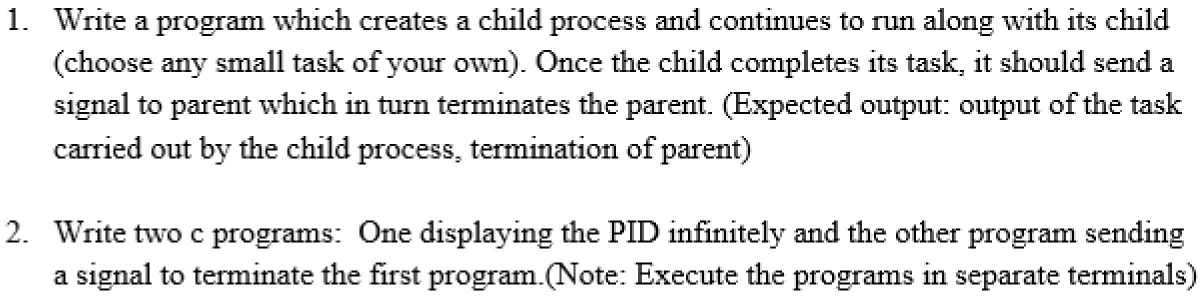
Submission Guidelines

Name of the File
:20BCE1798_CSE2005_Ex8_InterCommunicationProcess

Required Contents: Virtual Box

Reg No:20BCE1798
Name: ANSH GOEL
Course: CSE2005-Operating Systems (Embedded Lab)
Slot: L27+L28

Ex No 2: Shell Programming
Date: 6th April,2022

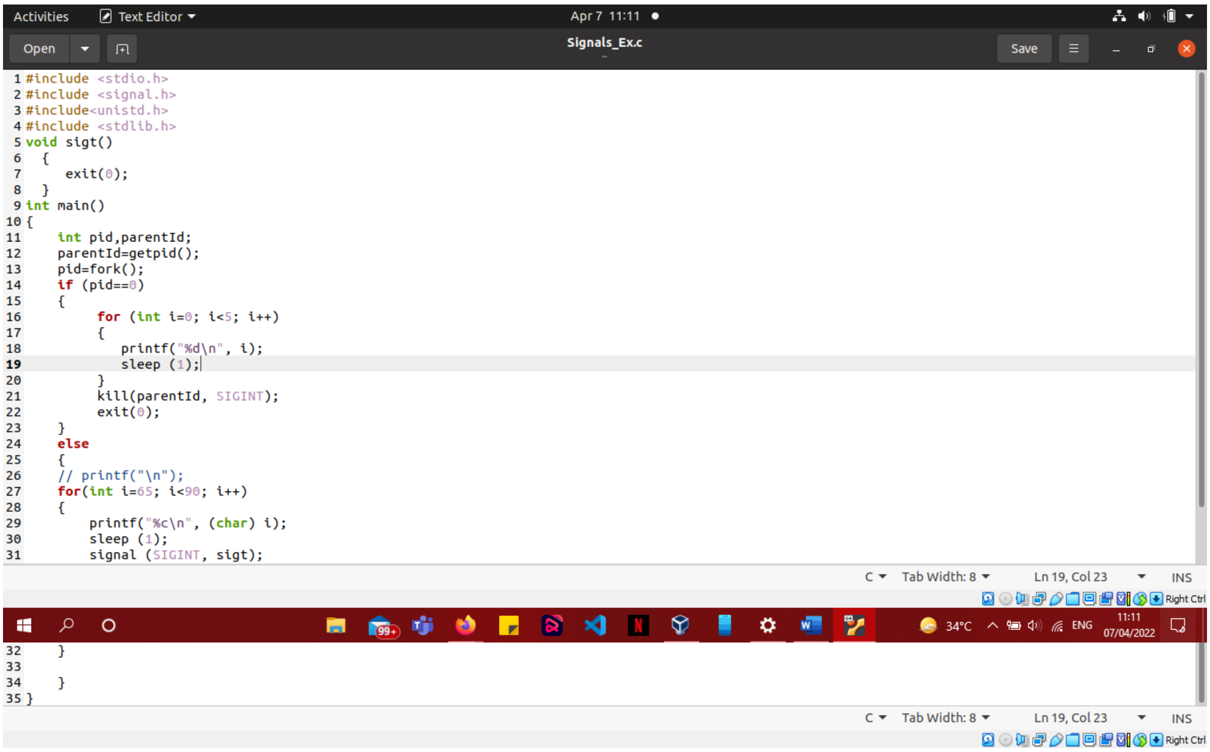


1. Write a program which creates a child process and continues to run along with its child (choose any small task of your own). Once the child completes its task, it should send a signal to parent which in turn terminates the parent. (Expected output: output of the task carried out by the child process, termination of parent)

2. Write two c programs: One displaying the PID infinitely and the other program sending a signal to terminate the first program.(Note: Execute the programs in separate terminals)

