

A2 – Onde está o loop?

Listas simplesmente encadeadas

Existem rumores de listas simplesmente encadeadas sendo adulteradas através de *loops*. Em geral, uma lista simplesmente encadeada tem seu último elemento, no campo que faz a ligação entre os nós, apontando para NULL. Sobretudo, há suspeitas de que algumas listas, com o último elemento sendo ligado a algum outro elemento da lista, criando assim *loops*. Os *loops* são indesejados, uma vez que a lista ficaria circulando eternamente, pois não saberíamos mais onde está o fim. Veja a Figura 1 para ter uma visão esquemática sobre listas deste tipo.

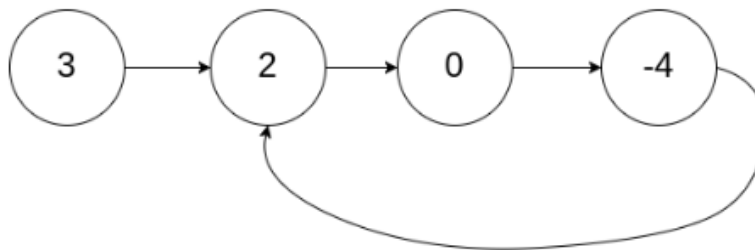


Figura 1: A próxima ligação do último elemento (-4) está apontando para 2, gerando assim um loop.

A divisão **EDSecurity I** está lhe contratando para implementar uma função que receba o nó descritor de uma lista simplesmente encadeada (o nó descritor possui apenas o ponteiro para o início da lista) e descubra duas informações: se há *loop* e em qual altura está este *loop*. No exemplo da Figura 1, o *loop* está na altura 2, pois ocupa o segundo lugar da lista, começando a contagem do início. Portanto, para listas sem *loop*, deve-se imprimir a informação 0.

A sugestão aqui é fazer uma lista simplesmente encadeada que armazene a posição e o endereço do nó de maneira única.

Entrada

A função a ser implementada deve receber apenas a lista e devolver a posição do loop. Para que você possa testar, desenvolvemos a entrada da seguinte maneira. A primeira linha é composta pelo número de testes. Cada teste, possui duas linhas. A primeira linha tem duas informações: a quantidade de dados da lista e onde ocorre o *loop* (esta informação não deve ser usada para compor



a solução do problema). A segunda linha de cada entrada são os números inteiros que compõem a lista. Para cada entrada, deve-se produzir a altura que foi detectado o loop (o valor zero indica que a lista não possui loops).

```
1                2
4 2
1 2 3 4
```

Composição do projeto

O projeto possui cinco arquivos de código e um arquivo de teste de entrada (input.txt). Sua missão é codificar a função `int TLinkedList_hasLoop(TLinkedList* list);` que se encontra no arquivo **TLinkedListUtils.c**. Os únicos arquivos que devem ser alterados são o **TLinkedListUtils.c** e o **TLinkedListUtils.h**. Caso você queira implementar as funções de uma lista simplesmente encadeada ou outro tipo de mecanismo, deve-se usar o **TLinkedListUtils.h** para compor a interface das funções e o **TLinkedListUtils.c** para as implementações. Todas as funções eventualmente criadas por você, devem ser chamadas a partir do corpo da função `TLinkedList_hasLoop`.

O programa será compilado da seguinte forma:

```
gcc main.c TLinkedList.c TLinkedListUtils.c -o A2
```

A execução dos testes:

```
./A2.exe <input.txt
```

Porém usando o `input.txt` para testes que tenho aqui. Ao final, checo se diferenças entre as entradas e saídas que possuo como gabarito e atribuo sua nota. Se acertar parcial, vai proporcional à quantidade de casos corretos. Após estes testes, a nota só é atribuída depois que o código é revisado.

O que deve ser enviado

Crie uma pasta com seu nome completo, contendo apenas os arquivos: **TLinkedListUtils.h** e **TLinkedListUtils.c**. Compacte a pasta com seu nome usando o tipo zip e submeta o arquivo na plataforma moodle, no local indicado.

Boa diversão!