Labsheet 7

Q1

Code :

```c
#include <stdio.h>
#include <unistd.h>
#include <ctype.h>
#include <string.h>
int main() {
    int fd[2];
    pipe(fd);
    pid_t pid = fork();
    if (pid == 0) {
          close(fd[0]); // Close read end
          char str[100];
          printf("Child: Enter a string: ");
          fgets(str, sizeof(str), stdin);
          write(fd[1], str, strlen(str) + 1);
          close(fd[1]);

    } else {
    close(fd[1]); // Close write end
    char buffer[100];
    read(fd[0], buffer, sizeof(buffer));
    for (int i = 0; buffer[i]; i++)
    buffer[i] = toupper(buffer[i]);
    printf("Parent: Uppercase String: %s\n", buffer);
    close(fd[0]);
    }
    return 0;
}
```

Output :

```
ubuntu@ubuntu:~$ gcc os_labsheet7.c
ubuntu@ubuntu:~$ ./a.out
Child: Enter a string: hello
Parent: Uppercase String: HELLO
```

Q2

Code

```c
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
int main() {
    int fd[2];
    pipe(fd);
    pid_t pid = fork();
    if (pid == 0) {
        close(fd[1]);
        int num, sum = 0;
        while (read(fd[0], &num, sizeof(int)) > 0) {
            sum += num; }
        printf("Child: Sum = %d\n", sum);
        close(fd[0]);
    } else {
        close(fd[0]);
        char input[20];
        while (1) {
            printf("Parent: Enter an integer (or # to stop): ");
            scanf("%s", input);
            if (input[0] == '#')
                break;
            int num = atoi(input);
            write(fd[1], &num, sizeof(int)); }
        close(fd[1]); }
    return 0;
}
```

Output

```
ubuntu@ubuntu:~$ ./a.out
Parent: Enter an integer (or # to stop): 1
Parent: Enter an integer (or # to stop): 2
Parent: Enter an integer (or # to stop): 3
Parent: Enter an integer (or # to stop): 4
Parent: Enter an integer (or # to stop): 5
Parent: Enter an integer (or # to stop): 6
Parent: Enter an integer (or # to stop): 7
Parent: Enter an integer (or # to stop): 8
Parent: Enter an integer (or # to stop): #
Child: Sum = 36
```

Q3

Code :

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/wait.h>
int main() {
    int sum_pipe[2], count_pipe[2], return_sum[2], return_count[2];
    pipe(sum_pipe);
    pipe(count_pipe);
    pipe(return_sum);
    pipe(return_count);
    pid_t c1 = fork();
    if (c1 == 0) {
        close(sum_pipe[1]);
        close(return_sum[0]);
        close(count_pipe[0]); close(count_pipe[1]);
        close(return_count[0]); close(return_count[1]);
        long num, sum = 0;
        while (read(sum_pipe[0], &num, sizeof(long)) > 0) {
            sum += num; }
        write(return_sum[1], &sum, sizeof(long));
        close(sum_pipe[0]);
        close(return_sum[1]);
        exit(0); }
    pid_t c2 = fork();
    if (c2 == 0) {
        close(count_pipe[1]);
        close(return_count[0]);
        close(sum_pipe[0]); close(sum_pipe[1]);
        close(return_sum[0]); close(return_sum[1]);
        int dummy, count = 0;
        while (read(count_pipe[0], &dummy, sizeof(int)) > 0) {
```

```
        int dummy, count = 0;
        while (read(count_pipe[0], &dummy, sizeof(int)) > 0) {
            count++;
        }
        write(return_count[1], &count, sizeof(int));
        close(count_pipe[0]);
        close(return_count[1]);
        exit(0);
    }
    close(sum_pipe[0]);
    close(count_pipe[0]);
    close(return_sum[1]);
    close(return_count[1]);
    char input[20];
    while (1) {
        printf("Enter integer (or # to stop): ");
        scanf("%19s", input);
        if (strcmp(input, "#") == 0)
            break;
        char *endptr;
        long n = strtol(input, &endptr, 10);
        if (*endptr != '\0') {
            printf("Invalid input, please enter a valid integer.\n");
            continue;
        }
        long square = n * n;
        write(sum_pipe[1], &square, sizeof(long));
        write(count_pipe[1], &n, sizeof(int));  // Send dummy value
    }
    close(sum_pipe[1]);
    close(count_pipe[1]);
    wait(NULL);
    wait(NULL);
    long total_sum;
    int total_count;
    read(return_sum[0], &total_sum, sizeof(long));
    read(return_count[0], &total_count, sizeof(int));
    close(return_sum[0]);
    close(return_count[0]);
    if (total_count == 0) {
        printf("No valid input provided.\n");
    } else {
        float mean = (float) total_sum / total_count;
        printf("Parent: Average of squares = %.2f\n", mean);
    }
    return 0;
}
```

Output :

```
ubuntu@ubuntu:~$ gcc os_labsheet7.c
ubuntu@ubuntu:~$ ./a.out
Enter integer (or # to stop): 1
Enter integer (or # to stop): 2
Enter integer (or # to stop): 3
Enter integer (or # to stop): 4
Enter integer (or # to stop): 5
Enter integer (or # to stop): #
Parent: Average of squares = 11.00
```