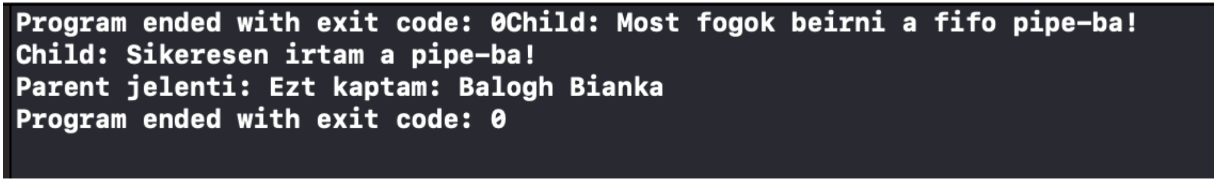
1, **unamed.c**

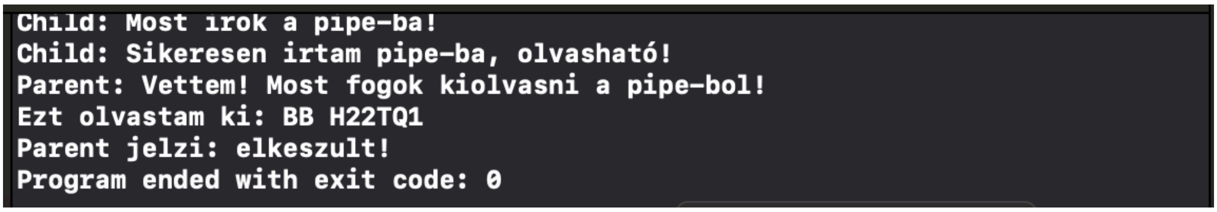
```
//
//  main.c
//  H22TQ1_unamed
//
//  Created by Balogh Bianka on 2021. 04. 19..
//

#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[] = "Balogh Bianka";
char * fifo = "/tmp/H22TQ1";
mkfifo(fifo, 0666);
pid = fork();
if (pid < 0)
   exit(2);
//Child Beleír
if (pid == 0){
   printf("Child: Most fogok beirni a fifo pipe-ba!\n");
   fileDesc = open(fifo, O_WRONLY);
   write(fileDesc, input, strlen(input)+1);
   printf("Child: Sikeresen irtam a pipe-ba!\n");
}
//Parent Kiolvassa
else if (pid > 0 ){
   fileDesc = open(fifo, O_RDONLY);
   read(fileDesc, inBuffer, strlen(input)+1);
   printf("Parent jelenti: Ezt kaptam: %s\n", inBuffer);
   close(fileDesc);
}

return 0;
}
```



```
Program ended with exit code: 0Child: Most fogok beirni a fifo pipe-ba!
Child: Sikeresen irtam a pipe-ba!
Parent jelenti: Ezt kaptam: Balogh Bianka
Program ended with exit code: 0
```

2, **named.c**

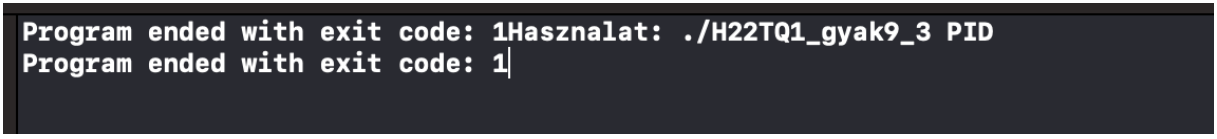
```
//
//  main.c
//  H22TQ1_named
//
//  Created by Balogh Bianka on 2021. 04. 19..
//

#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/wait.h>
#define MSIZE 10
int main(){
char inBuffer[MSIZE];
int p[2], nbytes, pid;
if (pipe(p) < 0){
   perror("Pipe hiba");
   exit(1);
}
pid = fork();
if (pid < 0)
   exit(2);
//Child beleír
if (pid == 0){
   printf("Child: Most irok a pipe-ba!\n");
   write(p[1], "BB H22TQ1", MSIZE);
   printf("Child: Sikeresen irtam pipe-ba, olvasható!\n");
}
//Parent Kiolvassa
else if (pid > 0 ){
   wait(NULL);
   printf("Parent: Vettem! Most fogok kiolvasni a pipe-bol!\n");
   read(p[0], inBuffer, MSIZE);
   printf("Ezt olvastam ki: %s\n", inBuffer);
   printf("Parent jelzi: elkeszult!\n");
}

return 0;
}
```



```
Child: Most irok a pipe-ba!
Child: Sikeresen irtam pipe-ba, olvasható!
Parent: Vettem! Most fogok kiolvasni a pipe-bol!
Ezt olvastam ki: BB H22TQ1
Parent jelzi: elkeszult!
Program ended with exit code: 0
```

3, **H22TQ1_gyak9_3.c**

```
//
//  main.c
//  H22TQ1_gyak9_3
//
//  Created by Balogh Bianka on 2021. 04. 19..
//

#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: ./H22TQ1_gyak9_3 PID\n");
        return 1;
    }

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



