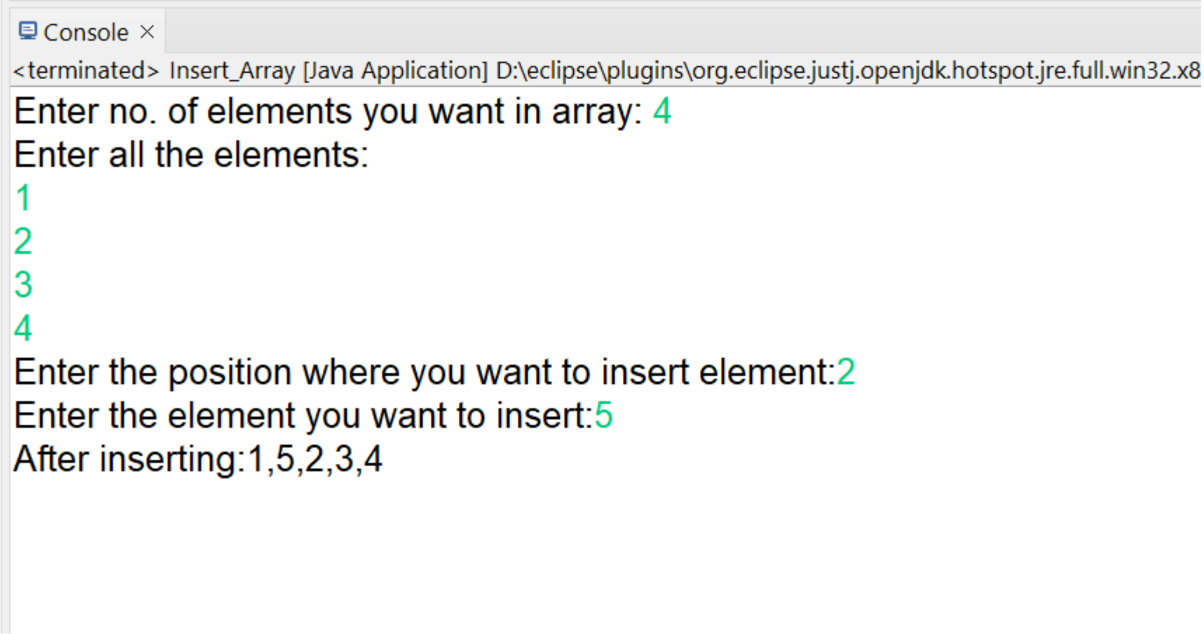
COP&OS Lab exam

Dive Jagdishchandra Anantrao

220960920025

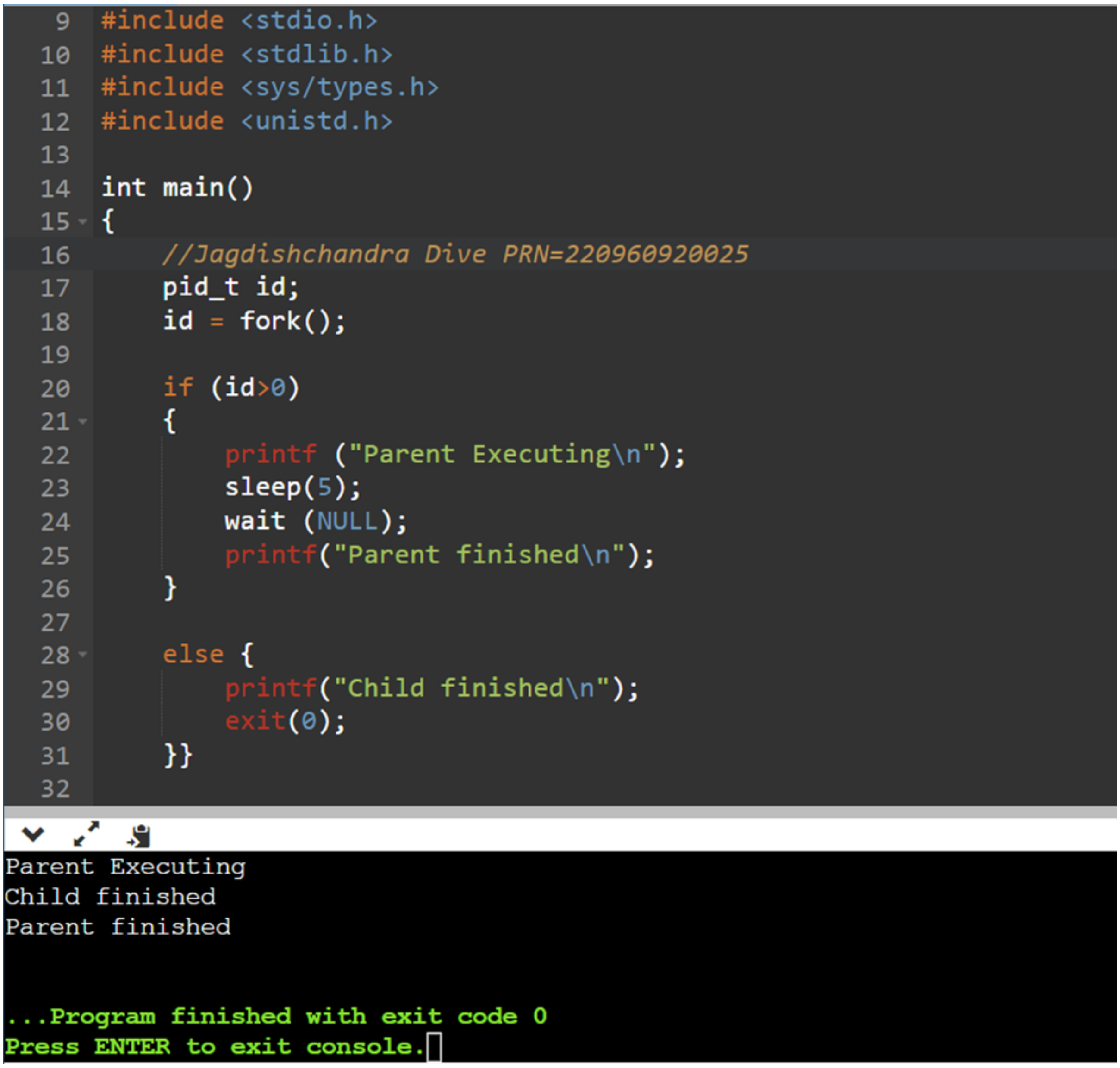
Q1. Write a program to insert an element at a specified position into an array, without the loss of any values. The program should not throw exception, if the array is currently full. Instead, it should omit the last element of the array and shift all numbers to insert the element in the specified position.

```
package com.copExam;
import java.util.Scanner;
public class Insert_Array
{
         public static void main(String[ ] args) {
        int n, pos, x;                                   //jagdishchandra dive
PRN=220960920025
        Scanner s = new Scanner(System.in);
        System.out.print("Enter no. of elements you want in array:");
        n = s.nextInt();
        int a[] = new int[n+1];
        System.out.println("Enter all the elements:");
        for(int i = 0; i < n; i++)
        {
            a[i] = s.nextInt();
        }
        System.out.print("Enter the position where you want to insert
element:");
        pos = s.nextInt();
        System.out.print("Enter the element you want to insert:");
        x = s.nextInt();
        for(int i = (n-1); i >= (pos-1); i--)
        {
            a[i+1] = a[i];
        }
       a[pos-1] = x;
        System.out.print("After inserting:");
        for(int i = 0; i < n; i++)
        {
            System.out.print(a[i]+",");
        }
        System.out.print(a[n]);
}}
```



```
<terminated> Insert_Array [Java Application] D:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x8
Enter no. of elements you want in array: 4
Enter all the elements:
1
2
3
4
Enter the position where you want to insert element:2
Enter the element you want to insert:5
After inserting:1,5,2,3,4
```

Q2. Write a C program to create a Zombie process.



```c
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>

int main()
{
    //Jagdishchandra Dive PRN=220960920025
    pid_t id;
    id = fork();

    if (id>0)
    {
        printf ("Parent Executing\n");
        sleep(5);
        wait (NULL);
        printf("Parent finished\n");
    }

    else {
        printf("Child finished\n");
        exit(0);
    }}

```

```
Parent Executing
Child finished
Parent finished


...Program finished with exit code 0
Press ENTER to exit console.
```

Q2.