

EA CTD 7

Liste doublement chaînée

→ pointeur sur head

→ dans chaque cellule:

- la clé
- pointeur vers next
- pointeur vers previous

Exercice 2

À la fin la pile est: L

Output des pop: AINRTSUTOiFTS

Exercice 1

Insérer(x, k, z):

```

nœud courant = x.head
while(courant != z){
    courant = courant.next
}
nœud n = nouveau nœud(k)
n.prev = x.prev
n.next = z
if(x.prev != null){
    x.prev.next = n
}
x.prev = z

```

Exercice 3

```

1) boolean empty(P){
    return P.head == null;
}

void push(x, P){
    addFirst(x);
}

```

```

pop(P){
    remove(0);
}

```

2)

```
class Stack {  
    int[] T;  
    int top;
```

```
    public Stack(int n) {
```

```
        T = new int[n];
```

```
    }
```

```
    public boolean isEmpty() {
```

```
        return top == -1;
```

```
    }
```

```
    public void push(int k) {
```

```
        if (top != T.length - 1) {
```

```
            T[top + 1] = k;
```

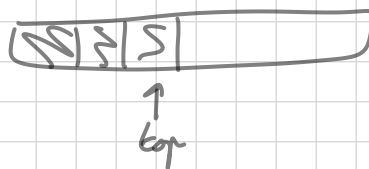
```
            top ++;
```

```
        }
```

```
    public int pop() {
```

```
        top --; // on n'élimine pas la  
        return T[top + 1]; // valeur du tableau
```

```
    }
```



Exercice 4

```
1) public static void reverseZDE(Liste l){  
    if (l.head != null) l.head = reverseAux(l.head);  
}  
  
public static Cell reverseAux(Cell c){  
    Cell tmp;  
    if (c.next != null){  
        tmp = c.next;  
        c.next = c.prev;  
        c.prev = tmp;  
        return reverseAux(tmp);  
    } else { // fin de la Liste  
        tmp = c.next;  
        c.next = c.prev;  
        c.prev = tmp;  
        return c;  
    }  
}
```