# TASK1:

#include <unistd.h>

#include <stdio.h>

#include <stdlib.h>

#include <sys/wait.h>

int main()

{

int pipefd[2], status;

pid\_t pid1, pid2, pid3;

if (pipe(pipefd) == -1)

{

perror("pipe");

exit(EXIT\_FAILURE);

}

pid1 = fork();

if (pid1 == -1)

{

perror("fork");

exit(EXIT\_FAILURE);

}

if (pid1 == 0)

{

close(pipefd[0]);

dup2(pipefd[1], STDOUT\_FILENO);

execlp("ls", "ls", NULL);

perror("execlp");

exit(EXIT\_FAILURE);

}

pid2 = fork();

if (pid2 == -1)

{

perror("fork");

exit(EXIT\_FAILURE);

}

if (pid2 == 0)

{

close(pipefd[1]);

dup2(pipefd[0], STDIN\_FILENO);

dup2(pipefd[1], STDOUT\_FILENO);

execlp("sort", "sort", "-r", NULL);

perror("execlp");

exit(EXIT\_FAILURE);

}

pid3 = fork();

if (pid3 == -1)

{

perror("fork");

exit(EXIT\_FAILURE);

}

if (pid3 == 0)

{

close(pipefd[1]);

dup2(pipefd[0], STDIN\_FILENO);

execlp("wc", "wc", NULL);

perror("execlp");

exit(EXIT\_FAILURE);

}

close(pipefd[0]);

close(pipefd[1]);

waitpid(pid1, &status, 0);

waitpid(pid2, &status, 0);

waitpid(pid3, &status, 0);

return 0;

}