**PROGRAM 1**

Write a program that creates a child process. Parent process writes data to pipe and child process reads the data from pipe and prints it on the screen.

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/wait.h>

#define BUFFER\_SIZE 256

int main()

{

int pipefd[2];

pid\_t pid;

char buffer[BUFFER\_SIZE];

int status;

if (pipe(pipefd) == -1)

{

perror("pipe");

exit(EXIT\_FAILURE);

}

pid = fork();

if (pid < 0)

{

perror("fork");

exit(EXIT\_FAILURE);

}

else if (pid > 0)

{

close(pipefd[0]);

printf("Parent process writing data to the pipe...\n");

char message[] = "\nHello, child process!";

write(pipefd[1], message, sizeof(message));

close(pipefd[1]);

wait(&status);

}

else

{

close(pipefd[1]);

printf("Child process reading data from the pipe...\n\n");

ssize\_t bytesRead = read(pipefd[0], buffer, sizeof(buffer));

printf("Data received from parent process: %.\*s\n", (int)bytesRead, buffer);

close(pipefd[0]);

exit(EXIT\_SUCCESS);

}

return 0;

}

**OUTPUT**

