LAB EXERCISE I : Pipe Command

Name : Ivan J Madathil

Reg. No: 23BRS1127

Q1. Parent process sends your Name to Child process and Child Process computes the number of vowels present in the received message (Name) and prints it. Implement the above IPC through named pipes using C

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <ctype.h>

int count_vowels(const char *str, char *vowel_list) {
    int count = 0;
    const char *vowels = "AEIOUaeiou";
    int pos = 0;

    while (*str) {
        if (strchr(vowels, *str)) {
            count++;
            vowel_list[pos++] = *str;
        }
        str++;
    }

    vowel_list[pos] = '\0';
    return count;
}

int is_valid_name(const char *name) {
    for (int i = 0; name[i]; i++) {
        if (!isspace(name[i])) {
            return 1;
        }
    }
    return 0;
}

int main() {
    int pipefd[2];
    char name[100];
    pid_t pid;

    if (pipe(pipefd) == -1) {
        perror("pipe");
        exit(EXIT_FAILURE);
    }

    pid = fork();

    if (pid < 0) {
        perror("fork");
        exit(EXIT_FAILURE);
```

```c
    int pipefd[2];
    char name[100];
    pid_t pid;

    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: Enter your name: ");
        if (scanf(" %99[^\n]", name) != 1 || !is_valid_name(name)) {
            printf("Error: Invalid input. Please enter a valid name.\n");
            close(pipefd[1]);
            exit(EXIT_FAILURE);
        }
        write(pipefd[1], name, strlen(name) + 1);
        close(pipefd[1]);
    } else {
        close(pipefd[1]);
        char received_name[100];
        char vowels_found[100];
        read(pipefd[0], received_name, sizeof(received_name));
        close(pipefd[0]);
        if (strlen(received_name) == 0 || !is_valid_name(received_name)) {
            printf("Error: Received invalid or empty input.\n");
            exit(EXIT_FAILURE);
        }
        int vowel_count = count_vowels(received_name, vowels_found);
        if (vowel_count > 0) {
            printf("Child: The name '%s' contains %d vowels: %s\n", received_name, vowel_count, vowels_found);
        } else {
            printf("Child: The name '%s' contains no vowels.\n", received_name);
        }
    }

    return 0;
}
```

```
[omar@jik Downloads]$ ./test
Parent: Enter your name: Ivan J Madathil
Child: The name 'Ivan J Madathil' contains 5 vowels: Iaaai
[omar@jik Downloads]$ ./test
Parent: Enter your name: kssdwff
Child: The name 'kssdwff' contains no vowels.
[omar@jik Downloads]$ ./test
Parent: Enter your name: 1211
Child: The name '1211' contains no vowels.
[omar@jik Downloads]$
```