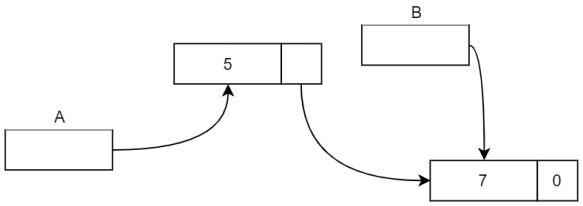
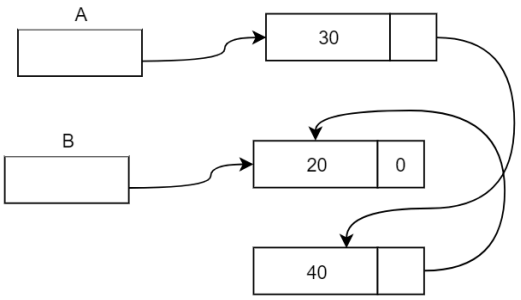
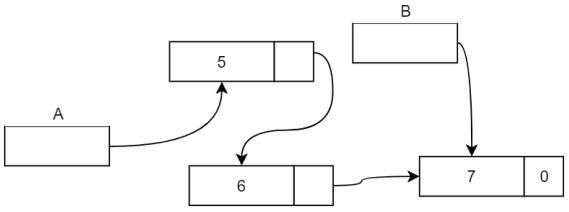
LinkedList WS

SAMPLE	Give two ways to access the value 7: **B->element** **A->next->element** Write lines to make the B node point to the A node and the A node point at nothing: **B->next = A;** **A->next = nullptr;**	
1.	List all the ways to access 40: List all the ways to access 30: List all the ways to access 20: Set the B node to point to A: Set the node with 40 in it to point to nothing:	

For each problem below, draw a picture representing the nodes and connections defined by the given code:

SAMPLE	`Node<int>* A = new Node<int>(5);`
	`Node<int>* B = new Node<int>(7);`
	`A->next = new Node<int>(6);`
	`A->next->next = B;`
		

2.	`Node<int>* A = new Node<int>(5);`
	`Node<int>* B = new Node<int>(7);`
	`B->next = new Node<int>(6);`
	`B->next->next = A;`	

3.	`Node<int>* A = new Node<int>(10);`
	`Node<int>* B = new Node<int>(11);`
	`B->next = A;`
	`A = new Node<int>(12);`
	`B->next->next = A;`	

4.	`Node<int>* A = new Node<int>(20);`
	`Node<int>* B = new Node<int>(30);`
	`Node<int>* C = new Node<int>(40);`
	`C->next = A;`
	`B->next = C;`	

5.	`Node<int>* A = new Node<int>(20);`
	`Node<int>* B = new Node<int>(30);`
	`A->next = B;`
	`B = new Node<int>(40);`
	`B->next = A;`
	`A = new Node<int>(50);`
	`A->next = B;`	

6.	`Node<int>* A = new Node<int>(20);`
	`Node<int>* B = new Node<int>(30);`
	`Node<int>* C = new Node<int>(40);`
	`A->next = B;`
	`B->next = C;`
	`Node<int>* x = A->next;`
	`x->element = 10;`
	`x->next = nullptr;`	

7.	`Node<int>* A = new Node<int>(20);`
	`Node<int>* B = new Node<int>(30);`
	`Node<int>* C = new Node<int>(40);`
	`A->next = B;`
	`B->next = C;`
	`A->next->next->next = A;`	

8.	`Node<int>* A = new Node<int>(20);`
	`Node<int>* B = new Node<int>(30);`
	`Node<int>* C = new Node<int>(40);`
	`A->next = B;`
	`B->next = C;`
	`A->next->next->next = B;`
	`A->next->next = A;`
	`A->next = nullptr;`	