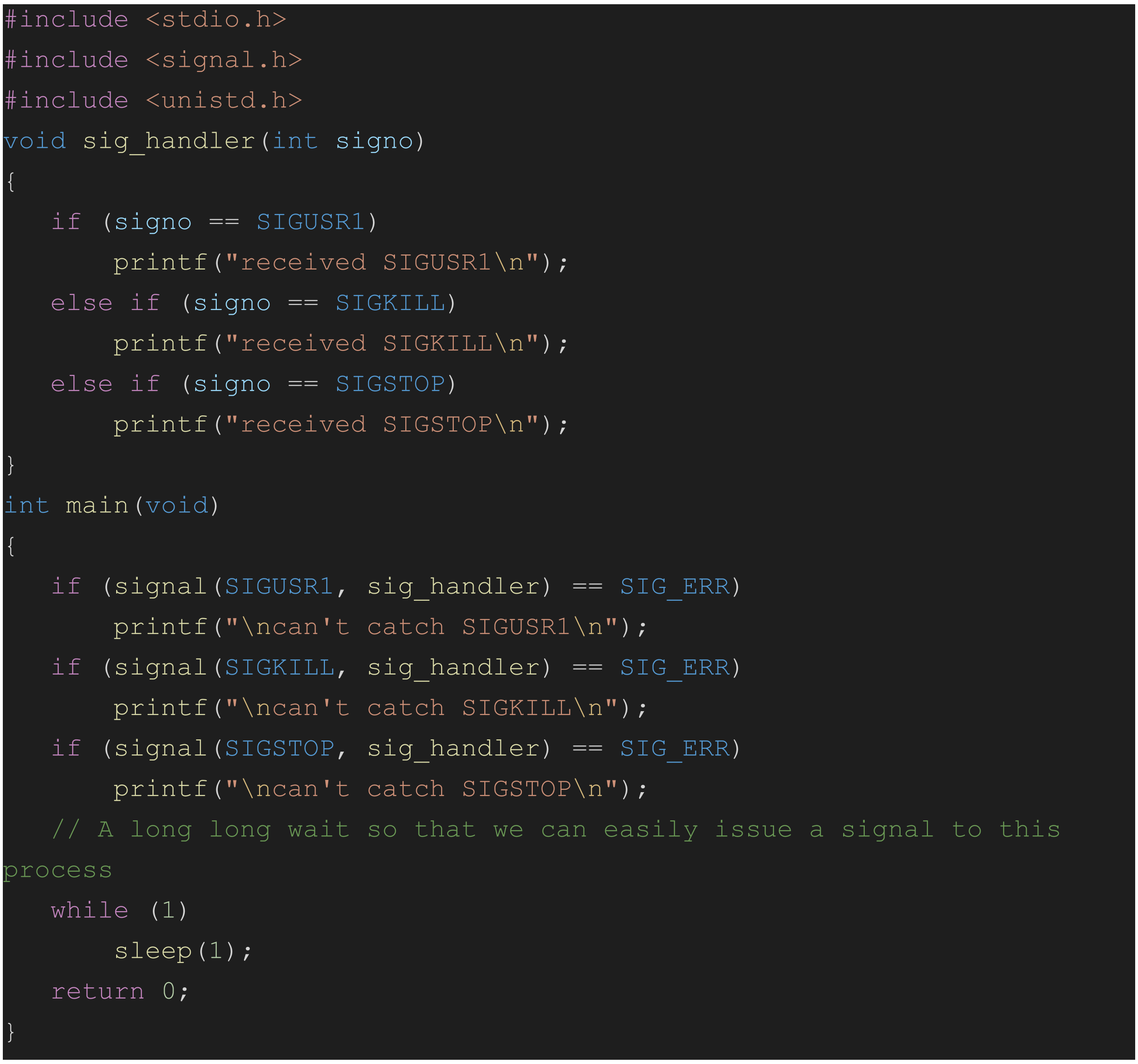
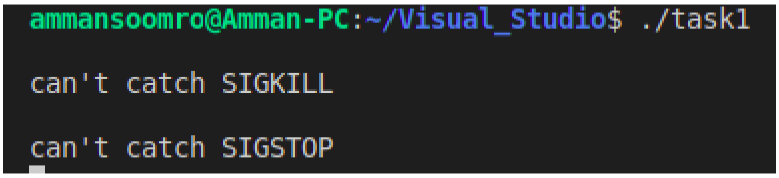
Task 1



```c
#include <stdio.h>
#include <signal.h>
#include <unistd.h>
void sig_handler(int signo)
{
   if (signo == SIGUSR1)
       printf("received SIGUSR1\n");
   else if (signo == SIGKILL)
       printf("received SIGKILL\n");
   else if (signo == SIGSTOP)
       printf("received SIGSTOP\n");
}
int main(void)
{
   if (signal(SIGUSR1, sig_handler) == SIG_ERR)
       printf("\ncan't catch SIGUSR1\n");
   if (signal(SIGKILL, sig_handler) == SIG_ERR)
       printf("\ncan't catch SIGKILL\n");
   if (signal(SIGSTOP, sig_handler) == SIG_ERR)
       printf("\ncan't catch SIGSTOP\n");
   // A long long wait so that we can easily issue a signal to this
process
   while (1)
       sleep(1);
   return 0;
}
```



