Laboratory Assignment 4

On

Design of Operating System (CSE 4049)

Submitted by

Name	**:**	**Anurag Jain**
Reg. No.	**:**	**2141010002**
Semester	**:**	**5th**
Branch	**:**	**CSE**
Section	**:**	**15 (‘O’)**
Session	**:**	**2023-2024**
Admission Batch	**:**	**2021**

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING FACULTY OF ENGINEERING & TECHNOLOGY (ITER)

SIKSHA ‘O’ ANUSANDHAN DEEMED TO BE UNIVERSITY

BHUBANESWAR, ODISHA – 751030

Laboratory Assignment 4 Subject: Design Of Operating System

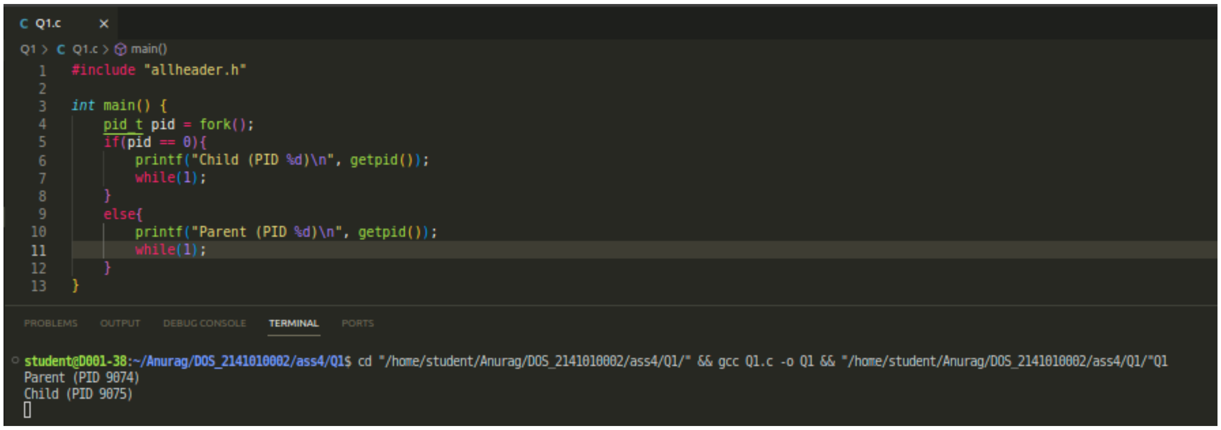
Subject code : CSE 4049

Assignment 4: Familiarization with Process Management in Unix environment.

Objective of this Assignment:

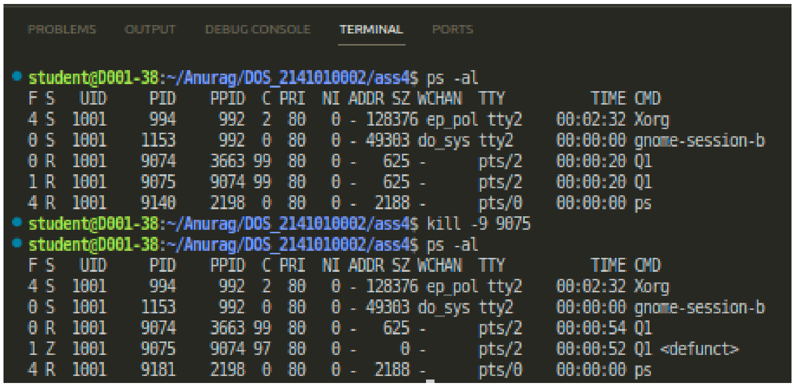
- To trace the different states of a process during its execution.
- To learn the use of different system calls such as (fork(),vfork(),wait(),execl()) for process handling in Unix environment.

1. Write a C program to create a child process using fork() system call. The child process will print the message "Child" with its process identifier and then continue in an indefinite loop. The parent process will print the message "Parent" with its process identifier and then continue in an indefinite loop.
   a) Run the program and trace the state of both processes.
   b) Terminate the child process. Then trace the state of processes.
   c) Run the program and trace the state of both processes. Terminate the parent process. Then trace the state of processes.
   d) Modify the program so that the parent process after displaying the message will wait for child process to complete its task. Again run the program and trace the state of both processes.
   e) Terminate the child process. Then trace the state of processes.

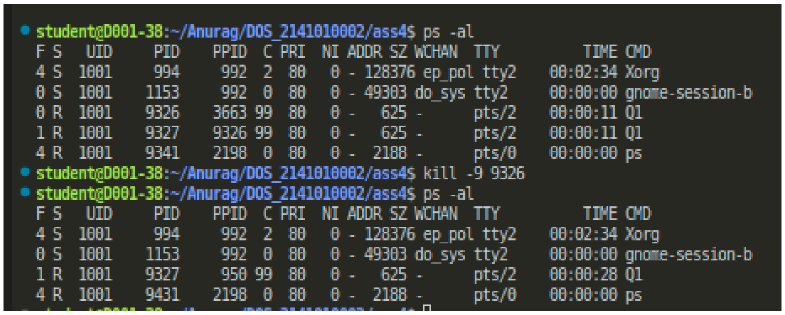


```c
#include "allheader.h"

int main() {
    pid_t pid = fork();
    if(pid == 0){
        printf("Child (PID %d)\n", getpid());
        while(1);
    }
    else{
        printf("Parent (PID %d)\n", getpid());
        while(1);
    }
}
```

