

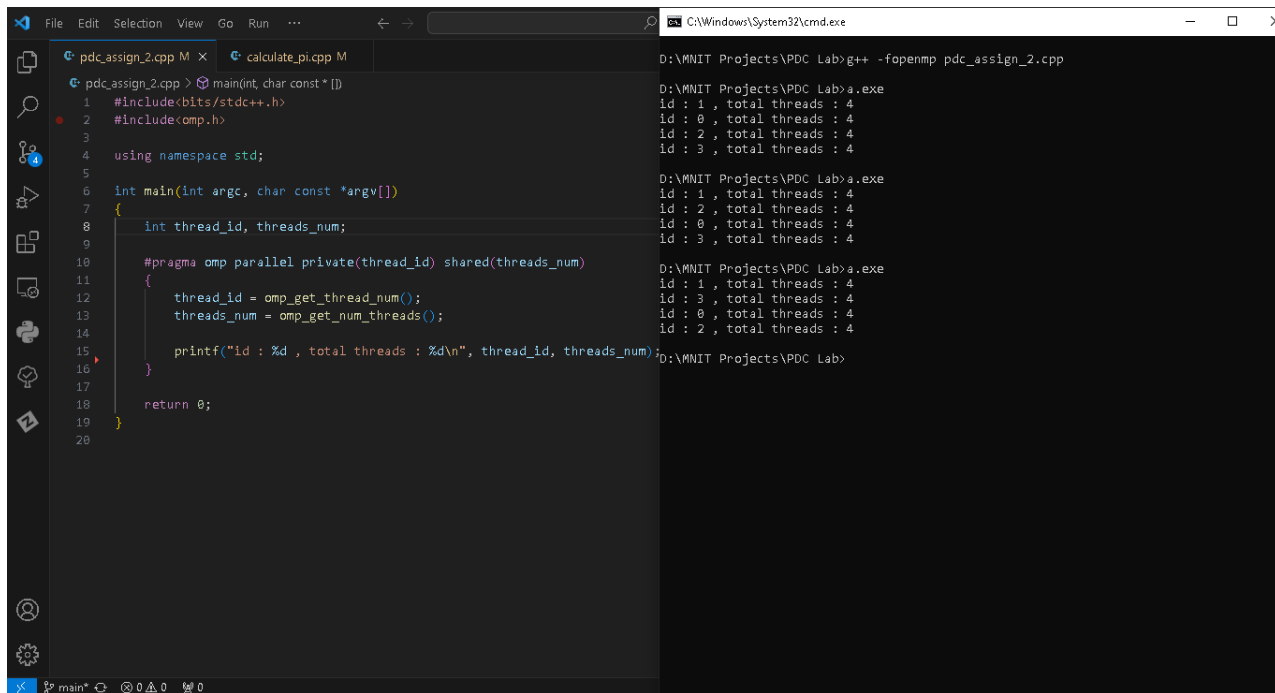
Name: Ranjan Baro

ID: 2023PCP5274

### Exercise 1: OpenMP Hello-world Program (Implement the first OpenMP program).

- Find out the default number of threads of your system.
- Observer the behaviour of the threads. Is there any possibility that two threads can interleave the printing of “hello world”.

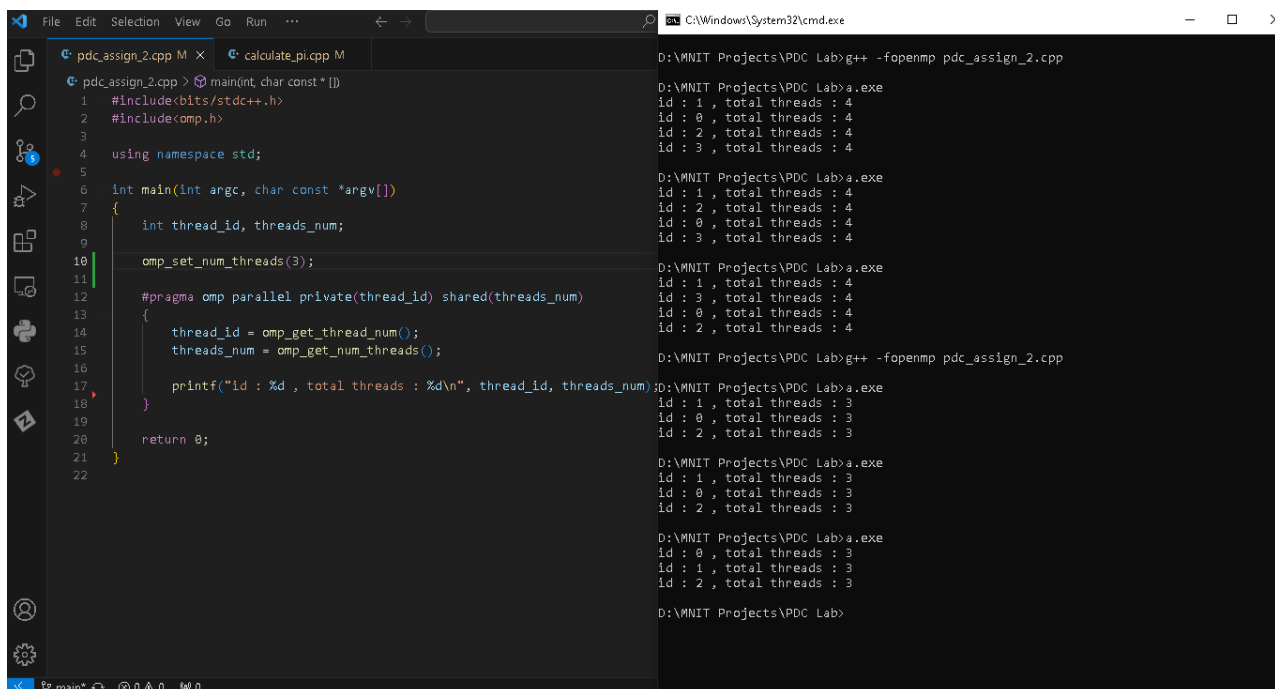
*Ans.* OpenMP Hello-world Program is implemented below:



```
File Edit Selection View Go Run ... C:\Windows\System32\cmd.exe
pdc_assign_2.cpp M X calculate_pi.cpp M
pdc_assign_2.cpp > main(int, char const* [])
1 #include<bits/stdc++.h>
2 #include<omp.h>
3
4 using namespace std;
5
6 int main(int argc, char const *argv[])
7 {
8     int thread_id, threads_num;
9
10    #pragma omp parallel private(thread_id) shared(threads_num)
11    {
12        thread_id = omp_get_thread_num();
13        threads_num = omp_get_num_threads();
14
15        printf("id : %d , total threads : %d\n", thread_id, threads_num);
16    }
17
18    return 0;
19 }
20

D:\MNIT Projects\PCD Lab>g++ -fopenmp pdc_assign_2.cpp
D:\MNIT Projects\PCD Lab>a.exe
id : 1 , total threads : 4
id : 0 , total threads : 4
id : 2 , total threads : 4
id : 3 , total threads : 4
D:\MNIT Projects\PCD Lab>a.exe
id : 1 , total threads : 4
id : 2 , total threads : 4
id : 0 , total threads : 4
id : 3 , total threads : 4
D:\MNIT Projects\PCD Lab>a.exe
id : 1 , total threads : 4
id : 3 , total threads : 4
id : 0 , total threads : 4
id : 2 , total threads : 4
D:\MNIT Projects\PCD Lab>
```

- The default number of threads of my system is 4 (four).
- We can set the number of OpenMP threads using **omp\_set\_num\_threads ( )** function and passing desired number of threads as an argument into the function.



```
File Edit Selection View Go Run ... C:\Windows\System32\cmd.exe
pdc_assign_2.cpp M X calculate_pi.cpp M
pdc_assign_2.cpp > main(int, char const* [])
1 #include<bits/stdc++.h>
2 #include<omp.h>
3
4 using namespace std;
5
6 int main(int argc, char const *argv[])
7 {
8     int thread_id, threads_num;
9
10    omp_set_num_threads(3);
11
12    #pragma omp parallel private(thread_id) shared(threads_num)
13    {
14        thread_id = omp_get_thread_num();
15        threads_num = omp_get_num_threads();
16
17        printf("id : %d , total threads : %d\n", thread_id, threads_num);
18    }
19
20    return 0;
21 }
22

D:\MNIT Projects\PCD Lab>g++ -fopenmp pdc_assign_2.cpp
D:\MNIT Projects\PCD Lab>a.exe
id : 1 , total threads : 4
id : 0 , total threads : 4
id : 2 , total threads : 4
id : 3 , total threads : 4
D:\MNIT Projects\PCD Lab>a.exe
id : 1 , total threads : 4
id : 2 , total threads : 4
id : 0 , total threads : 4
id : 3 , total threads : 4
D:\MNIT Projects\PCD Lab>a.exe
id : 1 , total threads : 4
id : 3 , total threads : 4
id : 0 , total threads : 4
id : 2 , total threads : 4
D:\MNIT Projects\PCD Lab>g++ -fopenmp pdc_assign_2.cpp
D:\MNIT Projects\PCD Lab>a.exe
id : 1 , total threads : 3
id : 0 , total threads : 3
id : 2 , total threads : 3
D:\MNIT Projects\PCD Lab>a.exe
id : 1 , total threads : 3
id : 0 , total threads : 3
id : 2 , total threads : 3
D:\MNIT Projects\PCD Lab>a.exe
id : 0 , total threads : 3
id : 1 , total threads : 3
id : 2 , total threads : 3
D:\MNIT Projects\PCD Lab>
```

- Yes, there is a possibility that two threads can interleave the printing of “hello world”. Because there is no guarantee about which thread executes first, which executes second.

**Ans.**

- The execution time decreases as the number of threads increases. But after a certain number of threads, we see increase in execution time, this is due to overhead from creation of thread and synchronization. Optimum number of threads that will give best performance depends on the number of available CPU cores and system configuration.