National University
of computer and emerging sciences

Course Name:	**Operating System**	**Course Code:**	**CS2006**
Degree Program:	**BS-CS**	**Semester:**	**Spring 2023**
Exam Duration:	**90 Minutes**	**Total Marks:**	**50**
Paper Date:	**30-March-2023**	**Weight**	**20 %**
Section:	**4B**	**Page(s):**	**3**
Exam Type:	**Midterm**		

Instructions:

- You can use Linux man pages for help.
- You must ensure proper code submission following the file naming instructions (below).
    - Your submission will contain your code and output screenshots.
    - Create separate folders for each question and put them in a folder.
    - Name the folder with your nameRollNoSectionSubsection i.e AhmadL21999B1
    - Then zip the file and drop it in the submission folder.
- Use C or C++ language to write all the source code files.
- Students will receive ZERO marks if the answers are plagiarized.
- Use of ANY helping material/code, cell phones, INTERNET, and flash drive are strictly prohibited
- A4-sized sheet is allowed.

Problem 1:
Create two programs, word_count and main_program, and a Makefile to build them.

The word_count program should take a text file as a command-line argument and return the total number of words in that file. A word is defined as a sequence of non-whitespace characters separated by whitespace.

The main_program should perform the following tasks:

1. Accept two command line arguments: an input file name and an output file name.
2. Fork two child processes.
3. In the first child process:
   a. Execute the word_count program with the input file as an argument.
   b. Redirect the standard output to an anonymous pipe.
4. In the second child process:
   a. Read the word count result from the pipe.
   b. Write the word count result to the output file in the following format: "Word count: X", where X is the word count.
5. The parent process should wait for both child processes to complete.

Make sure to handle errors, such as incorrect number of arguments, file opening issues, fork failures, or pipe creation issues.

Example:

Input file (input.txt):
Hello, World!
This is a test.
Pipes are cool!

Command:
./main_program input.txt output.txt

Output file (output.txt):
Word count: 9

Problem 2:
Create four programs: file_reader, word_reverser, file_writer, and main_program. Also, create a Makefile to build these programs.

The file_reader program should read data from a named pipe (FIFO) and write it to an output file. The word_reverser program should read data from an input file, reverse the order of the words in each line, and write the result to an output file. The file_writer program should read data from an input file and write it to a named pipe (FIFO).

The main_program should perform the following tasks:

1. Accept two command-line arguments: an input file name and an output file name.
2. Create a named pipe (FIFO) for IPC.
3. Fork two child processes.
4. In the first child process:
   Execute the file_reader program with the named pipe as input and a temporary output file as an argument.
5. In the second child process:
   Execute the file_writer program with the input file as an argument and the named pipe as output.
6. In the main process (parent process):
   a. Wait for both child processes to complete.
   b. Execute the word_reverser program with the temporary output file as input and the final output file as output.
7. Remove the temporary output file.

Make sure to handle errors, such as incorrect number of arguments, file opening issues, fork failures, or pipe creation issues.

Example:

Input file (input.txt):
Hello, World!
This is a test.
Pipes are cool!

Commands:
./main_program input.txt output.txt

Output file (output.txt):

World! Hello,
test. a is This
cool! are Pipes