```
public with sharing class PaginationController {

    // Returns one page of contacts
    @AuraEnabled(cacheable=true)
    public static List<Contact> getContacts(Integer pageNumber, Integer pageSize) {
        Integer offsetSize = (pageNumber - 1) * pageSize;

        return [
            SELECT Id, FirstName, LastName, Email, Phone
            FROM Contact
            ORDER BY LastName
            LIMIT :pageSize
            OFFSET :offsetSize
        ];
    }

    // Returns total number of records (for page links)
    @AuraEnabled(cacheable=true)
    public static Integer getTotalContacts(){
        return [SELECT COUNT() FROM Contact];
    }
}
```

This Apex class is a server-side controller for an Aura component that provides pagination functionality.

- `public with sharing class PaginationController`
  Declares a publicly accessible Apex class which respects the user's sharing settings.

Method 1: getContacts(Integer pageNumber, Integer pageSize)

This method retrieves a page of Contact records.
• pageNumber – current page index requested by the UI (1-based).
• pageSize – the number of records to return per page.

It calculates how many records to skip before selecting the next set using:
    Integer offsetSize = (pageNumber - 1) * pageSize;

Then it runs a SOQL query:
• SELECT Id, FirstName, LastName, Email, Phone – fields returned
• ORDER BY LastName – sorts records alphabetically
• LIMIT :pageSize – only returns the number of records requested
• OFFSET :offsetSize – skips records from previous pages

→ This returns only the relevant Contact records for the current page.

Method 2: getTotalContacts()

This method returns the total number of Contacts in the org using:
    SELECT COUNT() FROM Contact

The returned number is used in the UI to calculate how many pages to display (for page buttons or links).

In summary:
• getContacts(...) returns a single page of records
• getTotalContacts() returns the full count of records
Together they allow a component to implement proper pagination.