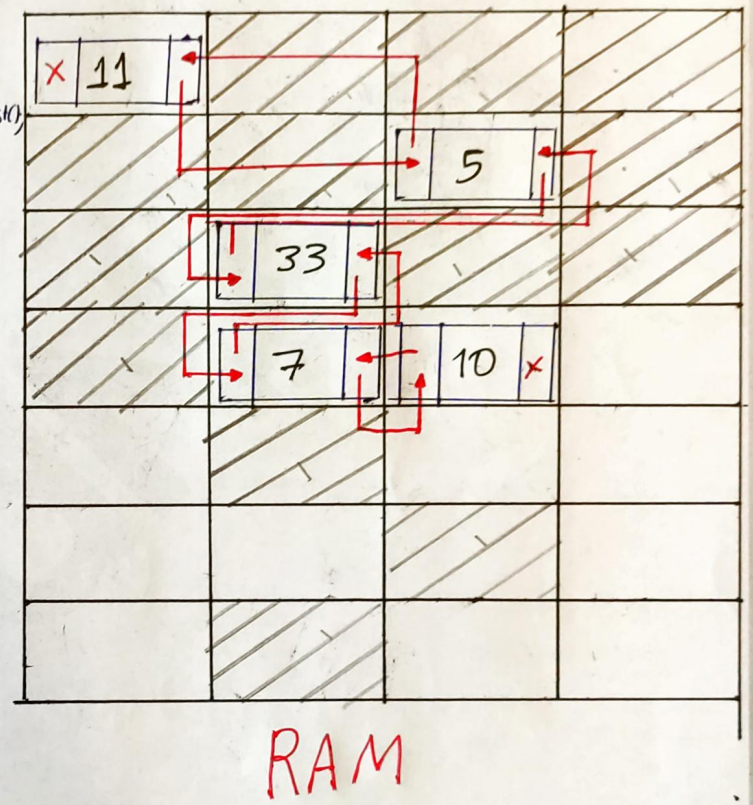
LinkedList

LinkedList is able to utilise the scattered memory present on RAM.

Example

```
LinkedList ll = new LinkedList();
ll.add(11);
ll.add(5);
ll.add(33);
ll.add(7);
ll.add(10);
```



Program On LinkedList

```
import java.util.LinkedList;

public class CollectionsDemo {
    public static void main (String[] args) {
        LinkedList ll = new LinkedList();
        ll.add(11);
        ll.add(5);
        ll.add(33);
        ll.add(7);
        ll.add(10);

        System.out.println(ll);
    }
}
```

Output

```
[11, 5, 33, 7, 10]
```

Some Inbuilt Methods Present In The LinkedList Class

contains() :- It is used to check whether a certain element present in LinkedList or not.

peek() :- It is used to return the first element.

poll() :- It is used to return and remove the element at the front end of the container.

```
import java.util.LinkedList;
public class CollectionsDemo {
    public static void main(String[] args) {
        LinkedList ll = new LinkedList();
        ll.add(11);
        ll.add("messi");
        ll.add(3.141);
        ll.add(99);
        ll.add(true);
        System.out.println(ll);

        System.out.println(ll.contains(3.141));
        System.out.println(ll.contains("ronaldo"));

        System.out.println("peek:" + ll.peek());
        System.out.println(ll);

        System.out.println("pool: " + ll.pool());
        System.out.println(ll);

    }
}
```

Output

```
[11, messi , 3.149 , 99, true ]
true
false
peek : 11
[11, messi, 3.149, 99, true ]
pool : 11
[messi, 3.149 , 99, true]
```