Graphic Era HILL UNIVERSITY

Established by an Act of the State Legislature of Uttarakhand (Adhiniyam Sankhya 12 of 2011)

Term Work

On

OPERATING SYSTEM

(PCS 506)

Submitted to:	**Submitted by:**
Dr. Pardeep Singh	Prahlad Singh Aswal
Associate Professor	University Roll. No.: 2018550
GEHU, D. Dun	Class Roll No./Section: 39/A

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GRAPHIC ERA HILL UNIVERSITY, DEHRADUN

Graphic Era HILL UNIVERSITY

Established by an Act of the State Legislature of Uttarakhand (Adhiniyam Sankhya 12 of 2011)

DEPARTMENT OF CSE
STUDENT LAB REPORT SHEET

Photograph
Passport Size

Name of Student ....**Mob. No** ....

Address Permanent ....

Father's Name .... **Occupation** .... **Mob. No** ....

Mother's Name .... **Occupation** .... **Mob. No** ....

Section .... **Branch** .... **Semester** .... **Class Roll No** .... **Grade** **A B C**

Local Address ....**Email** .... **Marks** **5 3 1**

S.No.	Practical	D.O.P.	Date of Submission	Grade (Viva)	Grade (Report File)	Total Marks (out of 10)	Student's Signature	Teacher's Signature
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								

PRACTICAL 1

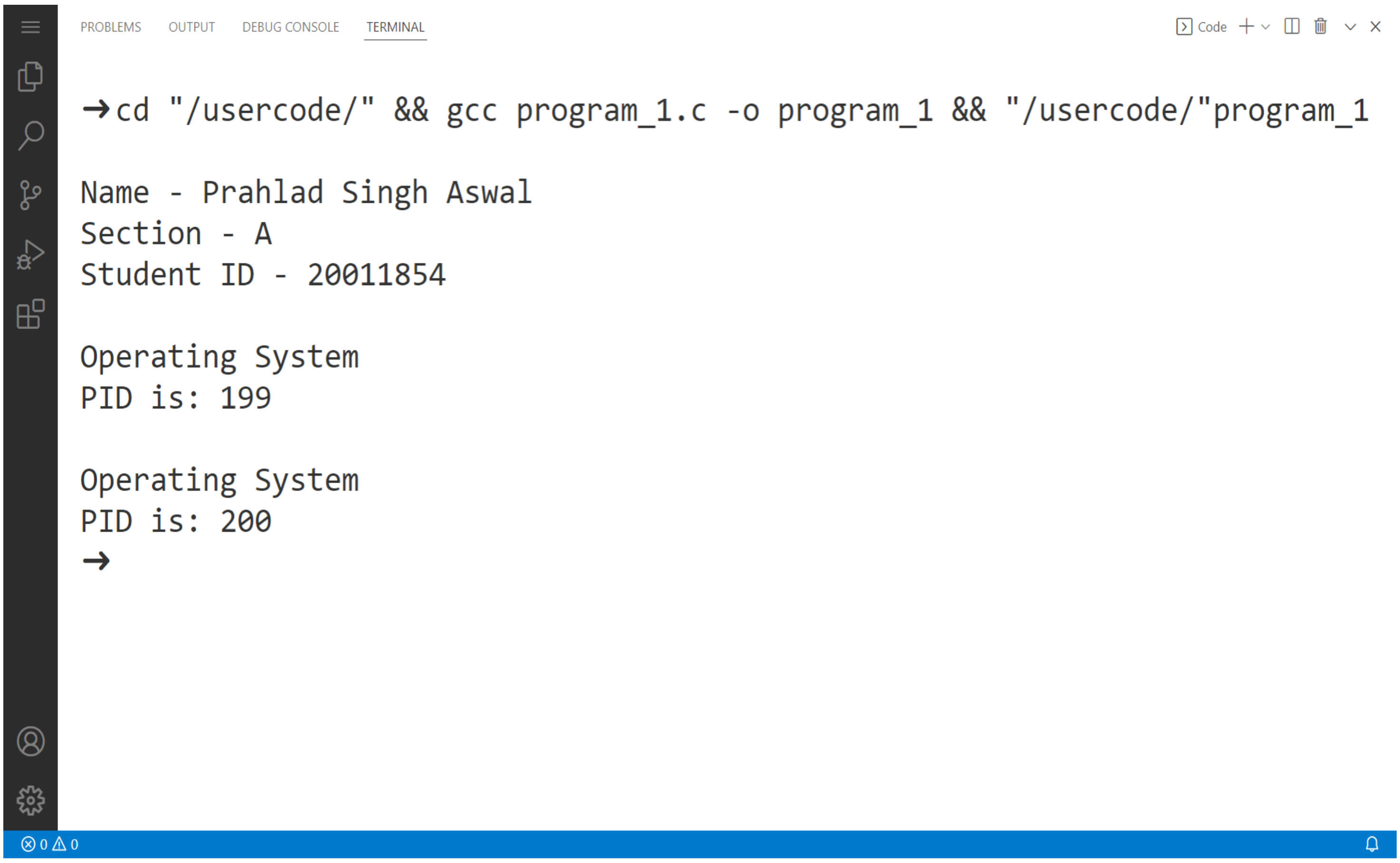
Question: Write a C program to demonstrate the use of fork( ) system call.

About Fork() function:
We use the fork() system call to create a new process from the calling process by duplicating it. fork() system call is used to create child processes in a C program. fork() is used where parallel processing is required in application. The fork() system function is defined in the headers **unistd.h**.
PID: - Process ID

Source Code:

```
#include <stdio.h>
#include <unistd.h>
int main()
{
   printf("\nName - Prahlad Singh Aswal\nSection - A\n");
   printf("Student ID - 20011854\n");

   fork();
   printf("\nOperating System\n");
   printf("PID is: %d\n", getpid());

   return 0;
}
```

Output



```
➜ cd "/usercode/" && gcc program_1.c -o program_1 && "/usercode/"program_1

Name - Prahlad Singh Aswal
Section - A
Student ID - 20011854

Operating System
PID is: 199

Operating System
PID is: 200
➜
```

PRACTICAL 2

Question: Write a C program in which parent process computes the sum of even numbers and child process computes the sum of odd number stored in an array using a fork().

About Fork( ) Function:

Fork system call is used for creating a new process, which is called *child process*, which runs concurrently with the process that makes the fork() call (parent process). After a new child process is created, both processes will execute the next instruction following the fork() system call.

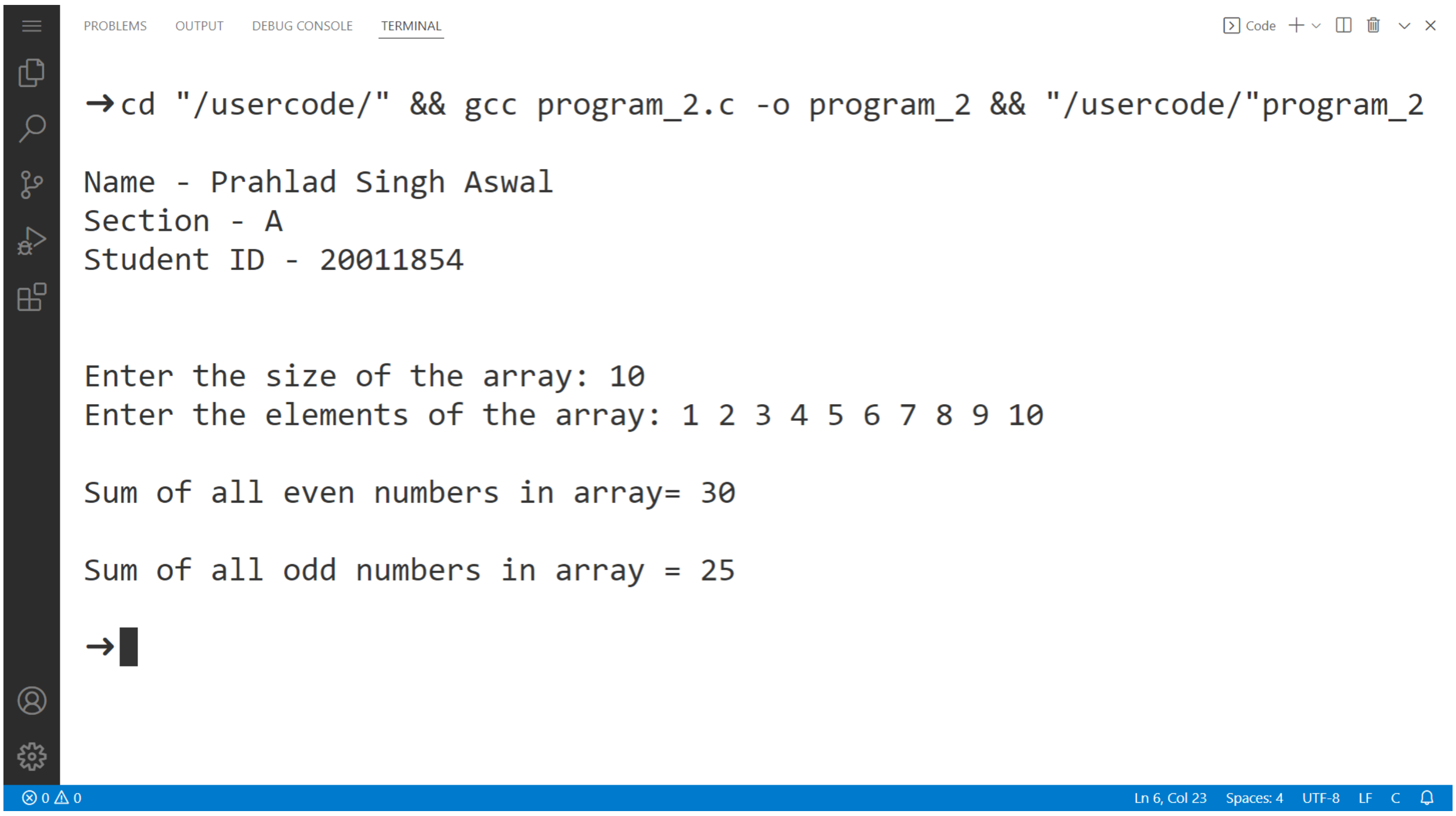
In this program, parent process computes the sum of even numbers in array and child process computer the sum of odd numbers in the array.

Source Code:-

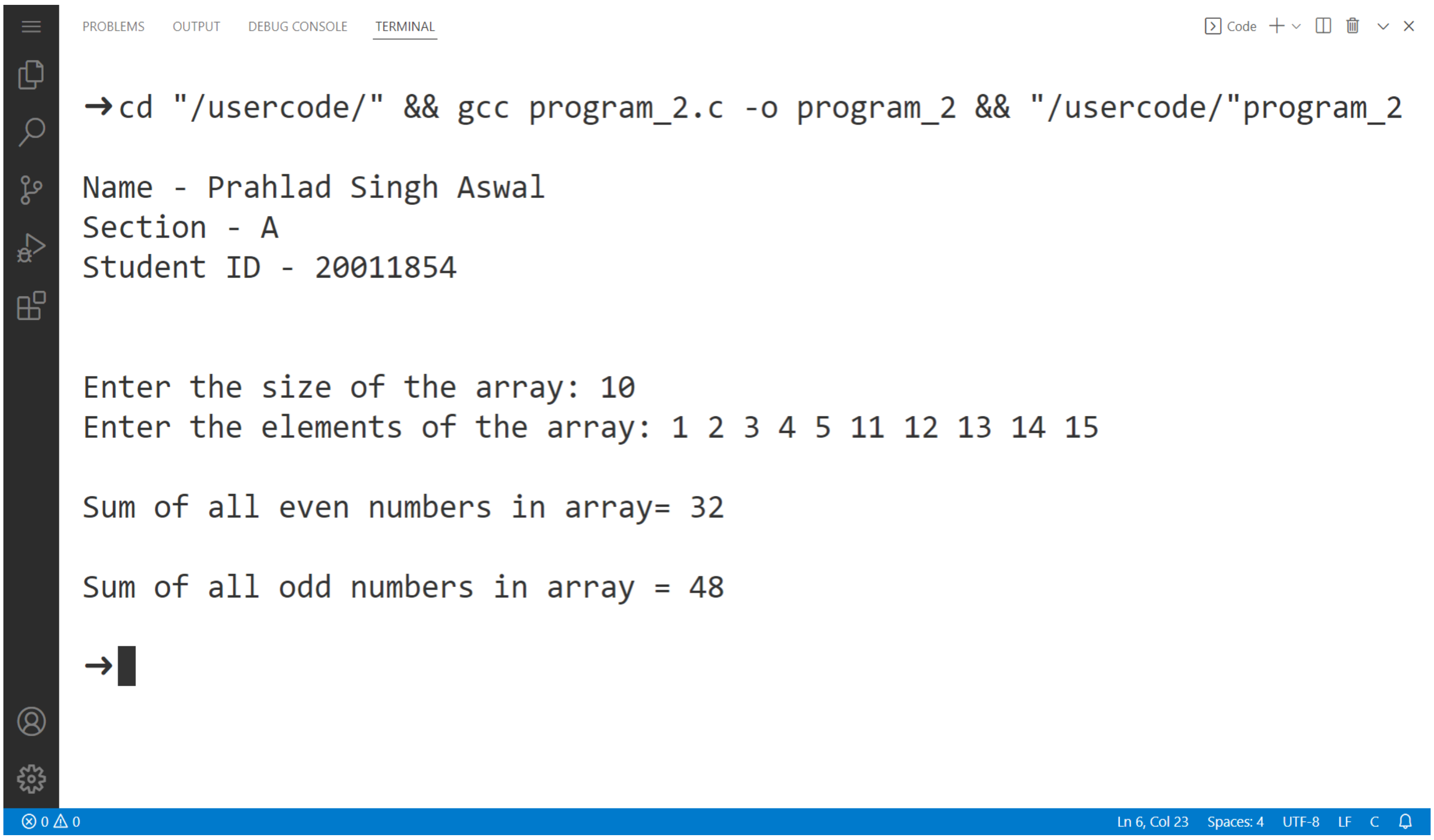
```
#include <stdio.h>
#include <unistd.h>

int main()
{
   printf("\nName - Prahlad Singh Aswal\nSection - A\n");
   printf("Student ID - 20011854\n\n");

   int n;
   int e_sum = 0, o_sum = 0;

   printf("\nEnter the size of the array: ");
   scanf("%d", &n);

   int arr[n];

   printf("Enter the elements of the array: ");
   for (int i = 0; i < n; i++)
      scanf("%d", &arr[i]);

   int pid = fork();
```

```
    int i = 0;
    if (pid == 0)
    {

       while (i < n)
        {
            if (arr[i] % 2 != 0)
                o_sum += arr[i];
            i++;
        }

        printf("\nSum of all odd numbers in array = %d\n\n", o_sum);
    }
    else
    {
        while (i < n)
        {
            if (arr[i] % 2 == 0)
                e_sum += arr[i];
            i++;
        }
        printf("\nSum of all even numbers in array= %d\n", e_sum);
    }
    return 0;
}
```

Output



```
➜cd "/usercode/" && gcc program_2.c -o program_2 && "/usercode/"program_2

Name - Prahlad Singh Aswal
Section - A
Student ID - 20011854


Enter the size of the array: 10
Enter the elements of the array: 1 2 3 4 5 6 7 8 9 10

Sum of all even numbers in array= 30

Sum of all odd numbers in array = 25

➜
```



```
➜cd "/usercode/" && gcc program_2.c -o program_2 && "/usercode/"program_2

Name - Prahlad Singh Aswal
Section - A
Student ID - 20011854


Enter the size of the array: 10
Enter the elements of the array: 1 2 3 4 5 11 12 13 14 15

Sum of all even numbers in array= 32

Sum of all odd numbers in array = 48

➜
```

PRACTICAL 3

Question: Write a C program to demonstrate Zombie Process.

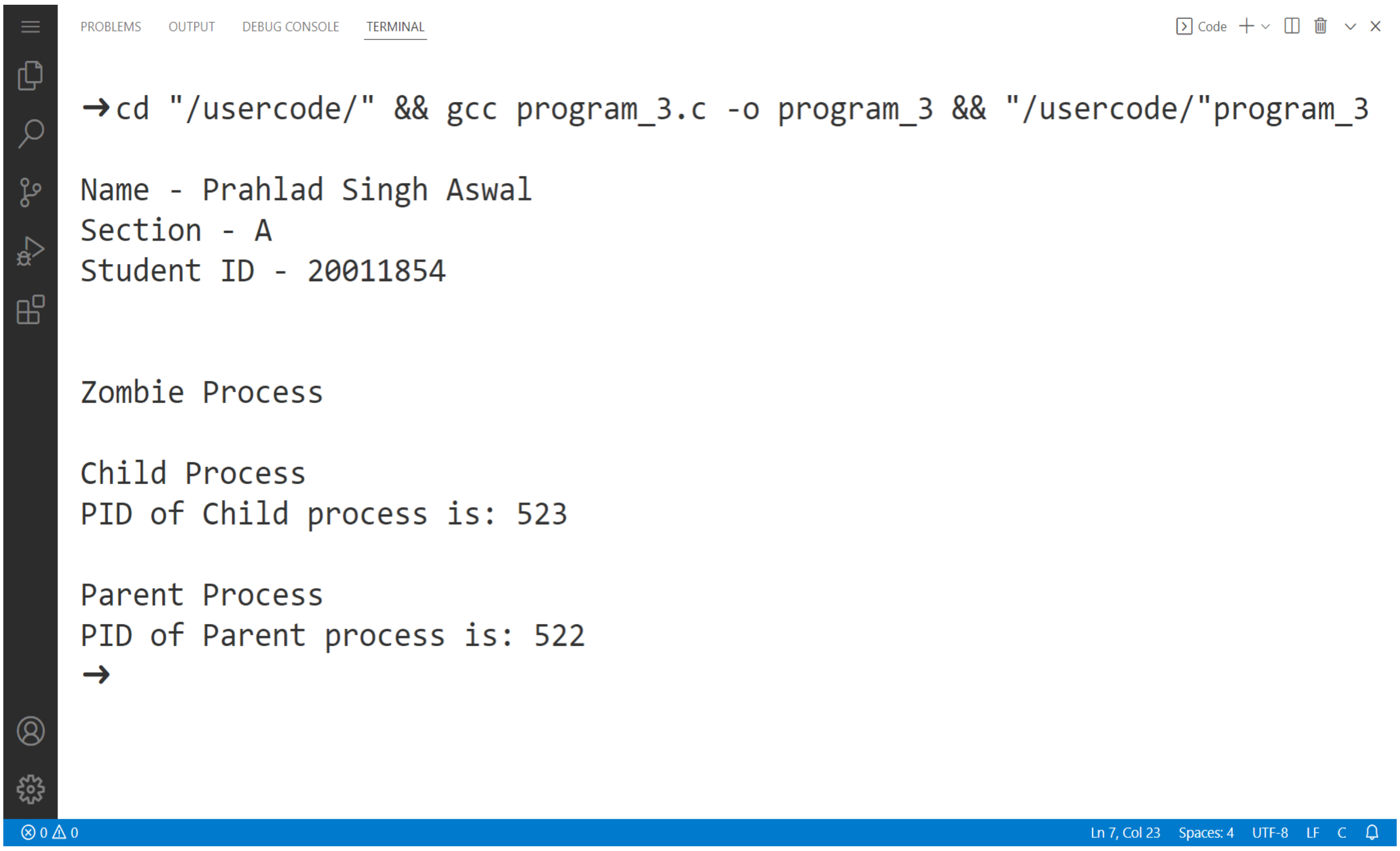
About Zombie Process:

A zombie process is a process in its terminated state. This usually happens in a program that has parent-child functions. After a child function has finished execution, it sends an exit status to its parent function.

Source Code:-

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
int main()
{
   printf("\nName - Prahlad Singh Aswal\nSection - A\n");
   printf("Student ID - 20011854\n\n");

   printf("\nZombie Process\n");
   int pid = fork();

   if (pid > 0)
   {
      sleep(10);
      printf("\nParent Process\n");
      printf("PID of Parent process is: %d\n", getpid());
   }
   else
   {
      printf("\nChild Process\n");
      printf("PID of Child process is: %d\n", getpid());
      exit(0);
   }
   return 0;
}
```

Output



```
➜cd "/usercode/" && gcc program_3.c -o program_3 && "/usercode/"program_3

Name - Prahlad Singh Aswal
Section - A
Student ID - 20011854


Zombie Process

Child Process
PID of Child process is: 523

Parent Process
PID of Parent process is: 522
➜
```

PRACTICAL 4

Question: Write a C program to demonstrate Orphan Process.

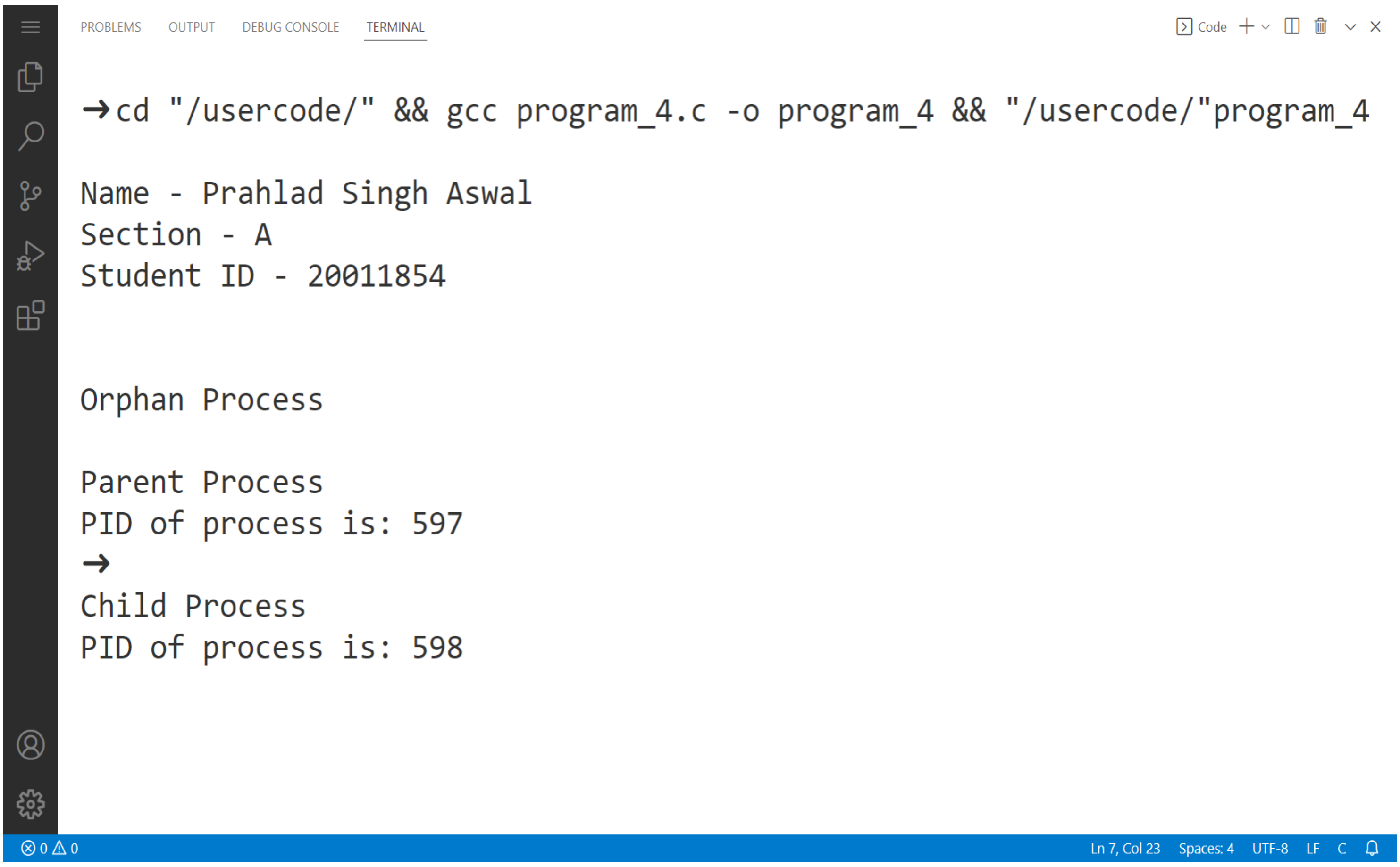
About Orphan Process:

A process whose parent process no more exists i.e. either finished or terminated without waiting for its child process to terminate is called an orphan process. Orphan process can be orphaned intentionally or unintentionally.

Source Code:-

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
int main()
{
   printf("\nName - Prahlad Singh Aswal\nSection - A\n");
   printf("Student ID - 20011854\n\n");

   printf("\nOrphan Process\n");
   int pid = fork();
   if (pid > 0)
   {
      printf("\nParent Process\n");
      printf("PID of process is: %d\n", getpid());
      exit(0);
   }
   else
   {
      sleep(10);
      printf("\nChild Process\n");
      printf("PID of process is: %d\n", getpid());
   }
   return 0;
}
```

Output



```
➜cd "/usercode/" && gcc program_4.c -o program_4 && "/usercode/"program_4

Name - Prahlad Singh Aswal
Section - A
Student ID - 20011854


Orphan Process

Parent Process
PID of process is: 597
➜
Child Process
PID of process is: 598
```