Q1)

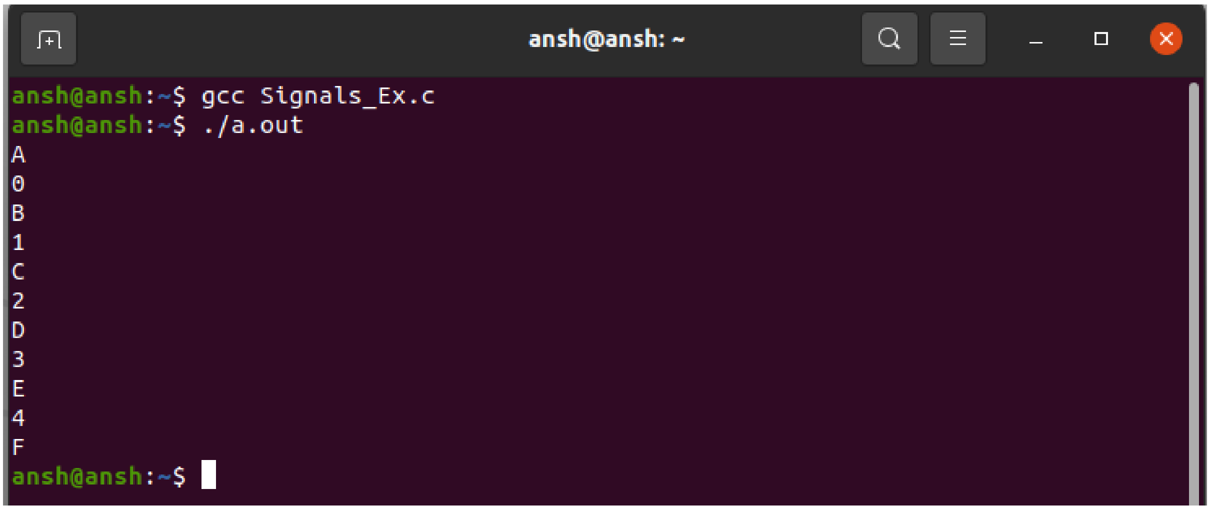
CODE:



```c
#include <stdio.h>
#include <signal.h>
#include<unistd.h>
#include <stdlib.h>
void sigt()
  {
     exit(0);
  }
int main()
{
    int pid,parentId;
    parentId=getpid();
    pid=fork();
    if (pid==0)
    {
         for (int i=0; i<5; i++)
         {
            printf("%d\n", i);
            sleep (1);
         }
         kill(parentId, SIGINT);
         exit(0);
    }
    else
    {
    // printf("\n");
    for(int i=65; i<90; i++)
    {
        printf("%c\n", (char) i);
        sleep (1);
        signal (SIGINT, sigt);
    }

    }
}
```

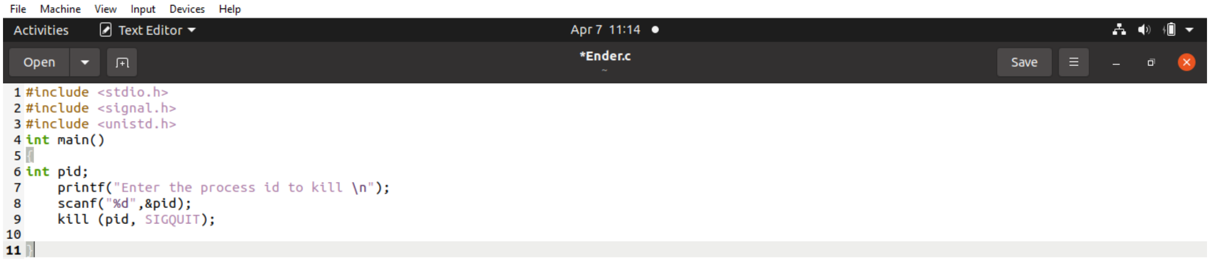
OUTPUT:



```
ansh@ansh:~$ gcc Signals_Ex.c
ansh@ansh:~$ ./a.out
A
0
B
1
C
2
D
3
E
4
F
ansh@ansh:~$
```

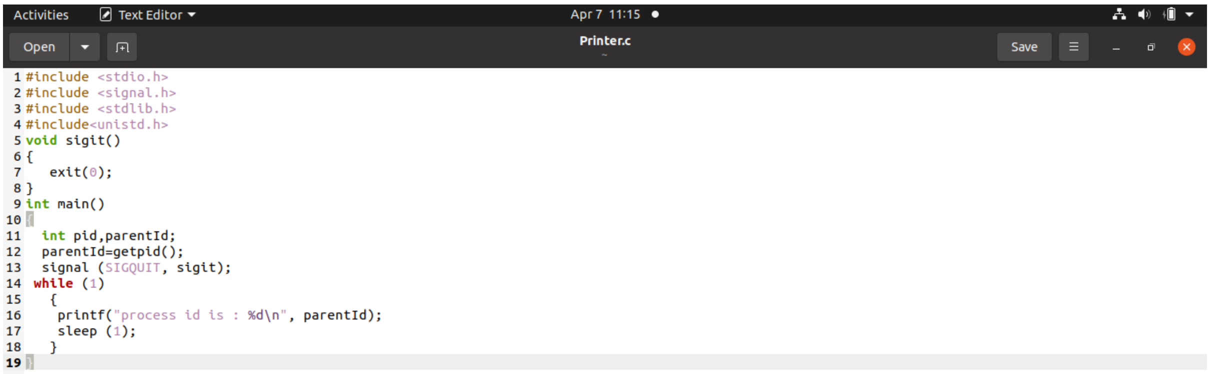
Q2)

CODE:



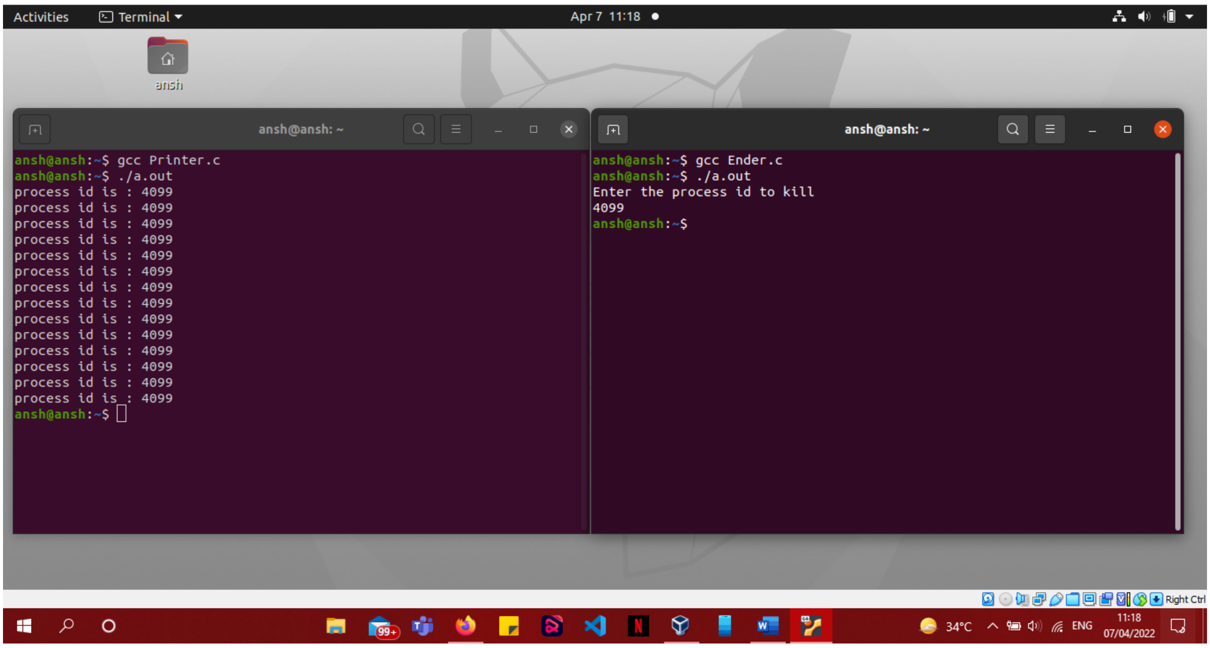
```
#include <stdio.h>
#include <signal.h>
#include <unistd.h>
int main()
{
int pid;
    printf("Enter the process id to kill \n");
    scanf("%d",&pid);
    kill (pid, SIGQUIT);

}
```



```
#include <stdio.h>
#include <signal.h>
#include <stdlib.h>
#include<unistd.h>
void sigit()
{
   exit(0);
}
int main()
{
  int pid,parentId;
  parentId=getpid();
  signal (SIGQUIT, sigit);
 while (1)
   {
    printf("process id is : %d\n", parentId);
    sleep (1);
   }
}
```

OUTPUT:



```
ansh@ansh:~$ gcc Printer.c
ansh@ansh:~$ ./a.out
process id is : 4099
process id is : 4099
process id is : 4099
process id is : 4099
process id is : 4099
process id is : 4099
process id is : 4099
process id is : 4099
process id is : 4099
process id is : 4099
process id is : 4099
process id is : 4099
process id is : 4099
process id is : 4099
ansh@ansh:~$
```

```
ansh@ansh:~$ gcc Ender.c
ansh@ansh:~$ ./a.out
Enter the process id to kill
4099
ansh@ansh:~$
```

NAME – ANSH GOEL REGISTER NO.- 20BCE1798