```
Program ended with exit code: 1Hasznalat: ./H22TQ1_gyak9_3 PID
Program ended with exit code: 1
```

H22TQ1_gyak_9_3_2

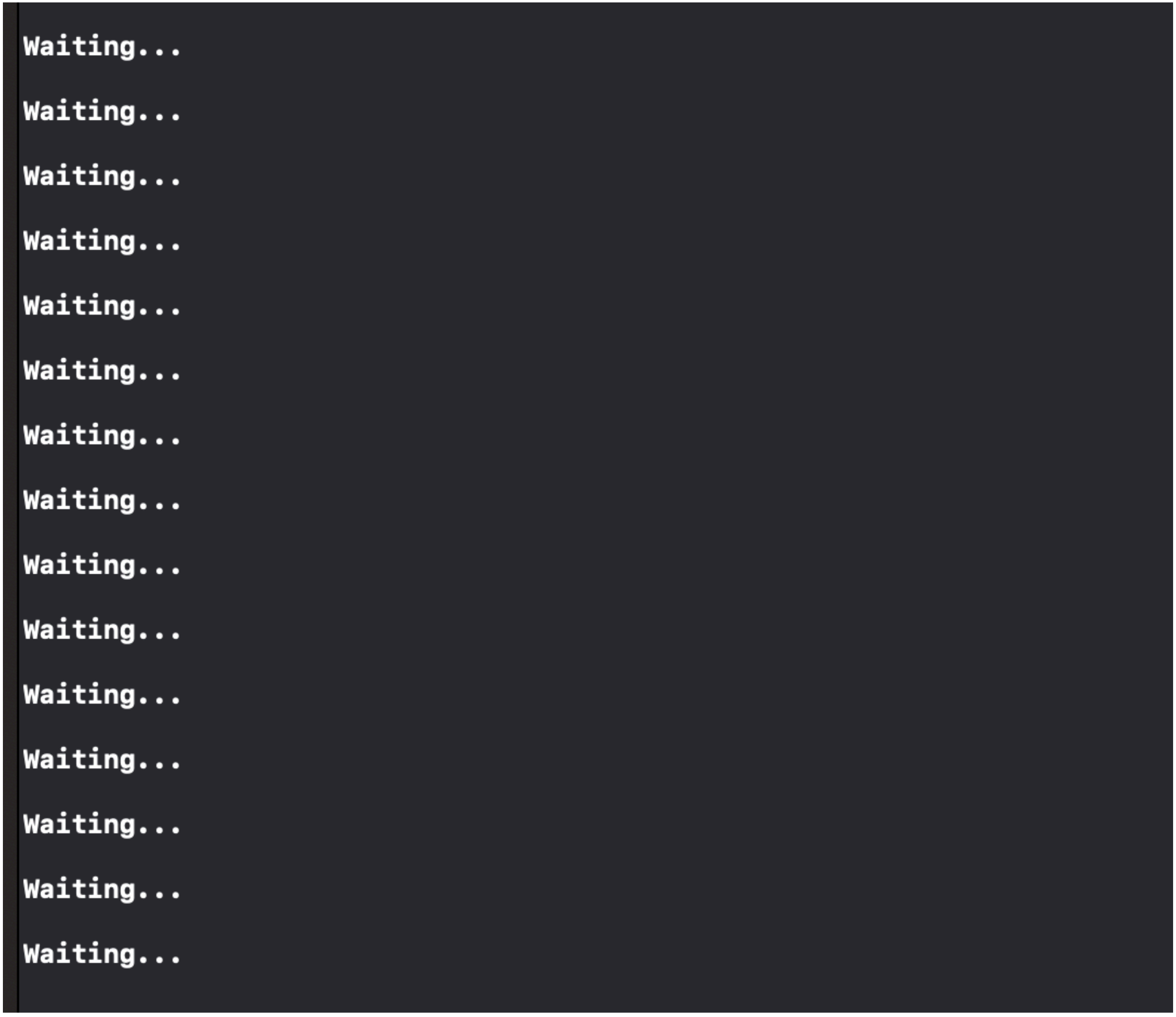
```
//
//  main.c
//  H22TQ1_gyak_9_3_2
//
//  Created by Balogh Bianka on 2021. 04. 29..
//

#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("H22TQ1\nMostantol nem blokkolodok!\n");
   exit(1);
}
```

4, **H22TQ1_gyak_9_4**

```
//
//  main.c
//  H22TQ1_gyak_9_4
//
//  Created by Balogh Bianka on 2021. 04. 19..
//

#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)
    {
        sleep(3);
    printf("\nWaiting...\n");
    }

    return 0;
}

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



```
Waiting...

Waiting...

Waiting...

Waiting...

Waiting...

Waiting...

Waiting...

Waiting...

Waiting...

Waiting...

Waiting...

Waiting...

Waiting...

Waiting...

Waiting...
```