

1. Készítsen C nyelvű programot, ahol egy szülő processz létrehoz egy csővezetékét, a gyerek processz beleír egy szöveget a csővezetékbe (A kiírt szöveg: XY neptunkod), a szülő processz ezt kiolvassa, és kiírja a standard kimenetre.

Mentés: neptunkod_unnamed.c

G2SKZ4_unnamed.c:

```
#include <stdio.h>
#include <unistd.h>

int main() {
    int fd[2];
    int child;

    if (pipe(fd)) {
        perror("error");
        return 1;
    }

    child = fork();

    if (child > 0) {
        char s[1024];
        close(fd[1]);
        read(fd[0], s, sizeof(s));
        printf("%s", s);
    }
    else if (child == 0) {
        close(fd[0]);
        write(fd[1], "AD G2SKZ4\n", 10);
        close(fd[1]);
    }
}
```

2. Készítsen C nyelvű programot, ahol egy szülő processz létrehoz egy nevesített csővezetékét (neve: neptunkod), a gyerek processz beleír egy szöveget a csővezetékbe (A hallgató neve: pl. Keserű Ottó), a szülő processz ezt kiolvassa, és kiírja a standard kimenetre.

Mentés: neptunkod_named.c

G2SKZ4_named.c:

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

int main() {
    int child;
    char * myfifo = "g2skz4";

    mkfifo(myfifo, S_IRUSR | S_IWUSR);

    child = fork();

    if (child > 0) {
        char s[1024];
        int fd = open(myfifo, O_RDONLY);
        read(fd, s, sizeof(s));
        close(fd);
        printf("%s", s);
        unlink(myfifo);
    }
    else if (child == 0) {
        int fd = open(myfifo, O_WRONLY);
        write(fd, "Arvai Dora\n", 11);
        close(fd);
    }

    return 0;
}
```