

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

24/11/2021 01.00 PM – 04.00 PM

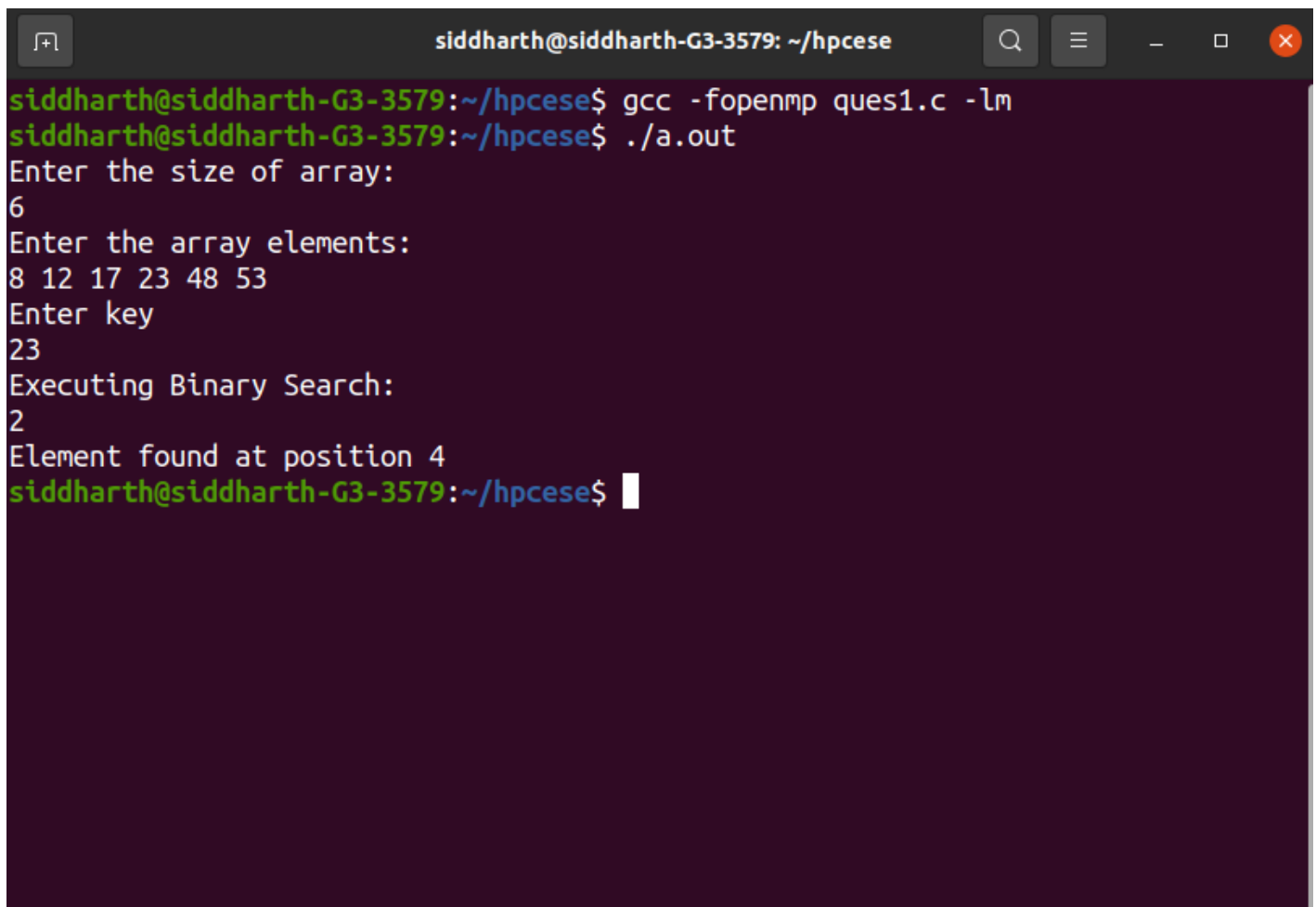
Exam Seat No: 2018BTECS00001

Name: Siddharth Sunil Satpute

Exam Seat Number: 2018BTECS00001

Problem Statement 1 : Implement Binary Search using OpenMP.

Screenshot 1:



```
siddharth@siddharth-G3-3579: ~/hpcese
siddharth@siddharth-G3-3579:~/hpcese$ gcc -fopenmp ques1.c -lm
siddharth@siddharth-G3-3579:~/hpcese$ ./a.out
Enter the size of array:
6
Enter the array elements:
8 12 17 23 48 53
Enter key
23
Executing Binary Search:
2
Element found at position 4
siddharth@siddharth-G3-3579:~/hpcese$
```

Information:

Compilation command: - gcc -fopenmp ques1.c -lm

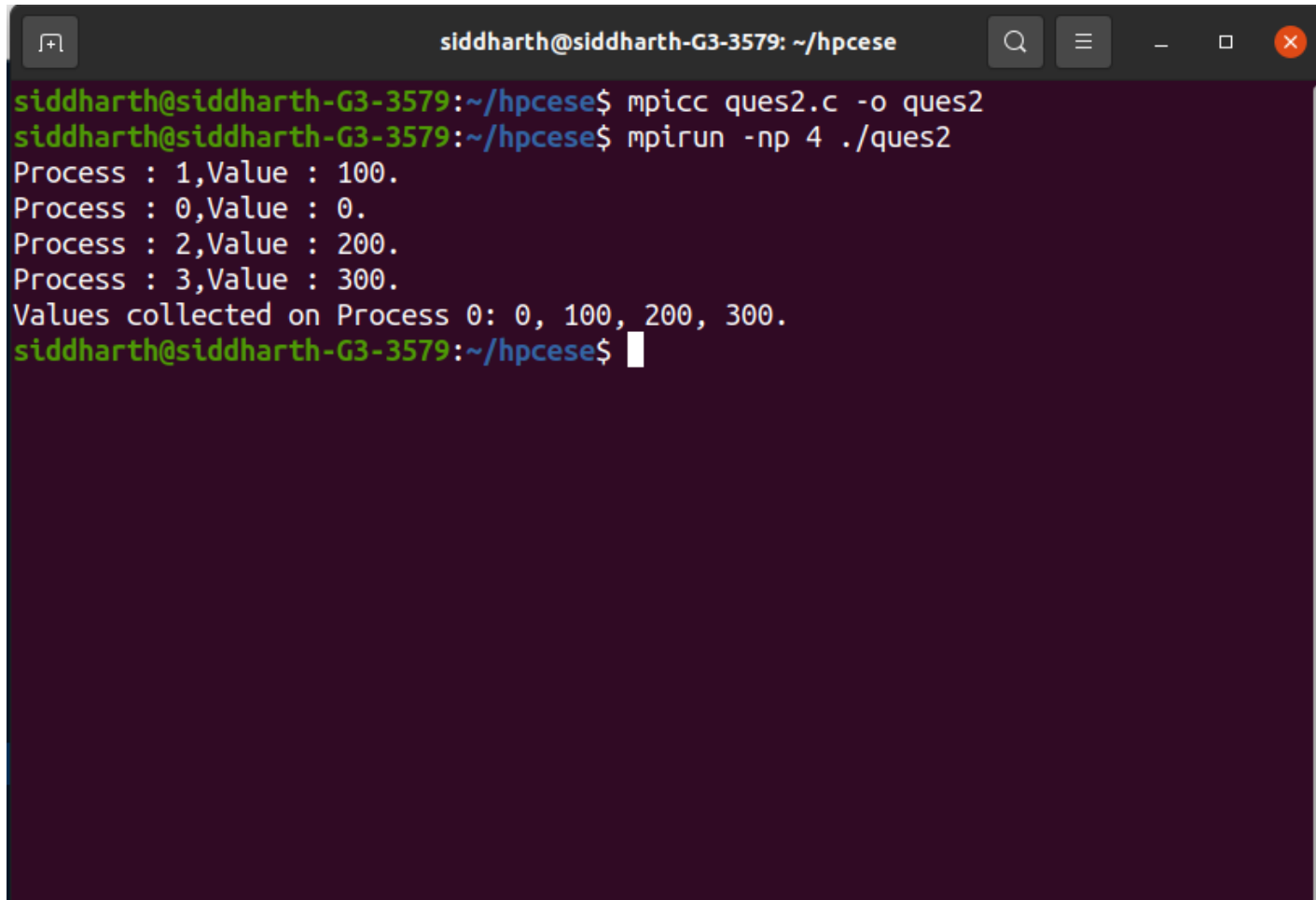
Execution command: - ./a.out

This searches the key element and displays the position at the console.

Problem Statement 2

Statement: Implement MPI program to gather the data from n processes to root process.

Screenshot #1:

A terminal window with a dark background and light-colored text. The window title is 'siddharth@siddharth-G3-3579: ~/hpcese'. The prompt is 'siddharth@siddharth-G3-3579:~/hpcese\$'. The first command is 'mpicc ques2.c -o ques2'. The second command is 'mpirun -np 4 ./ques2'. The output shows four processes reporting their values: Process 1: 100, Process 0: 0, Process 2: 200, Process 3: 300. The final line of output is 'Values collected on Process 0: 0, 100, 200, 300.' followed by a new prompt 'siddharth@siddharth-G3-3579:~/hpcese\$' with a cursor.

```
siddharth@siddharth-G3-3579:~/hpcese$ mpicc ques2.c -o ques2
siddharth@siddharth-G3-3579:~/hpcese$ mpirun -np 4 ./ques2
Process : 1,Value : 100.
Process : 0,Value : 0.
Process : 2,Value : 200.
Process : 3,Value : 300.
Values collected on Process 0: 0, 100, 200, 300.
siddharth@siddharth-G3-3579:~/hpcese$
```

Information:

Compilation command: - mpicc ques2.c -o ques2

Execution command: - mpirun -np 4 ./ques2

Here the MPI code runs with gather clause that collects the values of individual processes in the root process.

Problem Statement 3

Statement:

Implement Matrix scalar addition using CUDA.

Screenshot 1:



```
{
    printf("%d\t",c[i][j]);
}
printf("\n");
}
cudaFree(d);
cudaFree(e);
cudaFree(f);
return 0;
}
```

Matrix 1:

0	1	2
1	2	3

Matrix 2:

0	1	2
2	3	4

Sum = Matrix 1 + Matrix 2 :

0	2	4
3	5	7

Information:

Compiled and ran the code on Google Collab for Matrix Addition.

GitHub Link:

<https://github.com/siddharth810/HPC-ESE>