

**Name : Anupam Kunwar**  
**Reg : 19BCE1369**

**Q1.** Set up a interrupt handler for ctrl+c which ends the current process and also displays the current process ID.

**Solution :**

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

void sigproc()
{
printf("you have pressed ctrl-c \n");
printf("Process ID : %d\n",getpid());
exit(0);
}

void quitproc()
{
printf("ctrl-c pressed to quit\n");
exit(0);
}

int main()
{
signal(SIGINT, sigproc);
signal(SIGQUIT, quitproc);
printf("use ctrl-c to quit\n");
for(;;);
}
```

**Ouput :**

```
piratepanda@SastaPC: ~/Documents/oslab/week4
File Edit View Search Terminal Help
piratepanda@SastaPC:~/Documents/oslab/week4$ gedit ctrlC.c
^C
piratepanda@SastaPC:~/Documents/oslab/week4$ gcc ctrlC.c -o ctrlC.out
piratepanda@SastaPC:~/Documents/oslab/week4$ ./ctrlC.out
use ctrl-c to quit
^Cyou have pressed ctrl-c
Process ID : 9404
piratepanda@SastaPC:~/Documents/oslab/week4$ |
```

.

**Q2.** Set up handler for at least 4 signals.

**Solution :**

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

void handler(int num){
printf(" CTRL-C will not work!!\n");
}

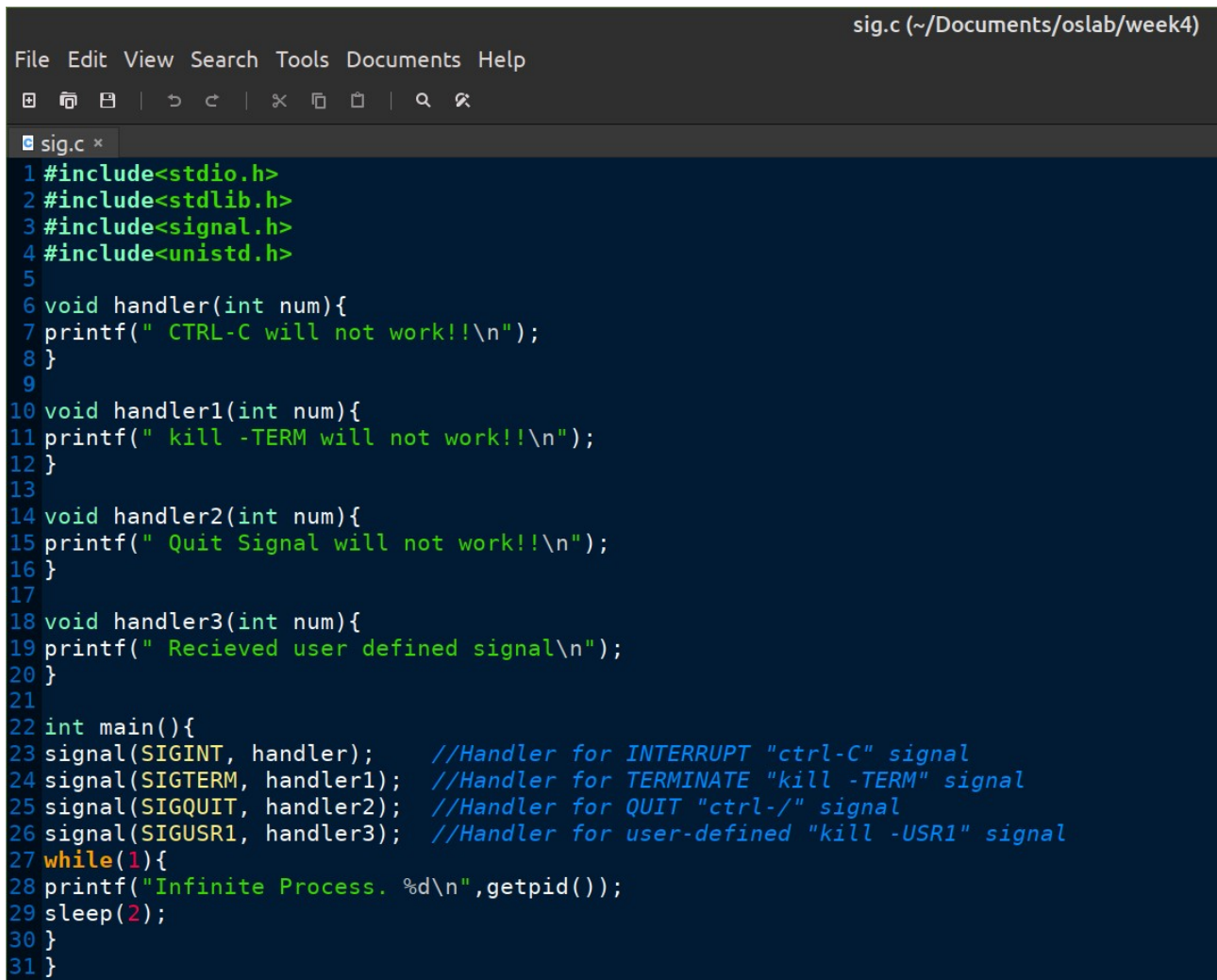
void handler1(int num){
printf(" kill -TERM will not work!!\n");
}

void handler2(int num){
printf(" Quit Signal will not work!!\n");
}

void handler3(int num){
printf(" Recieved user defined signal\n");
}

int main(){
signal(SIGINT, handler); //Handler for INTERRUPT "ctrl-C" signal
signal(SIGTERM, handler1); //Handler for TERMINATE "kill -
TERM" signal
signal(SIGQUIT, handler2); //Handler for QUIT "ctrl-/" signal
```

```
signal(SIGUSR1, handler3);    //Handler for user-defined "kill -USR1"
signal
while(1){
printf("Infinite Process. %d\n",getpid());
sleep(2);
}
}
```



```
sig.c (~/Documents/oslab/week4)
File Edit View Search Tools Documents Help
sig.c x
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<signal.h>
4 #include<unistd.h>
5
6 void handler(int num){
7 printf(" CTRL-C will not work!!\n");
8 }
9
10 void handler1(int num){
11 printf(" kill -TERM will not work!!\n");
12 }
13
14 void handler2(int num){
15 printf(" Quit Signal will not work!!\n");
16 }
17
18 void handler3(int num){
19 printf(" Recieved user defined signal\n");
20 }
21
22 int main(){
23 signal(SIGINT, handler);    //Handler for INTERRUPT "ctrl-C" signal
24 signal(SIGTERM, handler1);  //Handler for TERMINATE "kill -TERM" signal
25 signal(SIGQUIT, handler2);  //Handler for QUIT "ctrl-/" signal
26 signal(SIGUSR1, handler3);  //Handler for user-defined "kill -USR1" signal
27 while(1){
28 printf("Infinite Process. %d\n",getpid());
29 sleep(2);
30 }
31 }
```