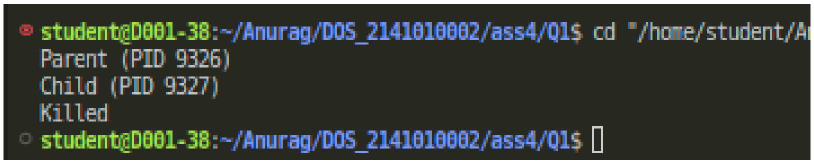
```
student@D001-38:~/Anurag/DOS_2141010002/ass4/Q1$ cd "/home/student/Anurag/DOS_2141010002/ass4/Q1/" && gcc Q1.c -o Q1 && "/home/student/Anurag/DOS_2141010002/ass4/Q1/"Q1
Parent (PID 9074)
Child (PID 9075)
```



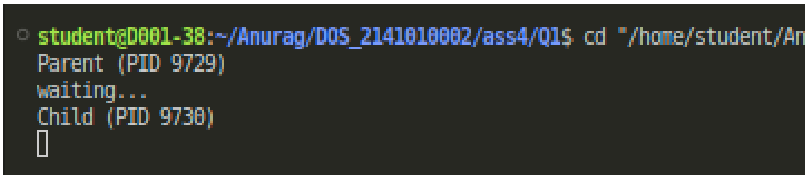
```
student@D001-38:~/Anurag/DOS_2141010002/ass4$ ps -al
 F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
 4 S  1001     994     992  2  80   0 - 128376 ep_pol tty2    00:02:32 Xorg
 0 S  1001    1153     992  0  80   0 - 49303 do_sys tty2     00:00:00 gnome-session-b
 0 R  1001    9074    3663 99  80   0 -   625 -      pts/2    00:00:20 Q1
 1 R  1001    9075    9074 99  80   0 -   625 -      pts/2    00:00:20 Q1
 4 R  1001    9140    2198  0  80   0 -  2188 -      pts/0    00:00:00 ps
student@D001-38:~/Anurag/DOS_2141010002/ass4$ kill -9 9075
student@D001-38:~/Anurag/DOS_2141010002/ass4$ ps -al
 F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
 4 S  1001     994     992  2  80   0 - 128376 ep_pol tty2    00:02:32 Xorg
 0 S  1001    1153     992  0  80   0 - 49303 do_sys tty2     00:00:00 gnome-session-b
 0 R  1001    9074    3663 99  80   0 -   625 -      pts/2    00:00:54 Q1
 1 Z  1001    9075    9074 97  80   0 -     0 -      pts/2    00:00:52 Q1 <defunct>
 4 R  1001    9181    2198  0  80   0 -  2188 -      pts/0    00:00:00 ps
```



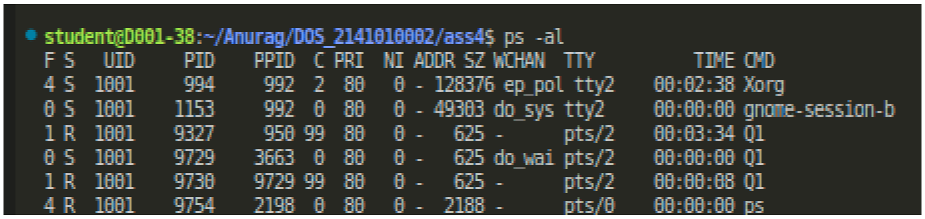
```
student@D001-38:~/Anurag/DOS_2141010002/ass4$ ps -al
 F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
 4 S  1001     994     992  2  80   0 - 128376 ep_pol tty2    00:02:34 Xorg
 0 S  1001    1153     992  0  80   0 - 49303 do_sys tty2     00:00:00 gnome-session-b
 0 R  1001    9326    3663 99  80   0 -   625 -      pts/2    00:00:11 Q1
 1 R  1001    9327    9326 99  80   0 -   625 -      pts/2    00:00:11 Q1
 4 R  1001    9341    2198  0  80   0 -  2188 -      pts/0    00:00:00 ps
student@D001-38:~/Anurag/DOS_2141010002/ass4$ kill -9 9326
student@D001-38:~/Anurag/DOS_2141010002/ass4$ ps -al
 F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
 4 S  1001     994     992  2  80   0 - 128376 ep_pol tty2    00:02:34 Xorg
 0 S  1001    1153     992  0  80   0 - 49303 do_sys tty2     00:00:00 gnome-session-b
 1 R  1001    9327     950 99  80   0 -   625 -      pts/2    00:00:28 Q1
 4 R  1001    9431    2198  0  80   0 -  2188 -      pts/0    00:00:00 ps
```



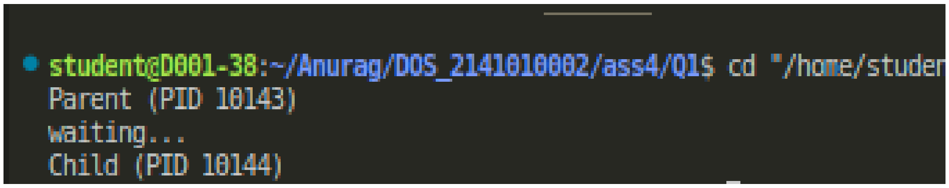
```
student@D001-38:~/Anurag/DOS_2141010002/ass4/Q1$ cd "/home/student/A
Parent (PID 9326)
Child (PID 9327)
Killed
student@D001-38:~/Anurag/DOS_2141010002/ass4/Q1$
```



```
student@D001-38:~/Anurag/DOS_2141010002/ass4/Q1$ cd "/home/student/A
Parent (PID 9729)
waiting...
Child (PID 9730)
```

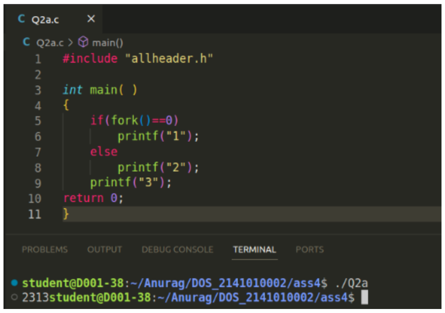


```
student@D001-38:~/Anurag/DOS_2141010002/ass4$ ps -al
 F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
 4 S  1001     994     992  2  80   0 - 128376 ep_pol tty2    00:02:38 Xorg
 0 S  1001    1153     992  0  80   0 - 49303 do_sys tty2     00:00:00 gnome-session-b
 1 R  1001    9327     950 99  80   0 -   625 -      pts/2    00:03:34 Q1
 0 S  1001    9729    3663  0  80   0 -   625 do_wai pts/2    00:00:00 Q1
 1 R  1001    9730    9729 99  80   0 -   625 -      pts/2    00:00:08 Q1
 4 R  1001    9754    2198  0  80   0 -  2188 -      pts/0    00:00:00 ps
```



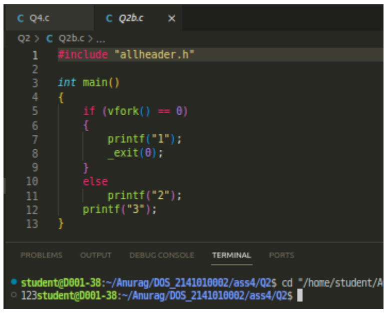
```
student@D001-38:~/Anurag/DOS_2141010002/ass4/Q1$ cd "/home/studen
Parent (PID 10143)
waiting...
Child (PID 10144)
```

2. Trace the output of the following codes:



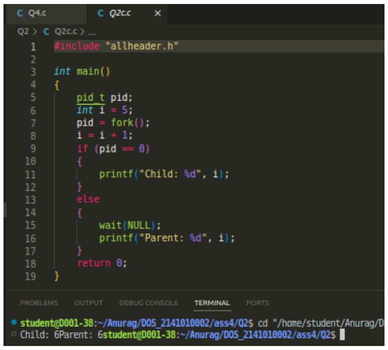
```c
#include "allheader.h"

int main( )
{
    if(fork()==0)
        printf("1");
    else
        printf("2");
    printf("3");
return 0;
}
```

```
student@D001-38:~/Anurag/DOS_2141010002/ass4$ ./Q2a
2313student@D001-38:~/Anurag/DOS_2141010002/ass4$ 
```



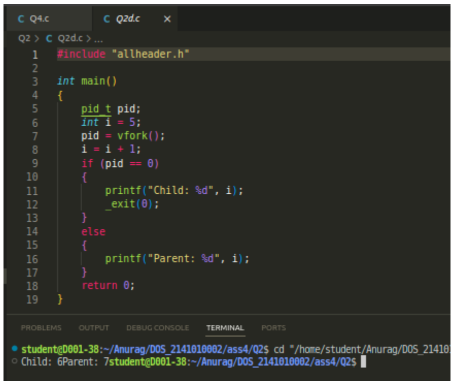
```c
#include "allheader.h"

int main()
{
    if (vfork() == 0)
    {
        printf("1");
        _exit(0);
    }
    else
        printf("2");
    printf("3");
}
```

```
student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ cd "/home/student/A
123student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ 
```



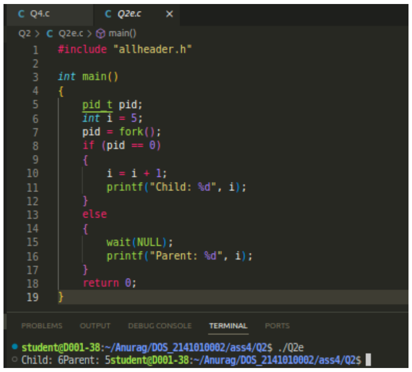
```c
#include "allheader.h"

int main()
{
    pid_t pid;
    int i = 5;
    pid = fork();
    i = i + 1;
    if (pid == 0)
    {
        printf("Child: %d", i);
    }
    else
    {
        wait(NULL);
        printf("Parent: %d", i);
    }
    return 0;
}
```

```
student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ cd "/home/student/Anurag/D
Child: 6Parent: 6student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ 
```



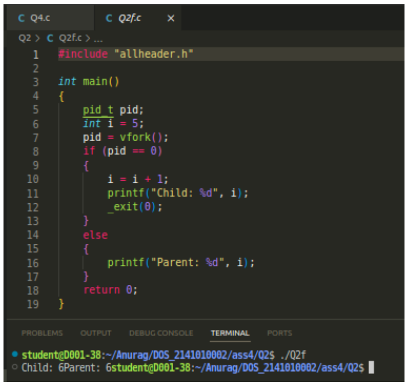
```c
#include "allheader.h"

int main()
{
    pid_t pid;
    int i = 5;
    pid = vfork();
    i = i + 1;
    if (pid == 0)
    {
        printf("Child: %d", i);
        _exit(0);
    }
    else
    {
        printf("Parent: %d", i);
    }
    return 0;
}
```

```
student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ cd "/home/student/Anurag/DOS_21410
Child: 6Parent: 7student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ 
```



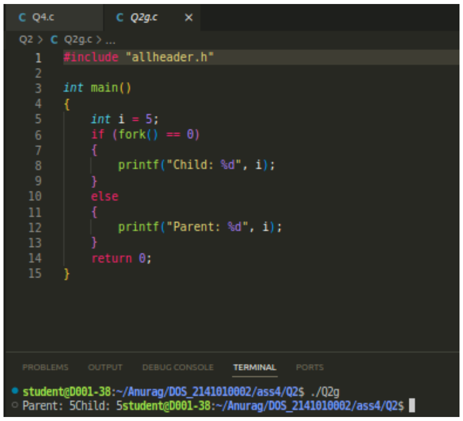
```c
#include "allheader.h"

int main()
{
    pid_t pid;
    int i = 5;
    pid = fork();
    if (pid == 0)
    {
        i = i + 1;
        printf("Child: %d", i);
    }
    else
    {
        wait(NULL);
        printf("Parent: %d", i);
    }
    return 0;
}
```

```
student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ ./Q2e
Child: 6Parent: 5student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ 
```



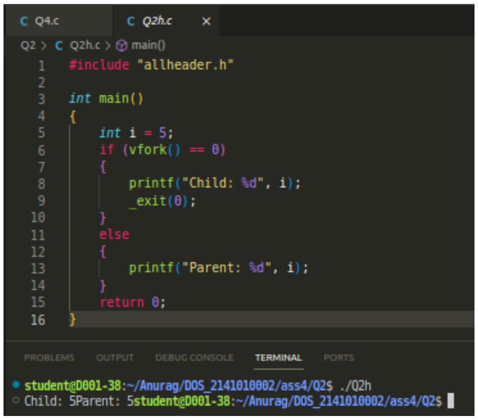
```c
#include "allheader.h"

int main()
{
    pid_t pid;
    int i = 5;
    pid = vfork();
    if (pid == 0)
    {
        i = i + 1;
        printf("Child: %d", i);
        _exit(0);
    }
    else
    {
        printf("Parent: %d", i);
    }
    return 0;
}
```

```
student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ ./Q2f
Child: 6Parent: 6student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ 
```



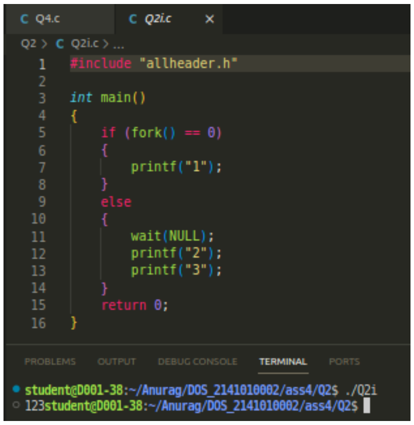
```c
#include "allheader.h"

int main()
{
    int i = 5;
    if (fork() == 0)
    {
        printf("Child: %d", i);
    }
    else
    {
        printf("Parent: %d", i);
    }
    return 0;
}
```

```
student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ ./Q2g
Parent: 5Child: 5student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ 
```



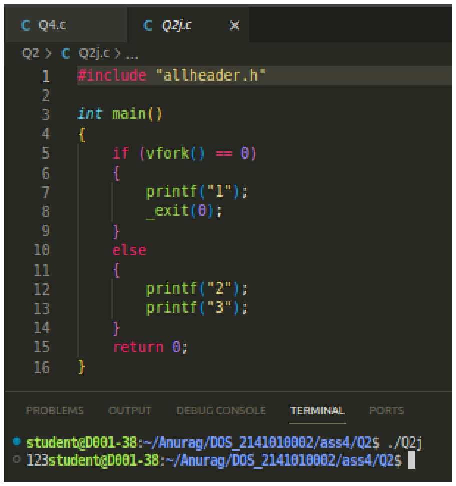
```c
#include "allheader.h"

int main()
{
    int i = 5;
    if (vfork() == 0)
    {
        printf("Child: %d", i);
        _exit(0);
    }
    else
    {
        printf("Parent: %d", i);
    }
    return 0;
}
```

```
student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ ./Q2h
Child: 5Parent: 5student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ 
```



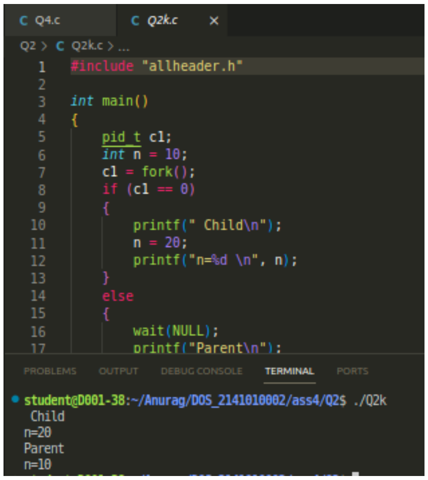
```c
#include "allheader.h"

int main()
{
    if (fork() == 0)
    {
        printf("1");
    }
    else
    {
        wait(NULL);
        printf("2");
        printf("3");
    }
    return 0;
}
```

```
student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ ./Q2i
123student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ 
```



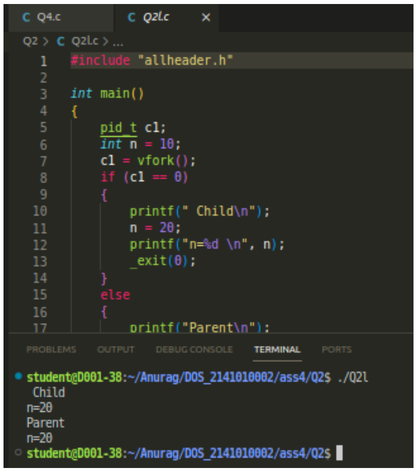
```c
#include "allheader.h"

int main()
{
    if (vfork() == 0)
    {
        printf("1");
        _exit(0);
    }
    else
    {
        printf("2");
        printf("3");
    }
    return 0;
}
```

```
student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ ./Q2j
123student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ 
```



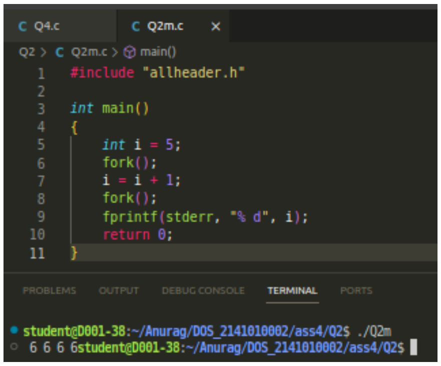
```c
#include "allheader.h"

int main()
{
    pid_t cl;
    int n = 10;
    cl = fork();
    if (cl == 0)
    {
        printf(" Child\n");
        n = 20;
        printf("n=%d \n", n);
    }
    else
    {
        wait(NULL);
        printf("Parent\n");
```

```
student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ ./Q2k
 Child
n=20
Parent
n=10
```



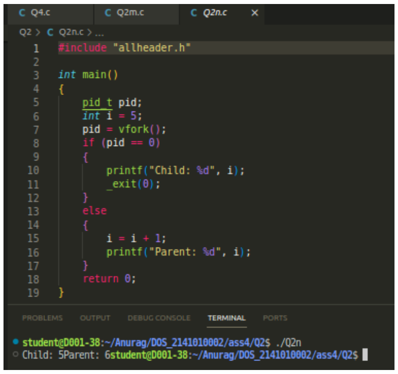
```c
#include "allheader.h"

int main()
{
    pid_t cl;
    int n = 10;
    cl = vfork();
    if (cl == 0)
    {
        printf(" Child\n");
        n = 20;
        printf("n=%d \n", n);
        _exit(0);
    }
    else
    {
        printf("Parent\n");
```

```
student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ ./Q2l
 Child
n=20
Parent
n=20
student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ 
```



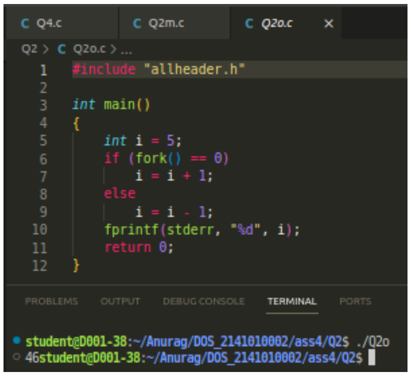
```c
#include "allheader.h"

int main()
{
    int i = 5;
    fork();
    i = i + 1;
    fork();
    fprintf(stderr, "% d", i);
    return 0;
}
```

```
student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ ./Q2m
 6 6 6 6student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ 
```



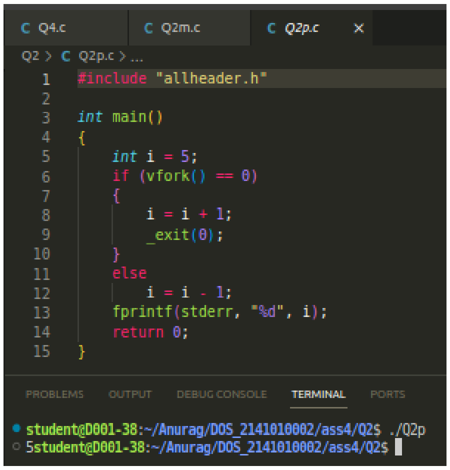
```c
#include "allheader.h"

int main()
{
    pid_t pid;
    int i = 5;
    pid = vfork();
    if (pid == 0)
    {
        printf("Child: %d", i);
        _exit(0);
    }
    else
    {
        i = i + 1;
        printf("Parent: %d", i);
    }
    return 0;
}
```

```
student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ ./Q2n
Child: 5Parent: 6student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ 
```



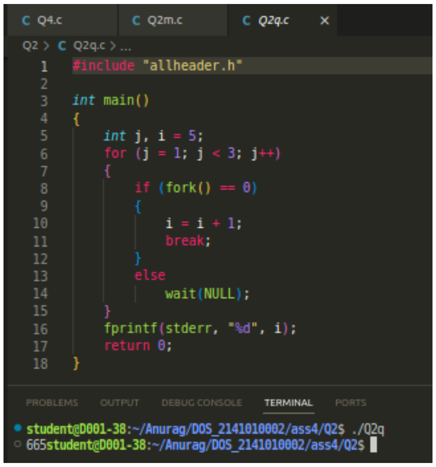
```c
#include "allheader.h"

int main()
{
    int i = 5;
    if (fork() == 0)
        i = i + 1;
    else
        i = i - 1;
    fprintf(stderr, "%d", i);
    return 0;
}
```

```
student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ ./Q2o
46student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ 
```



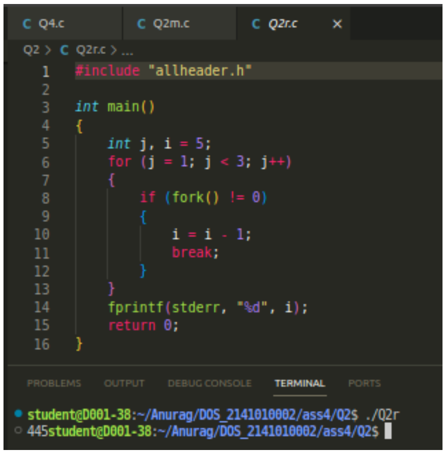
```c
#include "allheader.h"

int main()
{
    int i = 5;
    if (vfork() == 0)
    {
        i = i + 1;
        _exit(0);
    }
    else
        i = i - 1;
    fprintf(stderr, "%d", i);
    return 0;
}
```

```
student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ ./Q2p
5student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ 
```



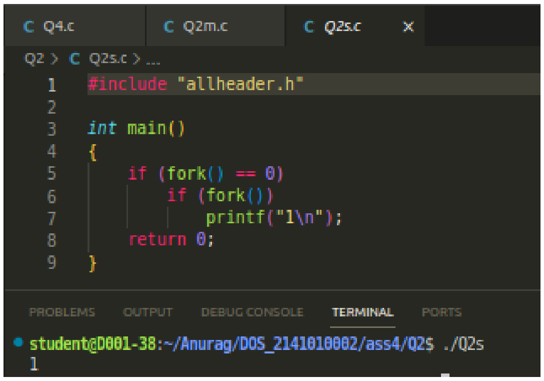
```c
#include "allheader.h"

int main()
{
    int j, i = 5;
    for (j = 1; j < 3; j++)
    {
        if (fork() == 0)
        {
            i = i + 1;
            break;
        }
        else
            wait(NULL);
    }
    fprintf(stderr, "%d", i);
    return 0;
}
```

```
student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ ./Q2q
665student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ 
```



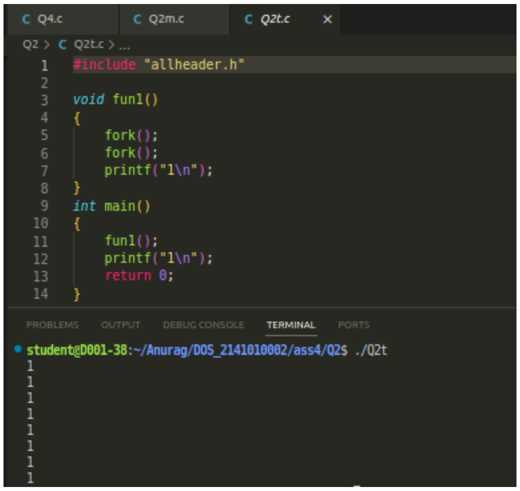
```c
#include "allheader.h"

int main()
{
    int j, i = 5;
    for (j = 1; j < 3; j++)
    {
        if (fork() != 0)
        {
            i = i - 1;
            break;
        }
    }
    fprintf(stderr, "%d", i);
    return 0;
}
```

```
student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ ./Q2r
445student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ 
```



```c
#include "allheader.h"

int main()
{
    if (fork() == 0)
        if (fork())
            printf("1\n");
    return 0;
}
```

```
student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ ./Q2s
1
```

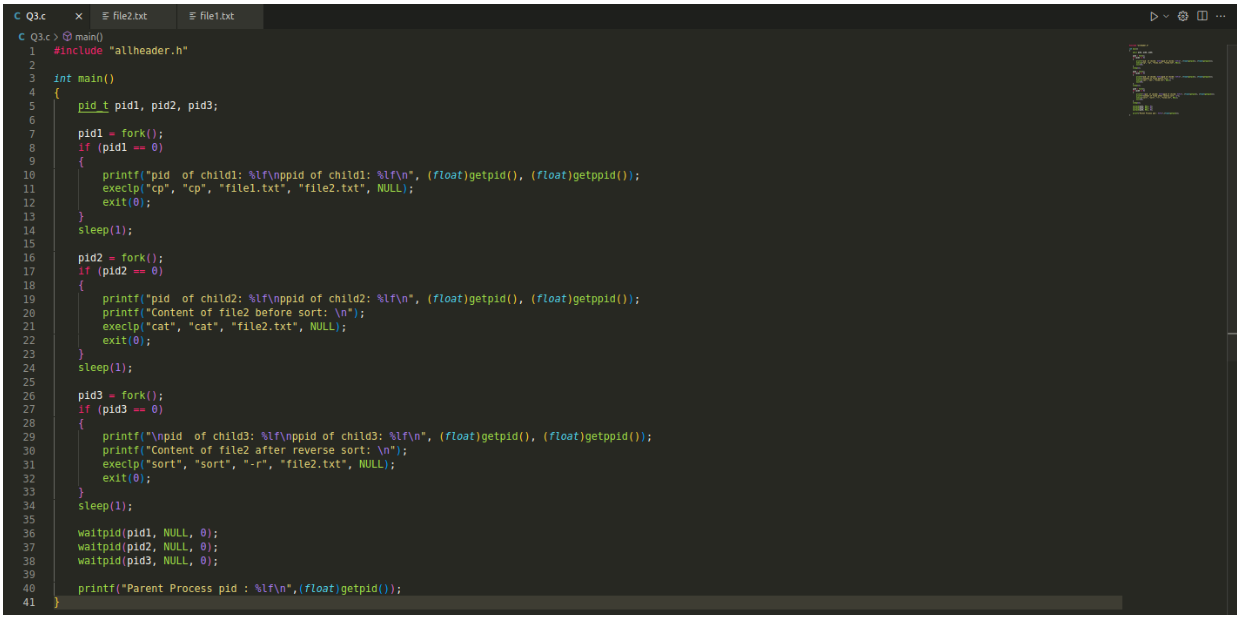


```c
#include "allheader.h"

void fun1()
{
    fork();
    fork();
    printf("1\n");
}
int main()
{
    fun1();
    printf("1\n");
    return 0;
}
```

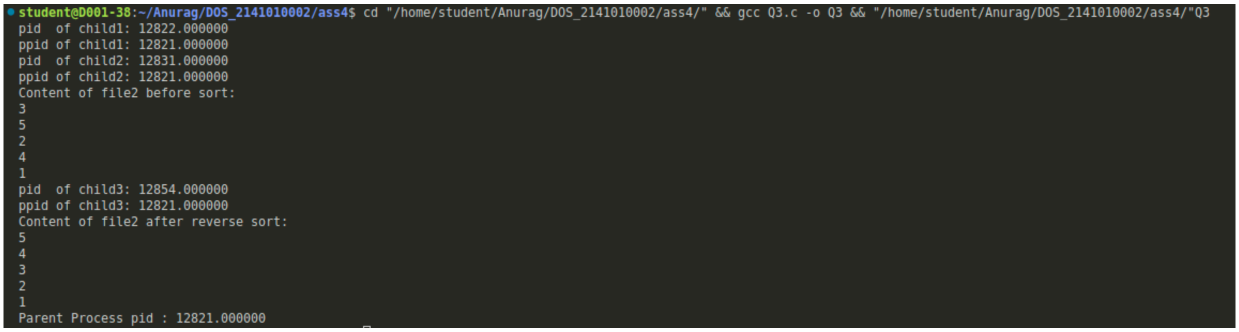
```
student@D001-38:~/Anurag/DOS_2141010002/ass4/Q2$ ./Q2t
1
1
1
1
1
1
1
1
```

3. Write a C program that will create three child process to perform the following operations respectively:

- First child will copy the content of file1 to file2
- Second child will display the content of file2
- Third child will display the sorted content of file2 in reverse order.
- Each child process being created will display its id and its parent process id with appropriate message.
- The parent process will be delayed for 1 second after creation of each child process. It will display appropriate message with its id after completion of all the child processes.

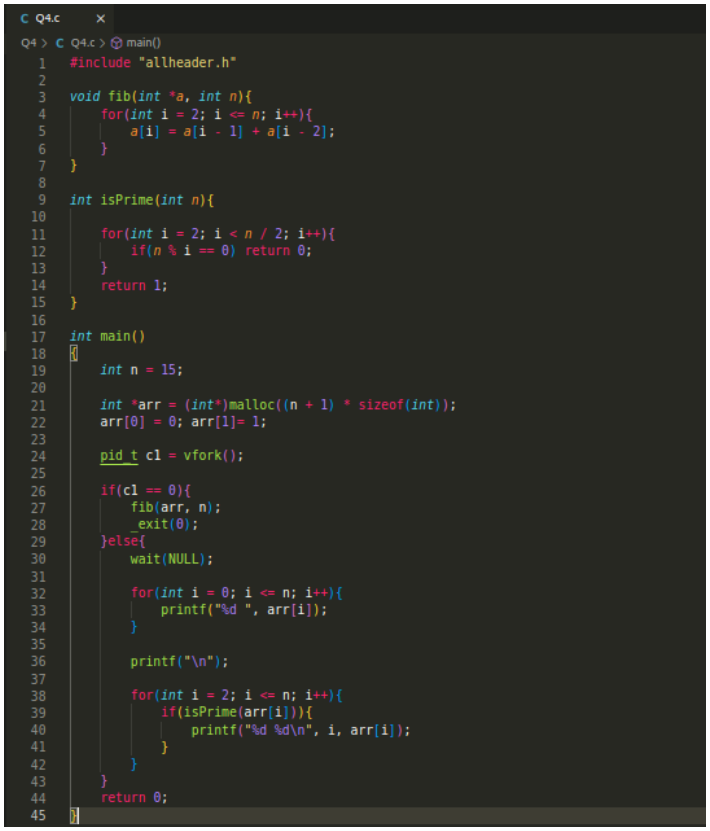


```c
#include "allheader.h"

int main()
{
    pid_t pid1, pid2, pid3;

    pid1 = fork();
    if (pid1 == 0)
    {
        printf("pid  of child1: %lf\nppid of child1: %lf\n", (float)getpid(), (float)getppid());
        execlp("cp", "cp", "file1.txt", "file2.txt", NULL);
        exit(0);
    }
    sleep(1);

    pid2 = fork();
    if (pid2 == 0)
    {
        printf("pid  of child2: %lf\nppid of child2: %lf\n", (float)getpid(), (float)getppid());
        printf("Content of file2 before sort: \n");
        execlp("cat", "cat", "file2.txt", NULL);
        exit(0);
    }
    sleep(1);

    pid3 = fork();
    if (pid3 == 0)
    {
        printf("\npid  of child3: %lf\nppid of child3: %lf\n", (float)getpid(), (float)getppid());
        printf("Content of file2 after reverse sort: \n");
        execlp("sort", "sort", "-r", "file2.txt", NULL);
        exit(0);
    }
    sleep(1);

    waitpid(pid1, NULL, 0);
    waitpid(pid2, NULL, 0);
    waitpid(pid3, NULL, 0);

    printf("Parent Process pid : %lf\n",(float)getpid());
}
```

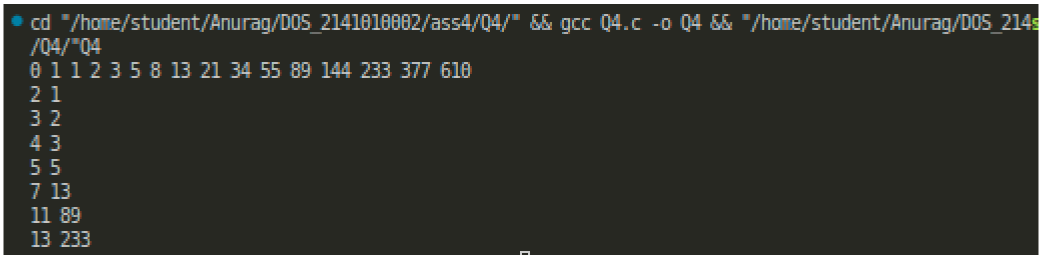


```
student@D001-38:~/Anurag/DOS_2141010002/ass4$ cd "/home/student/Anurag/DOS_2141010002/ass4/" && gcc Q3.c -o Q3 && "/home/student/Anurag/DOS_2141010002/ass4/"Q3
pid  of child1: 12822.000000
ppid of child1: 12821.000000
pid  of child2: 12831.000000
ppid of child2: 12821.000000
Content of file2 before sort:
3
5
2
4
1
pid  of child3: 12854.000000
ppid of child3: 12821.000000
Content of file2 after reverse sort:
5
4
3
2
1
Parent Process pid : 12821.000000
```

4. Write a C program that will create a child process to generate a Fibonacci series of specified length and store it in an array. The parent process will wait for the child to complete its task and then display the Fibonacci series and then display the prime Fibonacci number in the series along with its position with appropriate message.



```c
#include "allheader.h"

void fib(int *a, int n){
    for(int i = 2; i <= n; i++){
        a[i] = a[i - 1] + a[i - 2];
    }
}

int isPrime(int n){

    for(int i = 2; i < n / 2; i++){
        if(n % i == 0) return 0;
    }
    return 1;
}

int main()
{
    int n = 15;

    int *arr = (int*)malloc((n + 1) * sizeof(int));
    arr[0] = 0; arr[1]= 1;

    pid_t cl = vfork();

    if(cl == 0){
        fib(arr, n);
        _exit(0);
    }else{
        wait(NULL);

        for(int i = 0; i <= n; i++){
            printf("%d ", arr[i]);
        }

        printf("\n");

        for(int i = 2; i <= n; i++){
            if(isPrime(arr[i])){
                printf("%d %d\n", i, arr[i]);
            }
        }
    }
    return 0;
}
```



```
● cd "/home/student/Anurag/DOS_2141010002/ass4/Q4/" && gcc Q4.c -o Q4 && "/home/student/Anurag/DOS_214
  /Q4/"Q4
  0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610
  2 1
  3 2
  4 3
  5 5
  7 13
  11 89
  13 233
```