



Teknik Informatika Kampus 3 Nganjuk
Jurusan Teknologi Informasi

Politeknik Negeri
Jember

Doubly Linked List

Struktur Data

Ulfa Emi Rahmawati, S.Kom., M.Kom.

Definisi

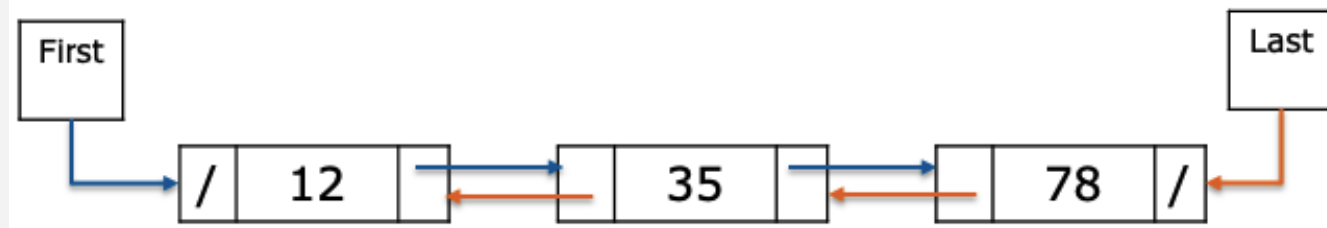
Doubly linked list

Doubly linked list

- Doubly Linked List merupakan suatu linked list yang memiliki dua variabel pointer.
- Satu menunjuk ke node selanjutnya (*next) dan pointer yang lain menunjuk ke node sebelumnya (*prev).
- Setiap head dan tail-nya juga menunjuk ke NULL.
- Dibandingkan dengan singly linked list, doubly-linked list membutuhkan variabel pointer lebih banyak.

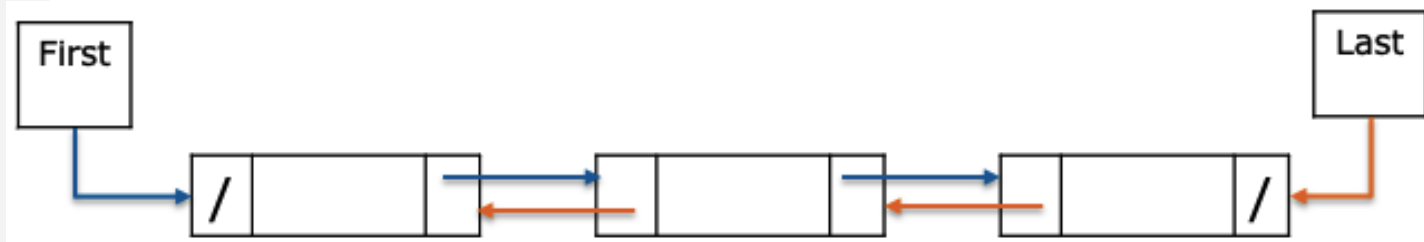
Doubly linked list

- Linked list dengan 2 pointer
- Menghubungkan node sebelumnya dan selanjutnya

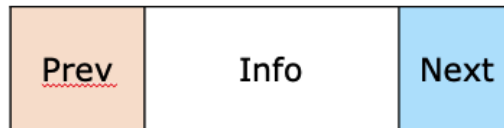


Struktur

- ▶ Memiliki 2 head pointer

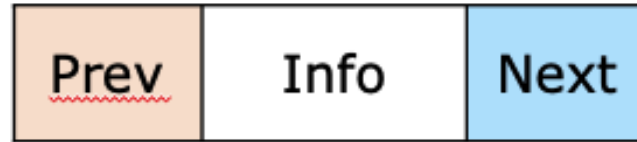


- ▶ Setiap node dibagi menjadi 3 bagian



ADT Element List

- Ada 2 pointer untuk menunjuk node berikutnya dan sebelumnya



- Ada 2 head untuk menunjuk node pertama dan terakhir dalam list



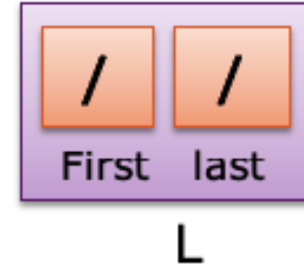
L

Membuat node baru

Algoritma:

