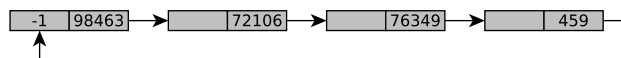


### Exercícios: listas encadeadas

1. Utilizando uma lista encadeada, implemente uma **fila** de prioridade única:
  - (a) Cria **Tipo Abstrato de Dado** que contenha: um campo inteiro para identificador e outro para prioridade. Este tipo de dados que será adicionado na lista.
  - (b) Prioridade é inversa: 1 significa prioridade máxima.
  - (c) A fila de prioridade deve manter sempre na cabeça o elemento com maior prioridade.
  - (d) Elabore um programa para provar a funcionalidade da sua implementação com números aleatórios:  
Alimente a semente do rand() com:  
`srand ( getpid() ^time(NULL));`
2. O hardware possui capacidade limitada para realizar operações aritméticas com números muito longos. Para somar dois inteiros longos, seus dígitos são atravessados da direita para a esquerda, e os dígitos correspondentes e um possível transporte da soma dos dígitos anteriores são acrescidos. Isto pode ser implementado através de listas encadeadas, de modo que o primeiro nó da lista contenha o número menos significativo e o último nó contenha o número mais significativo. Veja o exemplo na figura abaixo do número 984637210676349459.



Neste exemplo, cada nó armazena no máximo 5 dígitos (max = 99999).

Utilizando estas informações como base, implemente:

- (a) Um **Tipo Abstrato de Dado** que represente números longos divididos em 5 dígitos decimais.
- (b) Uma função *addint* que some estes dois números.
- (c) Uma função *mulint* que multiplique estes dois números.

---

#### Observações para ambos os exercícios:

1. Utiliza SEMPRE tipos abstratos de dados.
2. As listas encadeadas devem ser GENÉRICAS (void \*).