Java

```java
/**
 * The read4 API is defined in the parent class Reader4.
 *     int read4(char[] buf4);
 */

public class Solution extends Reader4 {
    /**
     * @param buf Destination buffer
     * @param n   Number of characters to read
     * @return    The number of actual characters read
     */
    public int read(char[] buf, int n) {

    }
}
```

JavaScript

```javascript
/**
 * Definition for read4()
 *
 * @param {character[]} buf Destination buffer
 * @return {number} The number of characters read
 * read4 = function(buf4) {
 *     ...
 * };
 */

/**
```

```
 * @param {function} read4()
 * @return {function}
 */
var solution = function(read4) {
    /**
     * @param {character[]} buf Destination buffer
     * @param {number} n Number of characters to read
     * @return {number} The number of actual characters read
     */
    return function(buf, n) {

    };
};
```

TypeScript

```
/**
 * Definition for read4()
 * read4 = function(buf4: string[]): number {
 *     ...
 * };
 */

var solution = function(read4: any) {

    return function(buf: string[], n: number): number {

    };
};
```

C++

```cpp
/**
 * The read4 API is defined in the parent class Reader4.
 *     int read4(char *buf4);
 */

class Solution {
public:
    /**
     * @param buf Destination buffer
     * @param n   Number of characters to read
     * @return    The number of actual characters read
     */
    int read(char *buf, int n) {

    }
};
```

C#

```csharp
/**
 * The Read4 API is defined in the parent class Reader4.
 *     int Read4(char[] buf4);
 */

public class Solution : Reader4 {
    /**
     * @param buf Destination buffer
     * @param n   Number of characters to read
```

```
     * @return    The number of actual characters read
     */
    public int Read(char[] buf, int n) {

    }
}
```

Kotlin

```
/**
* The read4 API is defined in the parent class Reader4.
* fun read4(buf4:CharArray): Int {}
*/

class Solution:Reader4() {
    /**
      * @param  buf Destination buffer
      * @param  n   Number of characters to read
      * @return     The number of actual characters read
      */
    override fun read(buf:CharArray, n:Int): Int {

    }
}
```

Go

```
/**
 * The read4 API is already defined for you.
 *
```

```
 *     read4 := func(buf4 []byte) int
 *
 * // Below is an example of how the read4 API can be called.
 * file := File("abcdefghijk") // File is "abcdefghijk", initially file pointer (fp) points to 'a'
 * buf4 := make([]byte, 4) // Create buffer with enough space to store characters
 * read4(buf4) // read4 returns 4. Now buf = ['a','b','c','d'], fp points to 'e'
 * read4(buf4) // read4 returns 4. Now buf = ['e','f','g','h'], fp points to 'i'
 * read4(buf4) // read4 returns 3. Now buf = ['i','j','k',...], fp points to end of file
 */

var solution = func(read4 func([]byte) int) func([]byte, int) int {
    // implement read below.
    return func(buf []byte, n int) int {

    }
}
----------------------------------------------------------------------
```