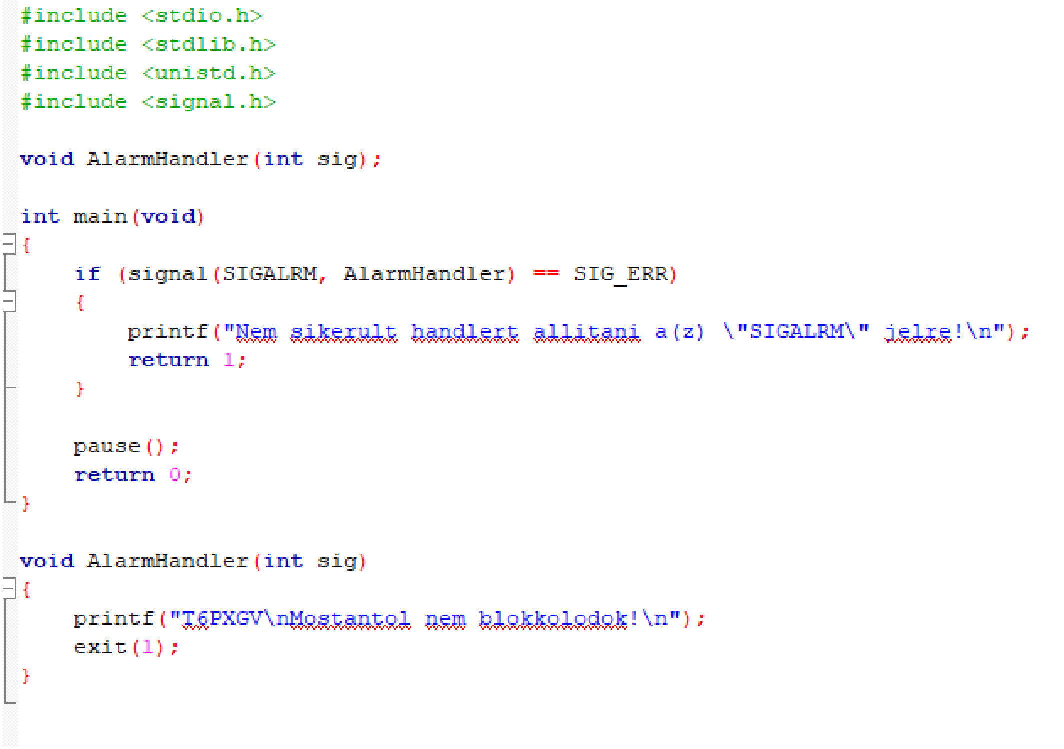


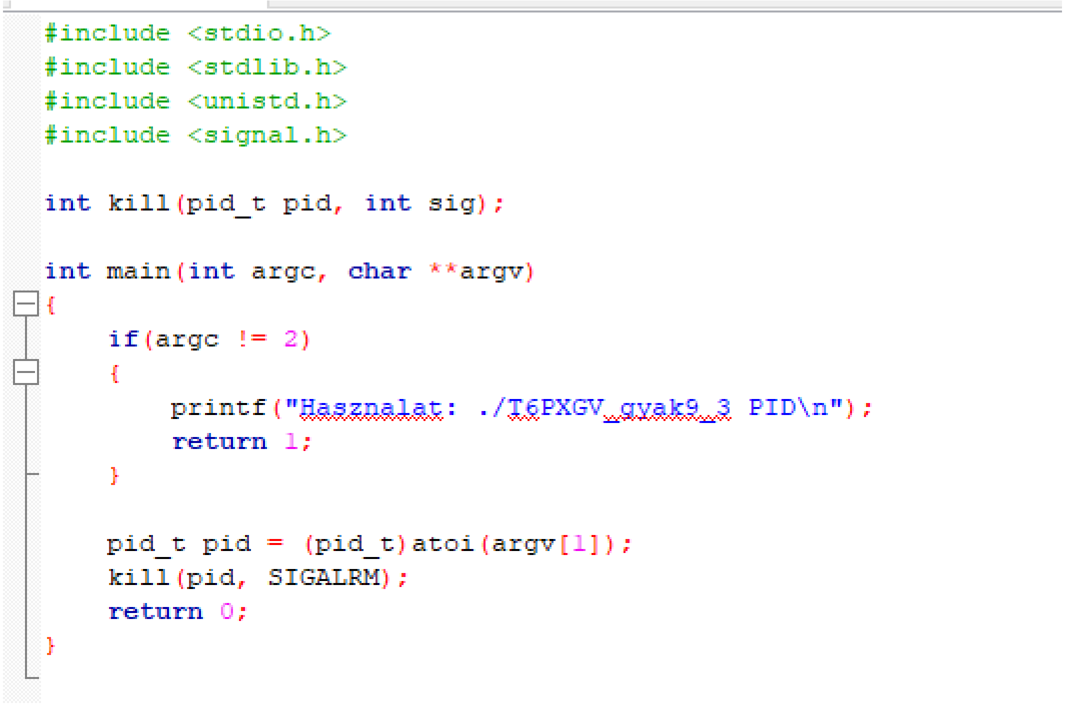
```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <signal.h>

void AlarmHandler(int sig);

int main(void)
{
    if (signal(SIGALRM, AlarmHandler) == SIG_ERR)
    {
        printf("Nem sikerult handlert allitani a(z) \"SIGALRM\" jelre!\n");
        return 1;
    }

    pause();
    return 0;
}

void AlarmHandler(int sig)
{
    printf("T6PXGV\nMostantol nem blokkolodok!\n");
    exit(1);
}
```