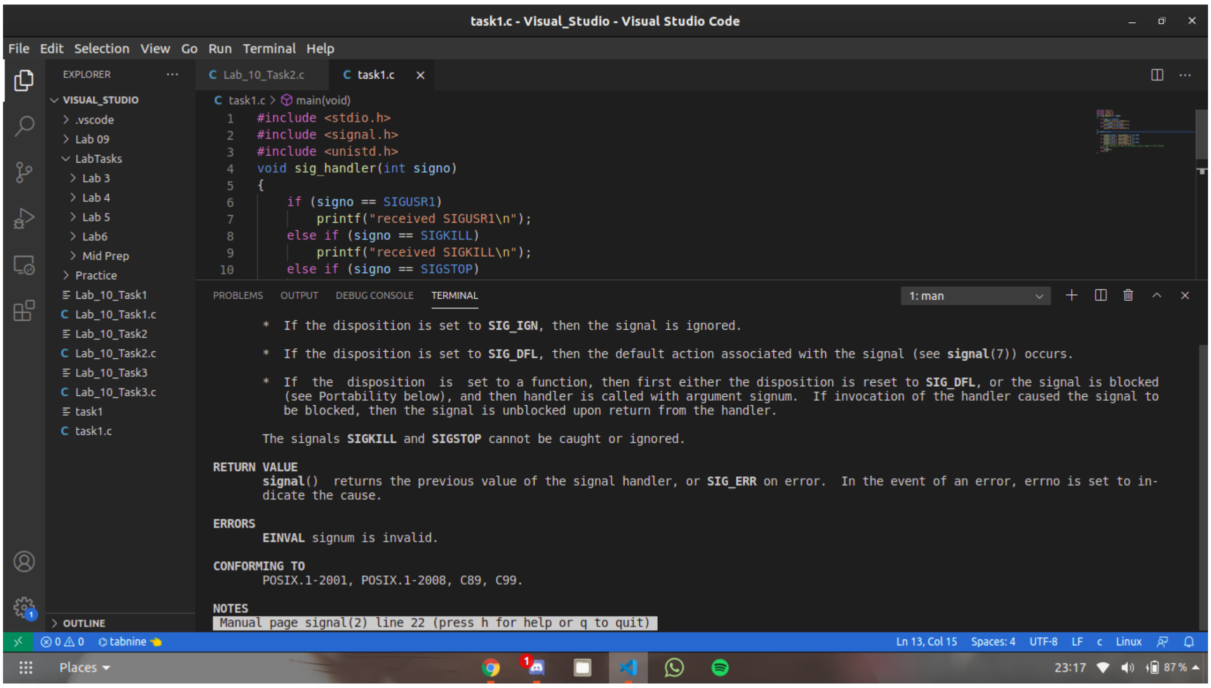
```
ammansoomro@Amman-PC:~/Visual_Studio$ ./task1

can't catch SIGKILL

can't catch SIGSTOP
```

The code is catching exceptions without any signals passed.

I don't know if that is an intended or unintended mistake, I would really request for you to explain it in the next lab and kindly extend this tab a little bit further.

Task 2

According to “man signal” SIGKILL and SIGSTOP cannot be ignored.



```c
#include <stdio.h>
#include <signal.h>
#include <unistd.h>
void sig_handler(int signo)
{
    if (signo == SIGUSR1)
        printf("received SIGUSR1\n");
    else if (signo == SIGKILL)
        printf("received SIGKILL\n");
    else if (signo == SIGSTOP)
```

```
       *  If the disposition is set to SIG_IGN, then the signal is ignored.

       *  If the disposition is set to SIG_DFL, then the default action associated with the signal (see signal(7)) occurs.

       *  If  the  disposition  is  set to a function, then first either the disposition is reset to SIG_DFL, or the signal is blocked
          (see Portability below), and then handler is called with argument signum.  If invocation of the handler caused the signal to
          be blocked, then the signal is unblocked upon return from the handler.

       The signals SIGKILL and SIGSTOP cannot be caught or ignored.

RETURN VALUE
       signal()  returns the previous value of the signal handler, or SIG_ERR on error.  In the event of an error, errno is set to in-
       dicate the cause.

ERRORS
       EINVAL signum is invalid.

CONFORMING TO
       POSIX.1-2001, POSIX.1-2008, C89, C99.

NOTES
 Manual page signal(2) line 22 (press h for help or q to quit)
```