First (L) \leftarrow Null

Last (L) \leftarrow Null



- Last(L) adalah kata kunci untuk mengetahui node terakhir dari list L
- Pada pembuatan node baru, tidak ada node, jadi first(L) dan last(L) adalah Null

Membuat node baru

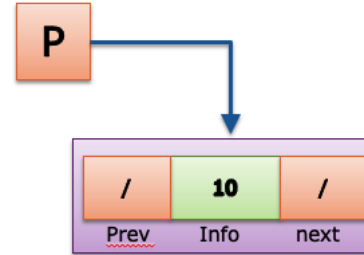
Algoritma:

Alokasikan (P)

$\text{Next}(P) \leftarrow \text{Null}$

$\text{Prev}(P) \leftarrow \text{Null}$

$\text{Info}(P) \leftarrow 10$



- Prev(P) adalah kata kunci untuk mengetahui node sebelumnya dari node setelahnya yang ditunjuk oleh P
- Pada pembuatan node baru, atur elemen Next dan Prev = Null

Keywords

- ▶ First(X)
- ▶ Last(X)
 - Pilih node terakhir dari list X
- ▶ Next(Y)
- ▶ Prev(Y)
 - Pilih node sebelumnya dari list Y
- ▶ Info(Y)

Insertion

Penyisipan node baru

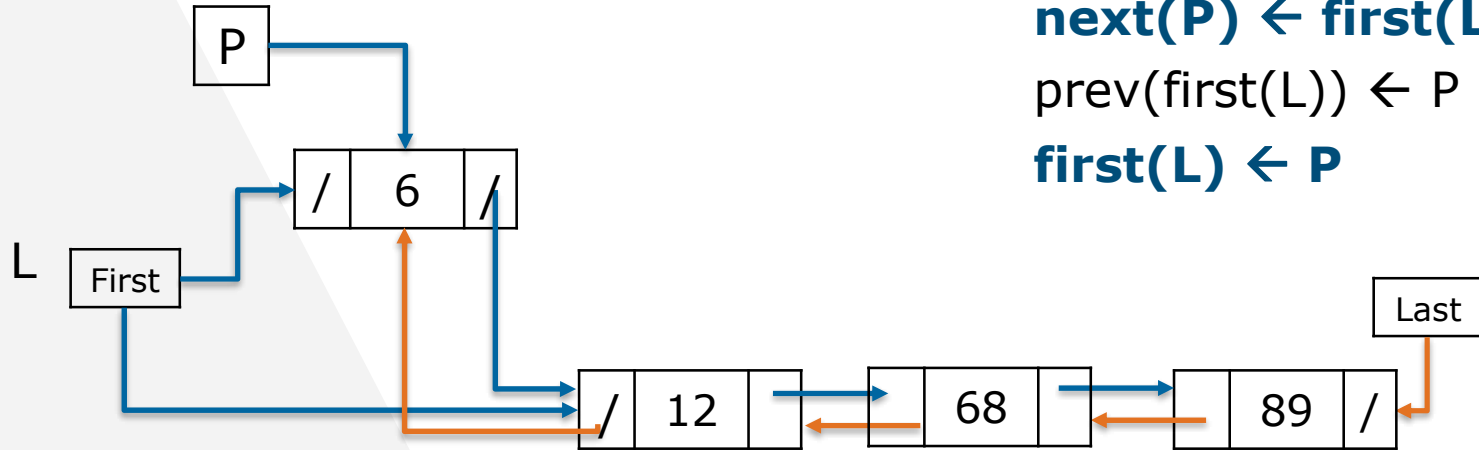
Insertion

Merupakan penyisipan node baru pada suatu linked list

- Insertion di awal
- Insertion di akhir
- Insertion setelah node tertentu atau diantara node

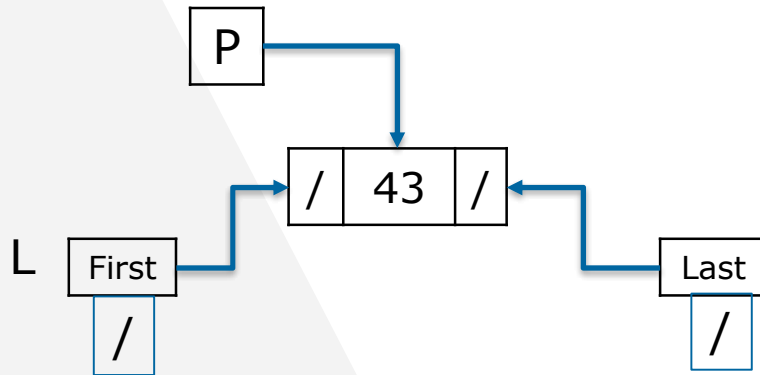
Insertion di awal

Algorithm

$$\text{next(P)} \leftarrow \text{first(L)}$$
$$\text{prev}(\text{first(L)}) \leftarrow P$$
$$\text{first(L)} \leftarrow P$$


Insertion node pertama

Algorithm



$\text{next}(P) \leftarrow \text{first}(L)$

$\text{prev}(\text{first}(L)) \leftarrow P$

ERROR

$\text{first}(L) \leftarrow P$

$\text{first}(L) \leftarrow P$

$\text{last}(L) \leftarrow P$

Insertion di awal

Algorithm

If (first(L) \neq Nil and last(L) \neq Nil)

next(P) \leftarrow first(L)