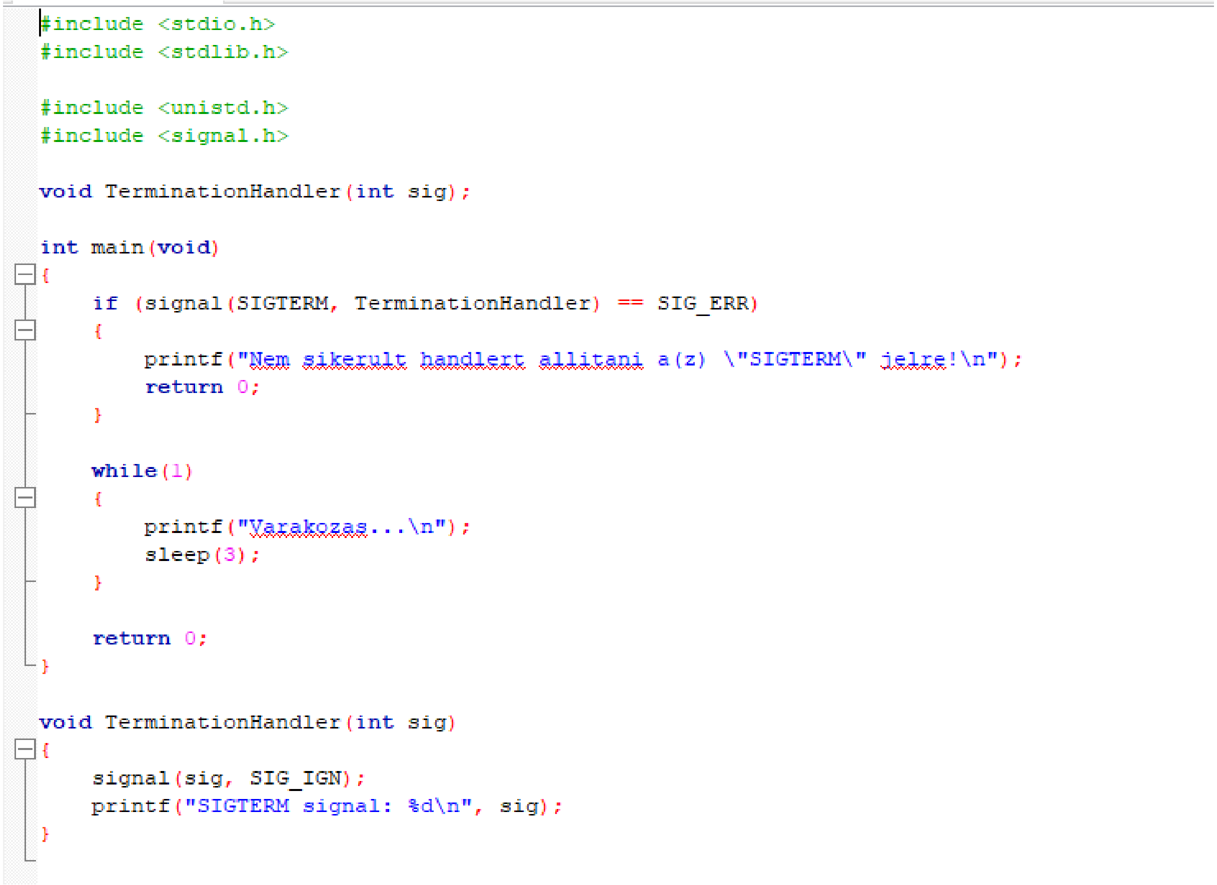


```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <signal.h>

int kill(pid_t pid, int sig);

int main(int argc, char **argv)
{
    if(argc != 2)
    {
        printf("Hasznalat: ./T6PXGV_gyak9_3 PID\n");
        return 1;
    }

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



```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 a(z) \"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);
}
```