```
/* The following code example is taken from the book
 * "The C++ Standard Library - A Tutorial and Reference"
 * by Nicolai M. Josuttis, Addison-Wesley, 1999
 *
 * (C) Copyright Nicolai M. Josuttis 1999.
 * Permission to copy, use, modify, sell and distribute this software
 * is granted provided this copyright notice appears in all copies.
 * This software is provided "as is" without express or implied
 * warranty, and with no claim as to its suitability for any purpose.
 */
#include <cstdio>
#include <streambuf>

// for write():
#ifdef _MSC_VER
# include <io.h>
#else
# include <unistd.h>
#endif

class outbuf : public std::streambuf {
  protected:
    static const int bufferSize = 10;   // size of data buffer
    char buffer[bufferSize];            // data buffer

  public:
    /* constructor
     * - initialize data buffer
     * - one character less to let the bufferSizeth character
     *    cause a call of overflow()
     */
    outbuf() {
        setp (buffer, buffer+(bufferSize-1));
    }

    /* destructor
     * - flush data buffer
     */
    virtual ~outbuf() {
        sync();
    }

  protected:
    // flush the characters in the buffer
    int flushBuffer () {
        int num = pptr()-pbase();
        if (write (1, buffer, num) != num) {
            return EOF;
        }
        pbump (-num);    // reset put pointer accordingly
        return num;
    }

    /* buffer full
     * - write c and all previous characters
     */
    virtual int_type overflow (int_type c) {
        if (c != EOF) {
            // insert character into the buffer
            *pptr() = c;
            pbump(1);
        }
        // flush the buffer
        if (flushBuffer() == EOF) {
            // ERROR
            return EOF;
        }
        return c;
    }

    /* synchronize data with file/destination
     * - flush the data in the buffer
     */
    virtual int sync () {
        if (flushBuffer() == EOF) {
            // ERROR
            return -1;
        }
        return 0;
    }
};
```