prev(first(L)) \leftarrow P

first(L) \leftarrow P

else

first(L) \leftarrow P

last(L) \leftarrow P

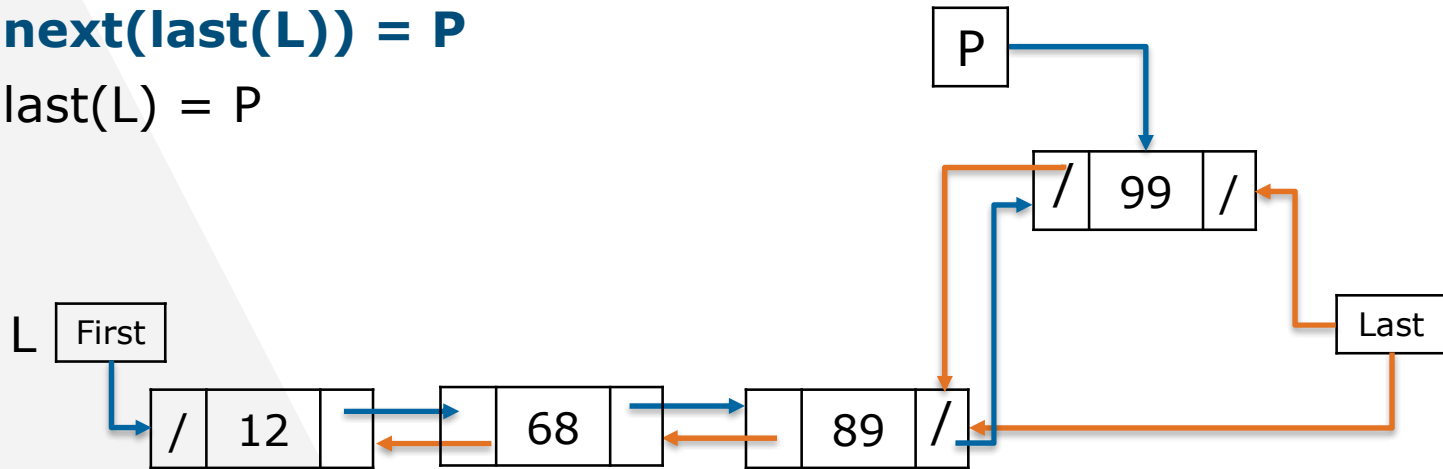
Insertion di akhir

Algorithm

$\text{prev}(P) = \text{Last}(L)$

$\text{next}(\text{last}(L)) = P$

$\text{last}(L) = P$



Insertion setelah node tertentu

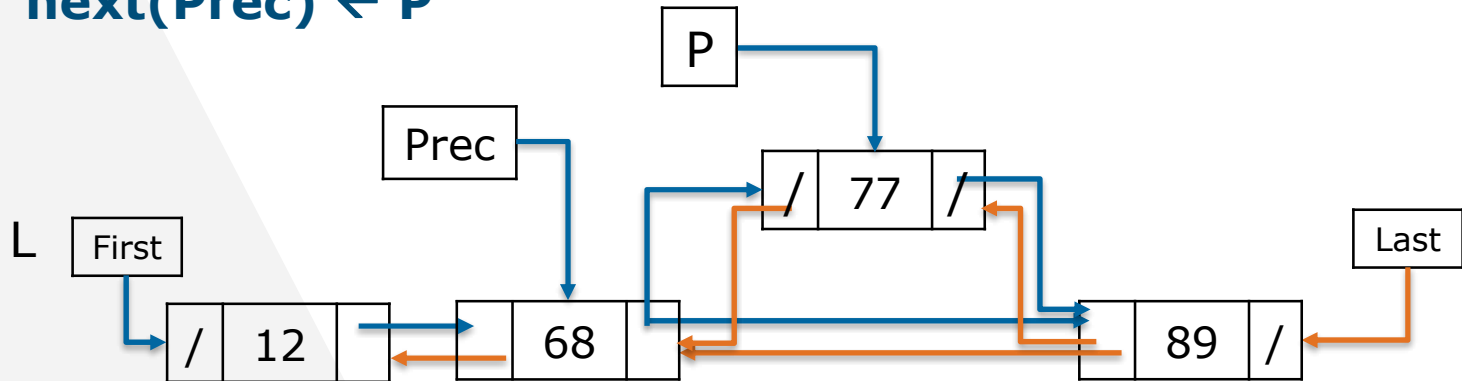
Algorithm

$\text{next(P)} \leftarrow \text{next(Prec)}$

$\text{prev(P)} \leftarrow \text{Prec}$

$\text{prev}(\text{next}(\text{prec})) \leftarrow \text{P}$

$\text{next(Prec)} \leftarrow \text{P}$



Deletion

Penghapusan node tertentu

Deletion

Merupakan penghapusan node tertentu pada suatu linked list

- Deletion di awal
- Deletion di akhir
- Deletion setelah node tertentu

Deletion di awal

Algorithm

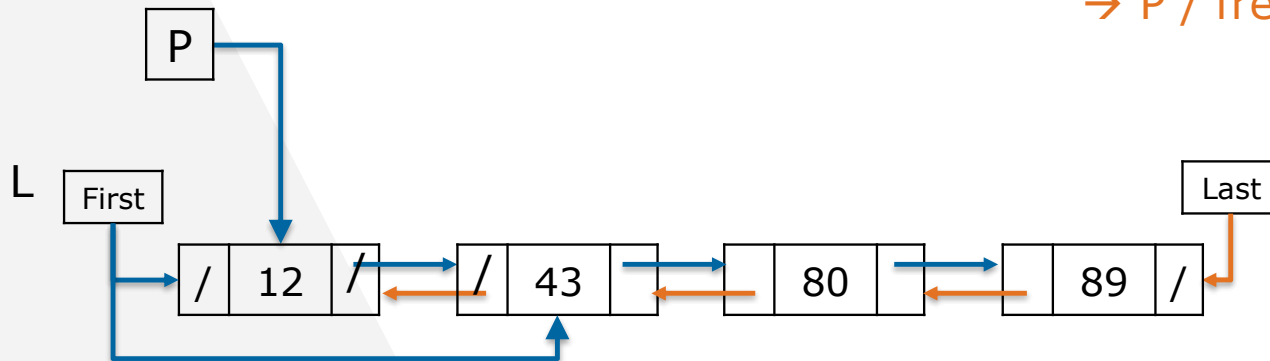
$P \leftarrow \text{first}(L)$

$\text{first}(L) \leftarrow \text{next}(P)$

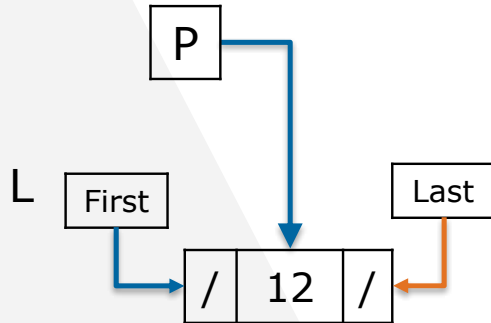
$\text{next}(P) \leftarrow \text{Nil}$

$\text{prev}(\text{first}(L)) \leftarrow \text{Nil}$

$\rightarrow P / \text{free}(P)$



Deletion di awal



Algorithm

$P \leftarrow \text{first}(L)$

$\text{first}(L) \leftarrow \text{next}(P)$

$\text{next}(P) \leftarrow \text{Nil}$

ERROR

$\text{prev}(\text{first}(L)) \leftarrow \text{Nil}$

$\text{first}(L) \leftarrow \text{Nil}$

$\text{last}(L) \leftarrow \text{Nil}$

$\rightarrow P / \text{free}(P)$

Deletion di awal

Algorithm

$P \leftarrow \text{first}(L)$

If ($\text{first}(L) \neq \text{last}(L)$)