## Output :

```
piratepanda@SastaPC: ~/Documents/oslab/week4
File Edit View Search Terminal Help
piratepanda@SastaPC:~/Documents/oslab/week4$ gcc sig.c -o sig.out
piratepanda@SastaPC:~/Documents/oslab/week4$ ./sig.out
Infinite Process. 10790
^C CTRL-C will not work!!
Infinite Process. 10790
Infinite Process. 10790
Infinite Process. 10790
Infinite Process. 10790
Infinite Process. 10790
kill -TERM will not work!!
Infinite Process. 10790
Infinite Process. 10790
^\ Quit Signal will not work!!
Infinite Process. 10790
Infinite Process. 10790
Infinite Process. 10790
Infinite Process. 10790
Infinite Process. 10790
Recieved user defined signal
Infinite Process. 10790
Infinite Process. 10790
Infinite Process. 10790
Infinite Process. 10790
Killed
piratepanda@SastaPC:~/Documents/oslab/week4$ |
```

.

```
piratepanda@SastaPC: ~
File Edit View Search Terminal Help
piratepanda@SastaPC:~$ kill -TERM 10790
piratepanda@SastaPC:~$ kill -USR1 10790
piratepanda@SastaPC:~$ kill -KILL 10790
piratepanda@SastaPC:~$ |
```