Programming Assignments

Q1. Write a program in C that creates a child process, waits for the termination of the child and lists its PID, together with the state inwhich the process was terminated (in decimal and hexadecimal)

Q2. In a C program, print the address of the variable and enter into along loop (say using while(1)).

•Start three to four processes of the same program and observethe printed address values.

•Show how two processes which are members of the relationshipparent-child are concurrent from execution point of view, initiallythe child is copy of the parent, but every process has its own data

Q3. Test the source code below:

```
for (i= 1;i≤10;i+ +)
{
        fork();
        printf("The process with the PID=%d",getpid());
}
```

In the next phase, modify the code, such as after all created pro-cesses have finished execution, in a fileprocessmanagement.txtthetotal number of created processes should be stored.

Q4. Write two programs file1.c and file2.c

Program file1.c uses these :

(a) fork() to launch another process
(b) exec() to replace the program driving this process, while supplying arguments to file2.c to complete its execution
(c) wait() to complete the execution of the child process
(d) file1.c takes two arguments x(a number less than 1) and n (number of terms to be added, 1 or more). For example: file1 0.5 5
(e) When the child process finishes, the parent prints:
Parent(PID=yyy) : Done

Program file2.c requires two arguments to obtain the approximate value of e^x by adding the first n terms in the relation : $e^x = 1+x+x^2/2!+x^3/3!+.....$ and prints the result in the format:

Child(PID=yyy) : For x = 0.5 the first 5 terms yields 1.6484375

Hint : Child-specific processing immediately following the fork() command should load file2.c into the newly created process using the exec() command. This exec() command should also pass 2 arguments to the child. Refer to the man page of exec() command to know how to pass on arguments to the child process. Parent-specific processing should ensure that the parent will wait() for the child- specific processing to complete