NAME: MOUNVI PODAPATI
REG. NO: 19BCE0396
SLOT: L15 + L16
FACULTY: PROF. DEEBAK B.D
DATED: 1ST SEPTEMBER 2021

LAB ASSESSMENT – 3

AIM:

Write a simple Open MP program to demonstrate the use of pattern generation in schedule clause

```
* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *
```

a. Statically assign the loop iterations to threads

b. Dynamically assign one iteration to each threads

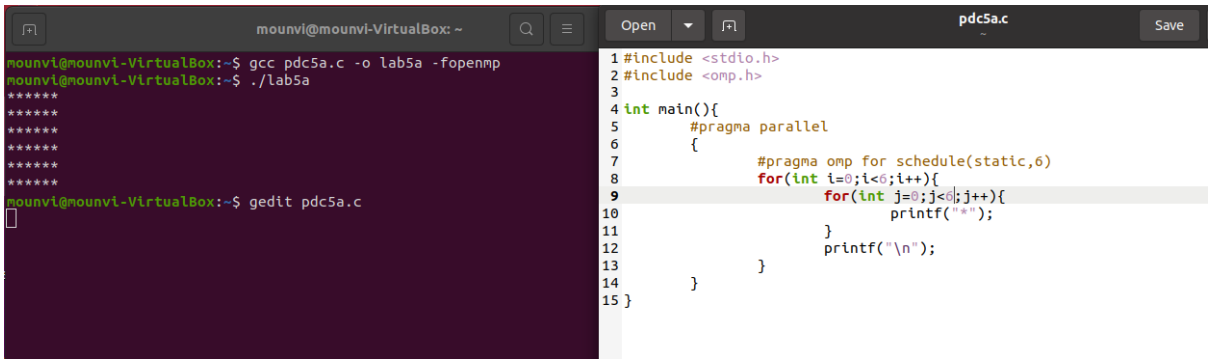
(a)

SOURCE CODE:

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

int main(){
	#pragma parallel
	{
		#pragma omp for schedule(static,1)
		for(int i=0;i<6;i++){
			for(int j=0;j<6;j++){
				printf("*");
			}
			printf("\n");
		}
	}
}
```

EXECUTION:



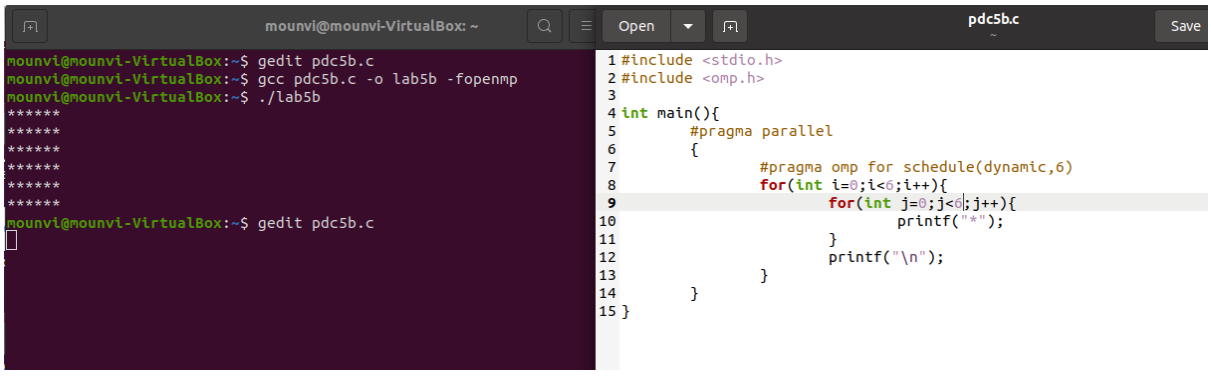
(b)

SOURCE CODE:

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

int main(){
        #pragma parallel
        {
                #pragma omp for schedule(dynamic,6)
                for(int i=0;i<6;i++){
                        for(int j=0;j<6;j++){
                                printf("*");
                        }
                        printf("\n");
                }
        }
}
```

EXECUTION:



RESULTS:

From this experiment, I was able to understand how threads are scheduled in OpenMP. There were two type of threads which were used in the experiment – *static* and *dynamic.*

(a) In case of static scheduling, each thread is assigned chunk of iterations in round robin format. All iterations are equally divided among the threads. The integer specified as the 2nd argument will allocate chunk number of contiguous iterations to a particular thread.

(b) In case of dynamic scheduling, each thread is assigned with a chunk of threads, then when each thread completes its iterations, it is assigned with the next set of iterations. The integer specified as the 2nd argument will allocate chunk number of contiguous iterations that are allocated to a thread at a time.