, float y);
#endif

(ponto.c)
...
#include "ponto.h"
struct ponto_{
  float x;
  float y;
};
...
```

```
(circulo.h)
#ifndef CIRCULO_H
 #define CIRCULO_H
 #include "ponto.h"
 typedef struct circulo_ CIRCULO;
 CIRCULO *circulo_criar (float c,float y, float raio);
 void circulo_apagar (CIRCULO *circulo);
 float circulo_area (CIRCULO *circulo);
#endif

(circulo.c)
...
#include "circulo.h"
#define PI 3.14159
struct circulo_{
   PONTO *ponto_c;
   float raio;
};
...
```