REPORT

DESIGN:

The code starts with taking the input from the command line, then copies it into a variable ‘n’. Here, it also checks whether the input is positive or not and whether the command line input is in the incorrect format.

After that, we use sys-call ‘fork ()’, which creates a new process and returns the fork value of the process, which we store in data type ‘pid_t’. the new child process carries a copy of the data of the parent process.

If we get the PID value negative that means forking is failed and was not able to create a new process.

If the PID value is positive, then it is a parent process. The execution goes to the else block.

If the PID value is zero, then it is a child process. The execution goes to else if block.

After forking, the child process from parent process is created, where each goes in their separate ways. The child process communicates with the parent process when completed. So, that is the reason, the parent process needs to wait but not terminate and because child processes are halted when the parent process terminates.

Here, if the PID value is positive parent process is executed. In this, the parent process waits for the child process to complete which was created by forking. While the parent process is waiting, we are doing operations on the copy of the input integer in the child process. Here, the PID value is zero and gives the sequence of collatz numbers as output with the input copy. After this child process is completed, the execution returns to the waiting parent process. Now parent process completes and ends its execution.

OUTPUT ANALYSIS:

```
1>>Input number is: 35
2>>Parent Process...
3>>Child process...
4>>35, 106, 53, 160, 80, 40, 20, 10, 5, 16, 8, 4, 2, 1
5>>...Child Process Complete
6>>The value of n: 35
7>>...Parent Process Complete
```

This is an output for integer 35.

In the 1st line, we are printing what is the input integer, then we are doing the forking.

In the 2nd line, after forking, the PID value is positive. So, execution of the parent process starts, then starts waiting for the child process to complete.

In the 3rd line, the child process starts.

In the 4th line, the collatz sequence is printed by doing the operation on input copy.

In the 5th line, the child process ends, and execution returns to the parent process.

In the 6th line, after returning to parent process from child process ends, we are printing the input value to show that parent and child process have separate data copies and go in their ways separately.

In the 7th line, the parent process ends, and the program terminates.