1. Készítsen C nyelvű programot, ahol egy szülő processz létrehoz egy csővezetéket, 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]);
    }
}
```

Futtatás eredménye:

```
[arvaid@x550 09_pipe]$ ./unnamed
AD G2SKZ4
```

2. Készítsen C nyelvű programot, ahol egy szülő processz létrehoz egy nevesített csővezetéket (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;
}
```

Futtatás eredménye:

```
[arvaid@x550 09_pipe]$ ./unnamed
AD G2SKZ4
```

3. Írjon C nyelvű programot, amelyik kill() seg.-vel SIGALRM-et küld egy argumentumként megadott PID-u processznek, egy másik futó program a SIGALRM-hez rendeljen egy fv.-t amely kiírja pl.neptunkodot, továbbá pause() fv.-el blokkolódjon, majd kibillenés után jelezze, hogy kibillent és terminálódjon.

Mentés. neptunkod_gyak9_3.c

G2SKZ4_gyak9_3.c_1:

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <ctype.h>
#include <signal.h>

int main(int argc, char** argv)
{
    int pid;

    // van-e argumentum?
    if (argc < 2) {
        perror("hibas parameter");
        return 1;
    }

    // szám-e az argumentum
    for (int i = 0; argv[1][i] != '\0'; i++) {
        if (!isdigit(argv[1][i])) {
            perror("hibas pid");
            return 1;
        }
    }

    pid = atoi(argv[1]); // char* to int
    kill(pid, SIGALRM);  // signal küldés
    return 0;
}
```

G2SKZ4_gyak9_3.c_2:

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

void signal_kezel();

int main() {
    int pid = getpid(); // saját pid lekérése
    printf("pid: %d\n", pid);  // pid kiíratása
    printf("varakozas...\n");
    signal(SIGALRM, signal_kezel); // signálra várakozás
    pause();
    printf("kesz\n");
}
```

```
void signal_kezel() {
    printf("G2SKZ4\n");
}
```

Futtatás eredménye: