Walchand College of Engineering, Sangli
Department of Computer Science and Engineering

Class: Final Year (Computer Science and Engineering)

Year: 2021-22 **Semester:** 1

Course: High Performance Computing lab

ESE Exam

23/11/2021

01.00 PM – 04.00 PM

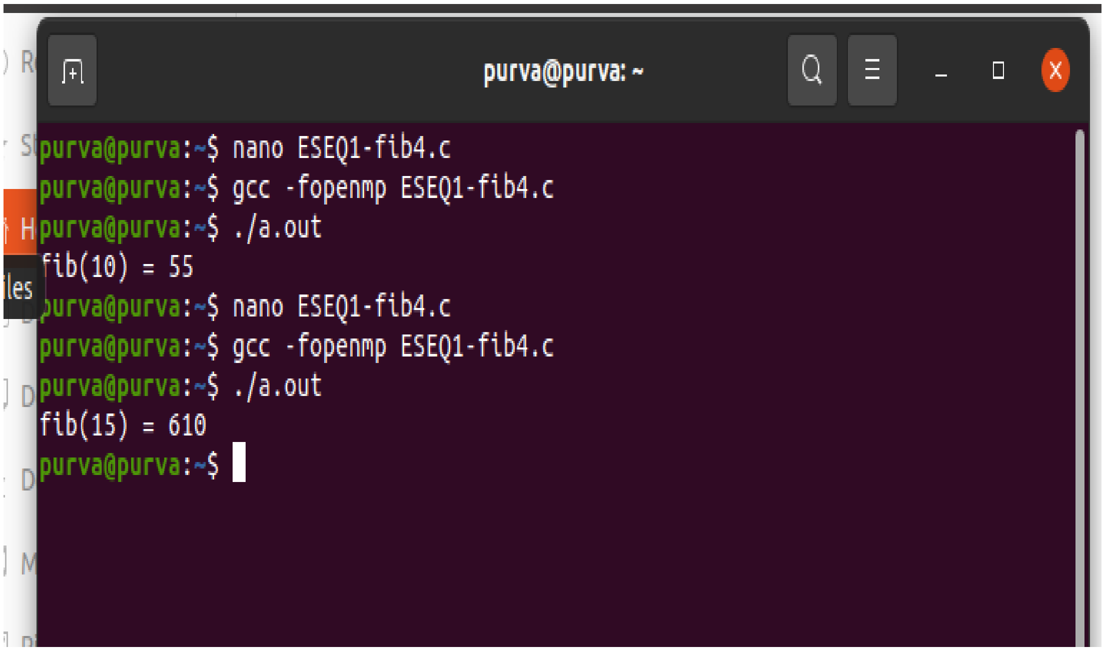
Exam Seat No:

Name: Purva Kudre
Exam Seat Number: 2018BTECS00095

Problem Statement 1

Statement: Implement Fibonacci Series using OpenMP.

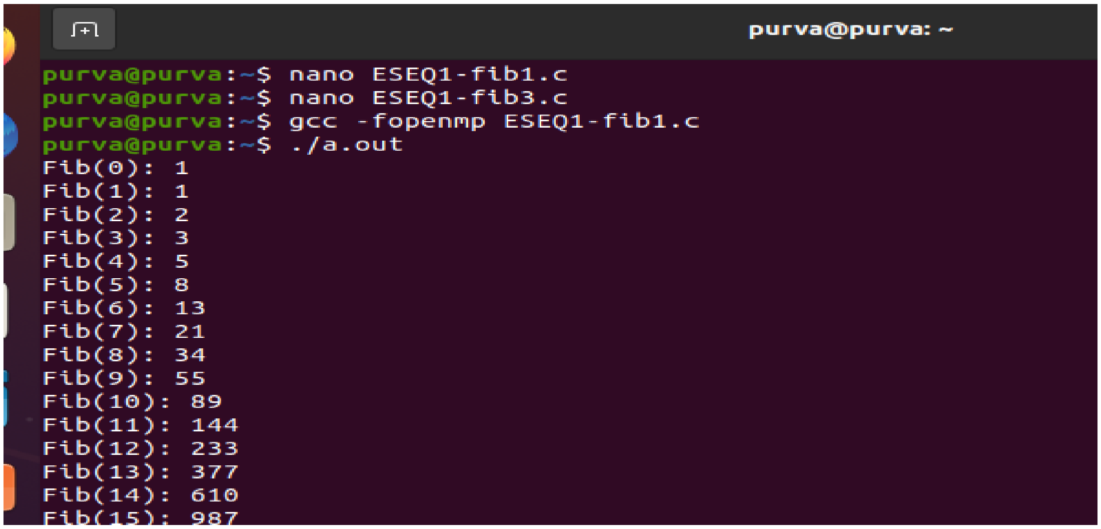
Screenshot 1.1:



```
purva@purva:~$ nano ESEQ1-fib4.c
purva@purva:~$ gcc -fopenmp ESEQ1-fib4.c
purva@purva:~$ ./a.out
fib(10) = 55
purva@purva:~$ nano ESEQ1-fib4.c
purva@purva:~$ gcc -fopenmp ESEQ1-fib4.c
purva@purva:~$ ./a.out
fib(15) = 610
purva@purva:~$
```

Information 1.1: Above is the screenshot of a Fibonacci Series Program implemented using OpenMP.

Screenshot 1.2:



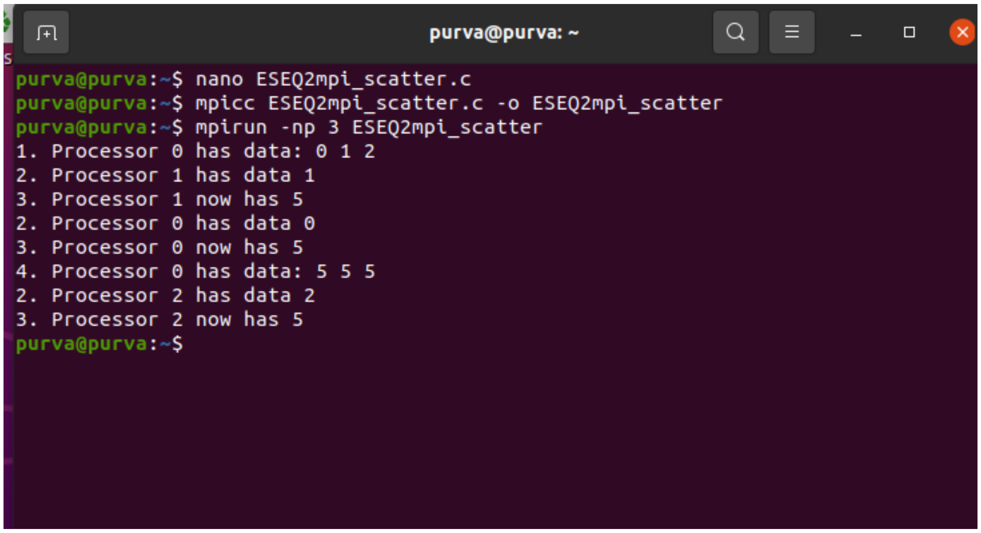
```
purva@purva:~$ nano ESEQ1-fib1.c
purva@purva:~$ nano ESEQ1-fib3.c
purva@purva:~$ gcc -fopenmp ESEQ1-fib1.c
purva@purva:~$ ./a.out
Fib(0): 1
Fib(1): 1
Fib(2): 2
Fib(3): 3
Fib(4): 5
Fib(5): 8
Fib(6): 13
Fib(7): 21
Fib(8): 34
Fib(9): 55
Fib(10): 89
Fib(11): 144
Fib(12): 233
Fib(13): 377
Fib(14): 610
Fib(15): 987
```

Information 1.2: Above is the screenshot of a Recursive Fibonacci Series Program in OpenMP.

Problem Statement 2

Statement: Implement MPI program to scatter the data from one process to other process.

Screenshot 2.1:



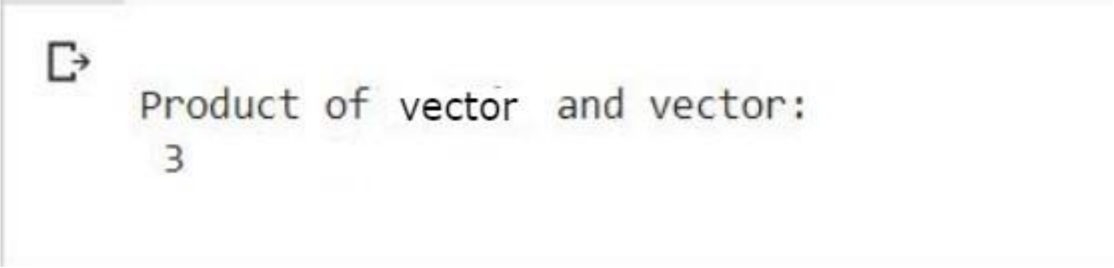
```
purva@purva:~$ nano ESEQ2mpi_scatter.c
purva@purva:~$ mpicc ESEQ2mpi_scatter.c -o ESEQ2mpi_scatter
purva@purva:~$ mpirun -np 3 ESEQ2mpi_scatter
1. Processor 0 has data: 0 1 2
2. Processor 1 has data 1
3. Processor 1 now has 5
2. Processor 0 has data 0
3. Processor 0 now has 5
4. Processor 0 has data: 5 5 5
2. Processor 2 has data 2
3. Processor 2 now has 5
purva@purva:~$
```

Information 2.2: Above is the screenshot of an implementation of an MPI program to scatter data from one process to other process

Problem Statement 3

Statement: Implement Vector-vector multiplication using CUDA.

Screenshot 3:



```
Product of vector and vector:
 3
```

Information 3: Above is the screenshot of the Implementation of Vector-Vector multiplication using CUDA.
The Vectors are <2,1> and <3,-2>

Technologies Used:

1. **Virtual box – Ubuntu 20.04**
2. **Google "Colab"**

GitHub Link: **https://github.com/Psk1999/HPC_LA3**