

```

#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/types.h>

int main()
{
    int fd[2];
    pid_t pid;

    if (pipe(fd) == -1) {
        fprintf(stderr, "Pipe failed");
        return 1;
    }

    pid = fork();

    if (pid < 0) {
        fprintf(stderr, "Fork failed");
        return 1;
    }

    if (pid == 0) {
        // Child process
        char name[100], roll[100];

        printf("Enter your name: ");
        scanf("%s", name);

        printf("Enter your roll number: ");
        scanf("%s", roll);

        close(fd[0]); // Close unused read end of the pipe

        write(fd[1], name, sizeof(name));
        write(fd[1], roll, sizeof(roll));

        exit(0);
    }
    else {
        // Parent process
        char name[100], roll[100];

        close(fd[1]); // Close unused write end of the pipe
    }
}

```

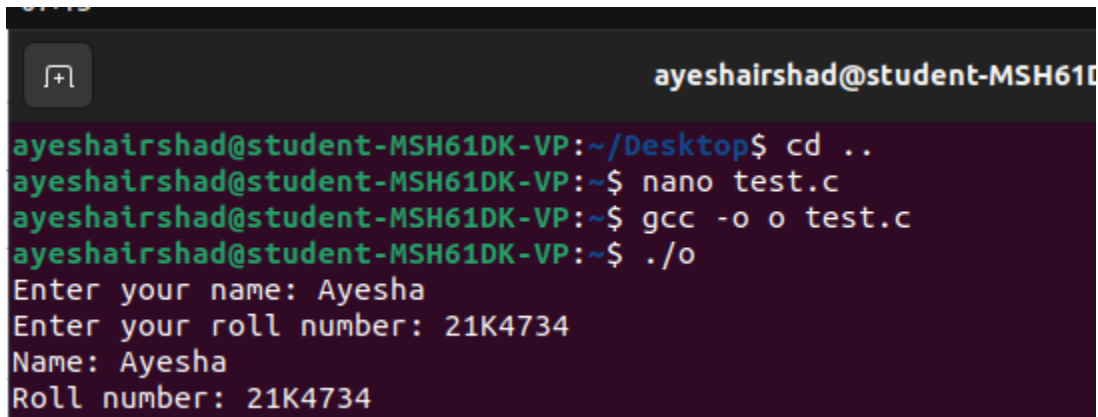
```

        read(fd[0], name, sizeof(name));
        read(fd[0], roll, sizeof(roll));

        printf("Name: %s\n", name);
        printf("Roll number: %s\n", roll);
    }

    return 0;
}

```



```

ayeshairshad@student-MSH61DK-VP:~/Desktop$ cd ..
ayeshairshad@student-MSH61DK-VP:~$ nano test.c
ayeshairshad@student-MSH61DK-VP:~$ gcc -o o test.c
ayeshairshad@student-MSH61DK-VP:~$ ./o
Enter your name: Ayesha
Enter your roll number: 21K4734
Name: Ayesha
Roll number: 21K4734

```

## Question no 2:

```

#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/types.h>

int main()
{
    int fd[2];
    pid_t pid;

    if (pipe(fd) == -1) {
        fprintf(stderr, "Pipe failed");
        return 1;
    }

    pid = fork();

    if (pid < 0) {
        fprintf(stderr, "Fork failed");
        return 1;
    }
}

```

```

}

if (pid == 0) {
    // Child process
    char name[100], roll[100];

    close(fd[0]); // Close unused read end of the pipe

    printf("Enter your name: ");
    scanf("%s", name);

    printf("Enter your roll number: ");
    scanf("%s", roll);

    write(fd[1], name, sizeof(name));
    write(fd[1], roll, sizeof(roll));

    exit(0);
}
else {
    // Parent process
    char name[100], roll[100];

    close(fd[1]); // Close unused write end of the pipe

    read(fd[0], name, sizeof(name));
    read(fd[0], roll, sizeof(roll));

    printf("Name: %s\n", name);
    printf("Roll number: %s\n", roll);
}

return 0;
}

```

**The output is same as question 1 but now child process is writing and parent is reading**

### Question no 3:

```

#include <stdio.h>
#include <stdlib.h>

```

```

#include <string.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>

#define FIFO_NAME "myfifo"

int main()
{
    char buf[100];
    int fd;

    // Create FIFO (named pipe)
    mkfifo(FIFO_NAME, 0666);

    // Write process 1
    if (fork() == 0) {
        fd = open(FIFO_NAME, O_WRONLY);
        write(fd, "Ayesha Irshad", strlen("Ayesha Irshad")+1);
        close(fd);
        exit(0);
    }

    // Write process 2
    if (fork() == 0) {
        fd = open(FIFO_NAME, O_WRONLY);
        write(fd, "Anything edible", strlen("Anything edible")+1);
        close(fd);
        exit(0);
    }

    // Read process 1
    if (fork() == 0) {
        fd = open(FIFO_NAME, O_RDONLY);
        read(fd, buf, sizeof(buf));
        printf("%s\n", buf);
        close(fd);
        exit(0);
    }

    // Read process 2
    if (fork() == 0) {
        fd = open(FIFO_NAME, O_RDONLY);

```

```

        read(fd, buf, sizeof(buf));
        printf("%s\n", buf);
        close(fd);
        exit(0);
    }

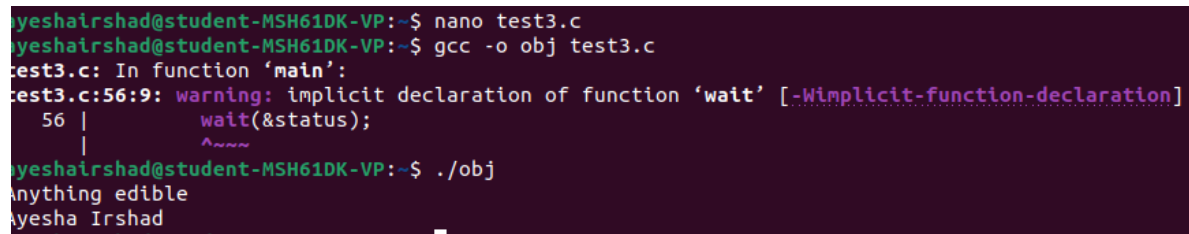
    // Wait for all child processes to finish
    int status;
    for (int i = 0; i < 4; i++) {
        wait(&status);
    }

    // Remove FIFO (named pipe)
    unlink(FIFO_NAME);

    return 0;
}

```

## Output:



```

ayeshairshad@student-MSH61DK-VP:~$ nano test3.c
ayeshairshad@student-MSH61DK-VP:~$ gcc -o obj test3.c
test3.c: In function 'main':
test3.c:56:9: warning: implicit declaration of function 'wait' [-Wimplicit-function-declaration]
   56 |         wait(&status);
      |         ^~~~
ayeshairshad@student-MSH61DK-VP:~$ ./obj
Anything edible
ayesha Irshad

```