$\text{first}(L) \leftarrow \text{next}(P)$

$\text{next}(P) \leftarrow \text{Nil}$

$\text{prev}(\text{first}(L)) \leftarrow \text{Nil}$

else

$\text{first}(L) \leftarrow \text{Nil}$

$\text{last}(L) \leftarrow \text{Nil}$

→ P / $\text{free}(P)$

Deletion di akhir

Algorithm

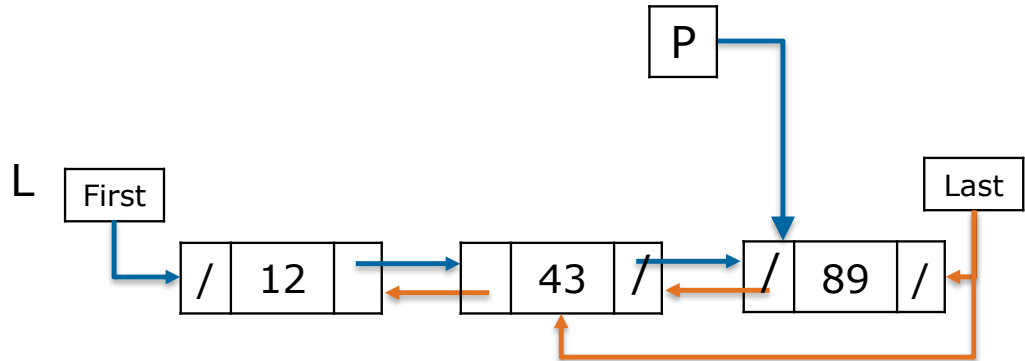
$P \leftarrow \text{Last}(L)$

$\text{Last}(L) \leftarrow \text{prev}(\text{last}(L))$

$\text{prev}(P) \leftarrow \text{Nil}$

$\text{next}(\text{last}(L)) \leftarrow \text{Nil}$

→ P / free(P)



Deletion setelah node tertentu

Algorithm

P \leftarrow next(Prec)

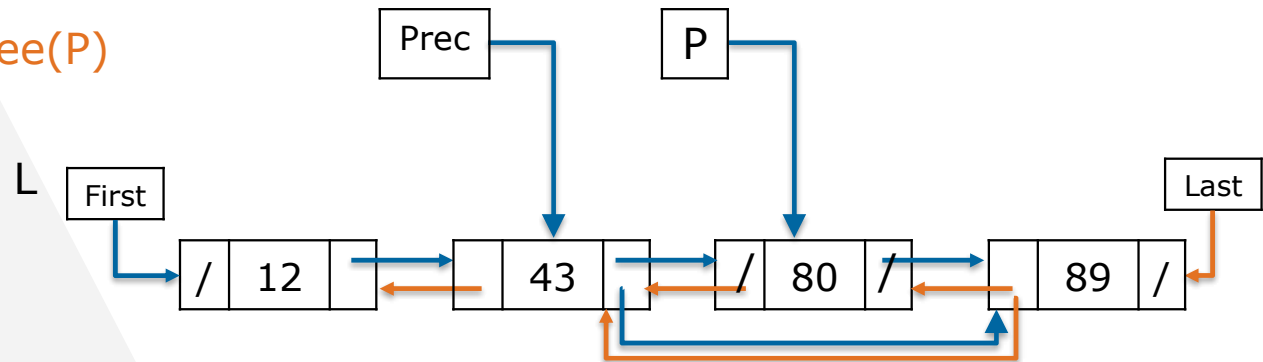
Next(Prec) \leftarrow next(P)

Prev(next(P)) \leftarrow Prec

Prev(P) \leftarrow Nil

Next(P) \leftarrow Nil

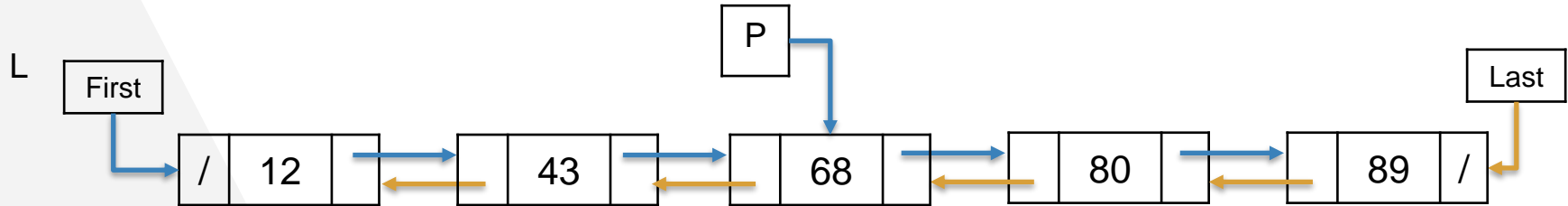
\rightarrow P / free(P)



Exercise

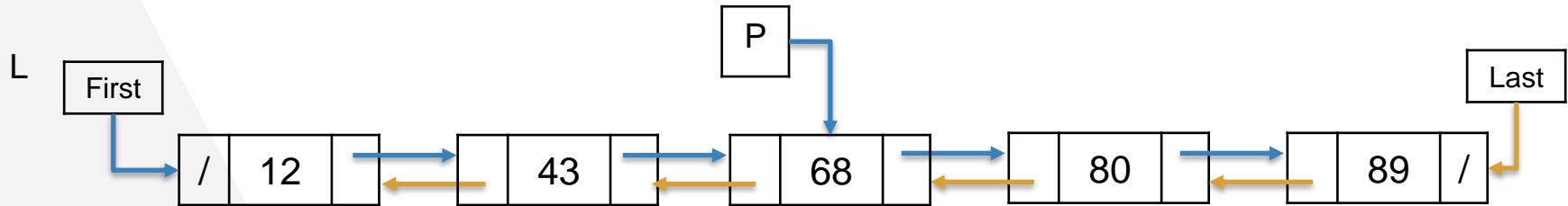
Latihan bersama

Exercise



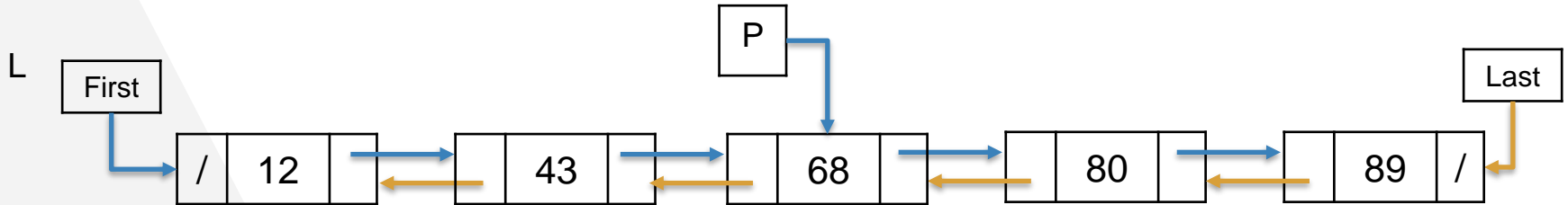
Draw the Pointer
$Q \leftarrow \text{prev}(\text{prev}(P))$
$S \leftarrow \text{next}(\text{next}(\text{prev}(\text{next}(P))))$

Exercise



Task	Output	Answer
Access info the first element of the list	12	
Access info of the second element of the list	43	

Exercise



Operation	Answer
$\text{Info}(\text{Prev}(\text{next}(\text{prev}(\text{last}(\text{L}))))$	
$\text{Info}(\text{Next}(\text{next}(\text{P})))$	
$\text{Info}(\text{next}(\text{P})) + \text{info}(\text{prev}(\text{last}(\text{L})))$	
$\text{Info}(\text{first}(\text{L})) - \text{info}(\text{next}(\text{next}(\text{P})))$	

THANKS!

Any questions?