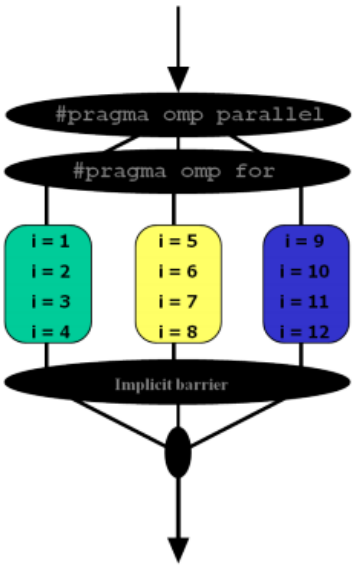
Dated :
Assessment No. : 3

SCENARIO – I

Write a simple OpenMP program to employ a *'Work Sharing'* clause to assign each thread an independent set of iterations. In order to explore its practical use, you are advised to read and understand the following statements.

1. Assign each thread an independent set of iterations;
2. Threads must wait at the end
3. Can combine the directives:
4. #pragma omp parallel for
5. Only simple kinds of for loops:
    a. Only one signed integer variable
    b. Initialization: var=init
    c. Comparison: var op last op: , <=, >=
    d. Increment: var++, var--, var+=incr, var-=incr, etc.

Execution Scenario:



BRIEF ABOUT YOUR APPROACH:

SOURCE CODE:

EXECUTION:

RESULTS:

Dated :
Assessment No. : 3

SCENARIO – II

Write an OpenMP program to specify that the enclosed section(s) of code are to be divided among the threads using OpenMP SECTION clause.

Description

Independent SECTION directives are nested within a SECTIONS directive. Each SECTION is executed once by a thread in the team. Different sections may be executed by different threads. It is possible that for a thread to execute more than one section if it is quick enough and the implementation permits such.

Execution Scenario

```
answer1 = long_computation_1();
answer2 = long_computation_2();
if (answer1 != answer2) { ... }
```

How to parallelize? These are just two independent computations!

```
#pragma omp sections
{
  #pragma omp section
   answer1 = long_computation_1();
  #pragma omp section
   answer2 = long_computation_2();
}
if (answer1 != answer2) { ... }
```

BRIEF ABOUT YOUR APPROACH:

SOURCE CODE:

EXECUTION:

RESULTS: