1.Feladat.

otvos1@DESKTOP-E0G2FRE: ~

```
 GNU nano 2.9.3

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

#define MSIZE 128

int main()
{
        char inBuffer[MSIZE];
        int p[2], nbytes, pid;

        if (pipe(p) <0)
        {
                perror("Hiba tortent\n");
                exit(1);
        }

        pid=fork();

        if(pid < 0)
        {
                exit(2);
        }

        if(pid == 0)
        {
                printf("Child irja a pipeba\n");
                write(p[1], "OA BKDZXB", MSIZE);
                printf("Child pipeba irt, lehet olvasni\n");
        }

        else if(pid>0)
        {
                wait(NULL);
                printf("Parent kiolvas a pipebol\n");
                read(p[0], inBuffer, MSIZE);
                printf("Parent kiolvasta ezt: %s\n", inBuffer);
        }

        return 0;
}
```

otvos1@DESKTOP-E0G2FRE: ~

```
otvos1@DESKTOP-E0G2FRE:~$ nano BKDZXB_named.c
otvos1@DESKTOP-E0G2FRE:~$ gcc BKDZXB_named.c -o named
otvos1@DESKTOP-E0G2FRE:~$ ./unnamed
Child irja a pipeba
Child pipeba irt, lehet olvasni
Parent kiolvas a pipebol
Parent kiolvasta ezt: OA BKDZXB
otvos1@DESKTOP-E0G2FRE:~$
```

2.Feladat

otvos1@DESKTOP-E0G2FRE: ~

GNU nano 2.9.3

```c
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <string.h>

#define MSIZE 128

int main()
{
        char inBuffer[MSIZE];
        int pid, fileDesc;
        char input[] = "Otvos Adam Balazs";
        char * fifo = "/tmp/BKDZXB";

        mkfifo(fifo, 0666);

        pid = fork();

        if(pid < 0)
        {
                exit(2);
        }

        if(pid == 0)
        {
                printf("child beir a fifo pipeba\n");

                fileDesc = open(fifo, O_WRONLY);
                write(fileDesc, input, strlen(input)+1);
                printf("child irt a pipeba\n");
        }

        else if(pid > 0)
        {
                fileDesc = open(fifo, O_RDONLY);
                read(fileDesc, inBuffer, strlen(input)+1);
                printf("Parent ezt kapta %s\n", inBuffer);
                close(fileDesc);
        }
        return 0;
```

otvos1@DESKTOP-E0G2FRE: ~

```
otvos1@DESKTOP-E0G2FRE:~$ ./named
child beir a fifo pipeba
child irt a pipeba
Parent ezt kapta Otvos Adam Balazs
otvos1@DESKTOP-E0G2FRE:~$
```

3.Feladat

otvos1@DESKTOP-E0G2FRE: ~

```
 GNU nano 2.9.3

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

int main(int argc, char **argv)
{
        if(argc != 2)
        {
                printf("Hasznal: ./BKDZXB_gyak9_3_1 PID\n");
                return 1;
        }

        pid_t pid = (pid_t)atoi(argv[1]);
        kill(pid, SIGALRM);
        return 0;
}
```

otvos1@DESKTOP-E0G2FRE: ~

```
Hasznal: ./BKDZXB_gyak9_3_1 PID
otvos1@DESKTOP-E0G2FRE:~$ nano BKDZXB_gyak9_3_1.c
otvos1@DESKTOP-E0G2FRE:~$ ./gyak91
Hasznal: ./BKDZXB_gyak9_3_1 PID
otvos1@DESKTOP-E0G2FRE:~$
```

otvos1@DESKTOP-E0G2FRE: ~

```
 GNU nano 2.9.3

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

void AlarmHandler(int sig);

int main(void)
{
        if(signal(SIGALRM, AlarmHandler) == SIG_ERR)
        {
                printf("hiba tortent\n");
                return 1;
        }

        pause();
        return 0;
}

void AlarmHandler(int sig)
{
        printf("BKDZXB\nMost mar nem blokkolodok\n");
        exit(1);
}
```

4.Feladat

otvos1@DESKTOP-E0G2FRE: ~

GNU nano 2.9.3

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

void TerminationHandler(int sig);

int main(void)
{
        if(signal(SIGTERM, TerminationHandler) == SIG_ERR)
        {
                printf("Nem sikerult Handlert allitani \"SIGTERM\" jelre\n");
                return 0;
        }

        while(1)
        {
                printf("Varakozas\n");
                sleep(3);
        }
        return 0;
}

void TerminationHandler(int sig)
{
        signal(sig, SIG_IGN);
        printf("SIGTERM signal: %d\n", sig);
}
```