```c
1   #include <sys/wait.h>
2   #include <unistd.h>
3   #include <string.h>
4   #include <stdlib.h>
5   #include <stdio.h>
6
7   #define BUF_SIZE 10
8
9   int main(int argc, char *argv[]) /* Over-simplified! */
10  {
11    int pfd[2]; /* Pipe file descriptors */
12    char buf[BUF_SIZE];
13    ssize_t numRead;
14
15    pipe(pfd); /* Create the pipe */
16
17    switch (fork()) {
18    case 0: /* Child - reads from pipe */
19      close(pfd[1]); /* Write end is unused */
20
21      for (;;) { /* Read data from pipe, echo on stdout */
22        if ( (numRead = read(pfd[0], buf, BUF_SIZE)) == 0)
23          break;  /* End-of-file */
24        if (write(1, buf, numRead) != numRead)
25          perror("child - partial/failed write");
26      }
27      puts(""); /* newline */
28
29      close(pfd[0]);
30
31      _exit(EXIT_SUCCESS);
32
33    default: /* Parent - writes to pipe */
34      close(pfd[0]); /* Read end is unused */
35
36      if (write(pfd[1], argv[1], strlen(argv[1])) != strlen(argv[1]))
37        perror("parent - partial/failed write");
38
39      close(pfd[1]); /* Child will see EOF */
40
41      wait(NULL); /* Wait for child to finish */
42      exit(EXIT_SUCCESS);
43    }
44  }
```