

Question 5

Correct

Mark 1.00 out of 1.00

In a similar style to the lab exercise `dup2Pipe.c`, write a program which takes input from `stdin` sorts it using `/usr/bin/sort`, then takes the top 2 lines using `/usr/bin/head`. A sample run might look like this if you named your program `quiz`:

```
~$ printf "Hello\nWorld\nA\nB\n" | ./quiz
```

A
B

Warning: Unlike the lab machines, the quiz server will kill any child process as soon as the parent process exits. Consider the order that your processes will complete and ensure the last process to exit is the parent process.

For example:

Input	Result
Hello	A
World	B
A	
B	
C	

Answer: (penalty regime: 10, 20, ... %)

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <unistd.h>
4  #include <sys/wait.h>
5
6  void errExit(const char *msg) {
7      perror(msg);
8      exit(EXIT_FAILURE);
9  }
10
11 int main() {
12     int pipe1[2], pipe2[2];
13
14     if (pipe(pipe1) == -1) {
15         errExit("pipe1");
16     }
17     if (pipe(pipe2) == -1) {
18         errExit("pipe2");
19     }
20
21     pid_t pid1 = fork();
22     if (pid1 == -1) {
23         errExit("fork1");
24     }
25
26     if (pid1 == 0) { // First child: sort process
27         close(pipe1[1]);
28         dup2(pipe1[0], STDIN_FILENO);
29         close(pipe1[0]);
30
31         close(pipe2[0]);
32         dup2(pipe2[1], STDOUT_FILENO);
33         close(pipe2[1]);
34
35         execl("/usr/bin/sort", "sort", (char *) NULL);
36         errExit("execl sort");
37     }
38
39     pid_t pid2 = fork();
40     if (pid2 == -1) {
41         errExit("fork2");
42     }
43
44     if (pid2 == 0) { // Second child: head process
45         close(pipe1[0]);
46         close(pipe1[1]);
47
48         close(pipe2[1]);
49         dup2(pipe2[0], STDIN_FILENO);
50         close(pipe2[0]);
51
52         execl("/usr/bin/head", "head", "-n", "2", (char *) NULL);

```

Check

	Input	Expected	Got	
✓	Hello World A B C	A B	A B	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.