Read N Characters Given Read4

The API: `int read4(char *buf)` reads 4 characters at a time from a file.

The return value is the actual number of characters read. For example, it returns 3 if there is only 3 characters left in the file.

By using the `read4` API, implement the function `int read(char *buf, int n)` that reads *n* characters from the file.

Note:
The `read` function will only be called once for each test case.

Solution 1

This question is very unclear to me. what are we supposed to accomplish in this problem. Our read function returns an int but the expected result looks like its a String. if we are forced to read 4 chars at a time how do we ever read n chars that are not a factor of 4 if we actually use the read4 method? is 'n' the max we can return or must we return exactly n assuming at least n items exist other wise return the number of items. clearly I'm missing a lot here can someone please explain this problem to me.

written by mlblount45 original link here

Solution 2

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

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

        char[] buffer = new char[4];
        boolean endOfFile = false;
        int readBytes = 0;

        while (readBytes < n && !endOfFile) {
            int currReadBytes = read4(buffer);
            if (currReadBytes !=4) {
                endOfFile = true;
            }
            int length = Math.min(n - readBytes, currReadBytes);
            for (int i=0; i<length; i++) {
                buf[readBytes + i] = buffer[i];
            }
            readBytes += length;
        }
        return readBytes;
    }
}
```

personally, I feel this problem is hard to understand. I would prefer the return result is the buf instead of an int. copy the buf is error prone as it is a fixed array size. lots of assumption.

written by richilee original link here

Solution 3

```
public int read(char[] buf, int n) {
  boolean eof = false;      // end of file flag
  int total = 0;            // total bytes have read
  char[] tmp = new char[4]; // temp buffer

  while (!eof && total < n) {
    int count = read4(tmp);

    // check if it's the end of the file
    eof = count < 4;

    // get the actual count
    count = Math.min(count, n - total);

    // copy from temp buffer to buf
    for (int i = 0; i < count; i++)
      buf[total++] = tmp[i];
  }

  return total;
}
```

written by jeantimex original link here

From Leetcoder.