LAB EXAMINATION

NAME: SOUMYADIP GHOSH

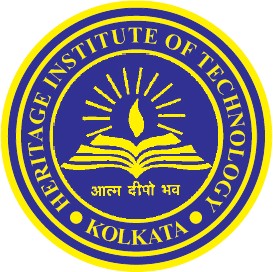
STREAM: CSE-A

ROLL NUMBER: 1951007

AUTONOMY ROLL NUMBER: 12619001158

SUBJECT: OPERATING SYSTEMS LAB

SUBJECT CODE: CSEN 2253

****

**HERITAGE INSTITUTE OF TECHNOLOGY**

Stream : Soumyadip Ghosh

Paper : Operating Systems Lab

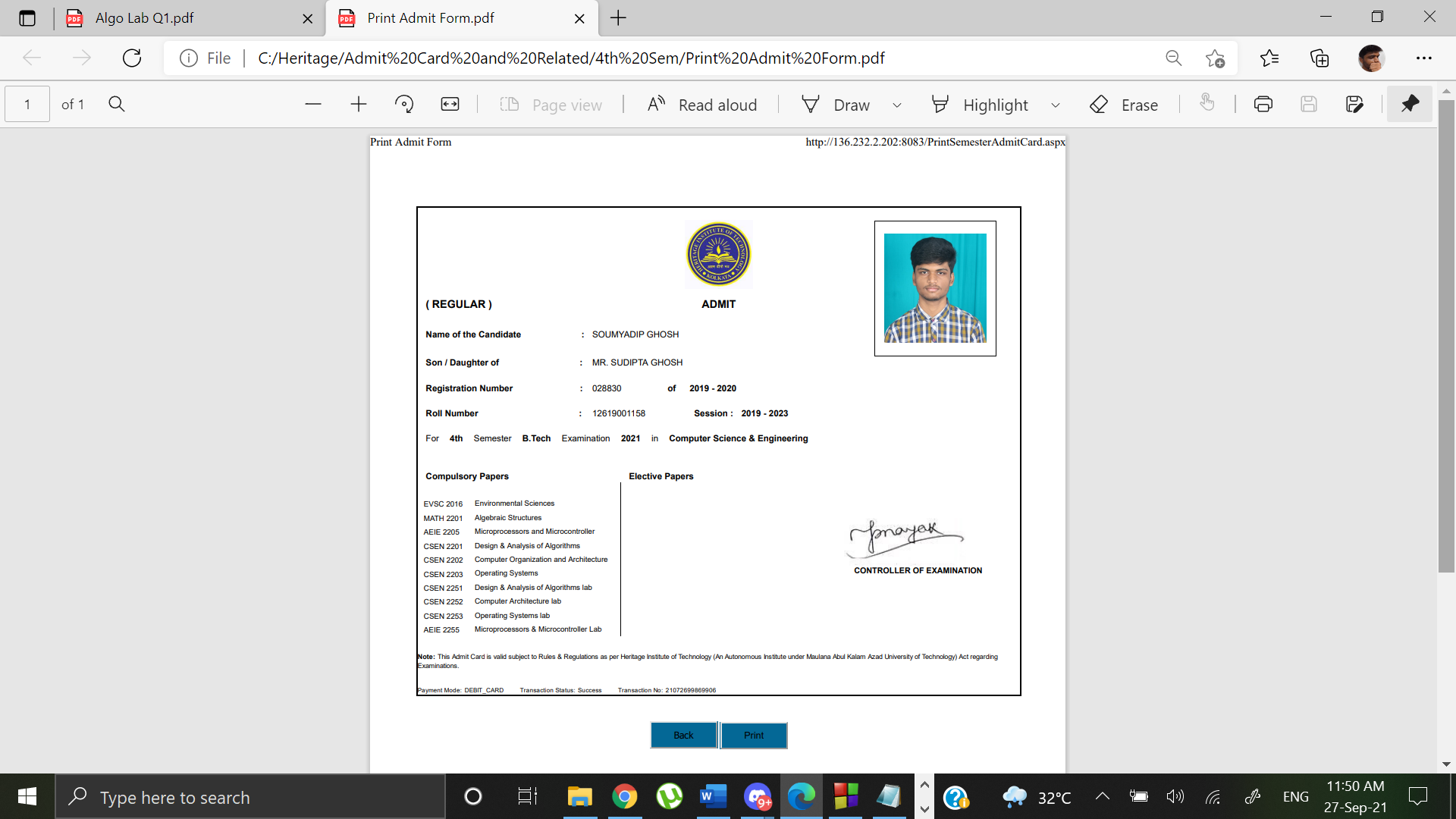
Paper Code : CSEN 2253

Date of Examination : 28.09.21

Full Signature of Student : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Roll No.** | 1 | 2 | 6 | 1 | 9 | 0 | 0 | 1 | 1 | 5 | 8 |  |  |
| **Reg. No.** | 0 | 2 | 8 | 8 | 3 | 0 |  |  |  |  |  |  |  |
| **of** | **2019................................... 2020......................................** | | | | | | | | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Q. No.** | **Marks** | |  | **Q. No.** | **Marks** |
| **1.** |  | | **7.** |  |
| **2.** |  | | **8.** |  |
| **3.** |  | | **9.** |  |
| **4.** |  | | **10.** |  |
| **5.** |  | | **11.** |  |
| **6.** |  | | **12.** |  |
| **Total Marks** | |  | | | |
| **Total in Words** | |  | | | |
| **Signature** | |  | | | |



**Question 1**

Write a shell script to display Fibonacci Series up to Nth term. N must be accepted from the command line as command line input. Program should give an appropriate message in case the number N is negative or 0. For e.g. program1.sh 5

**Code**

if [ "$#" -ne 1 ];then

echo "Number of arguments is not equal to 1"

exit 1

fi

n=$1

if [ $n -lt 1 ];then

echo "The number N must be greater than or equal to 1"

exit 1

fi

t1=0

t2=1

echo "The Fibonacci series:"

for((i=0;i<n;i++))

do

echo "$t1"

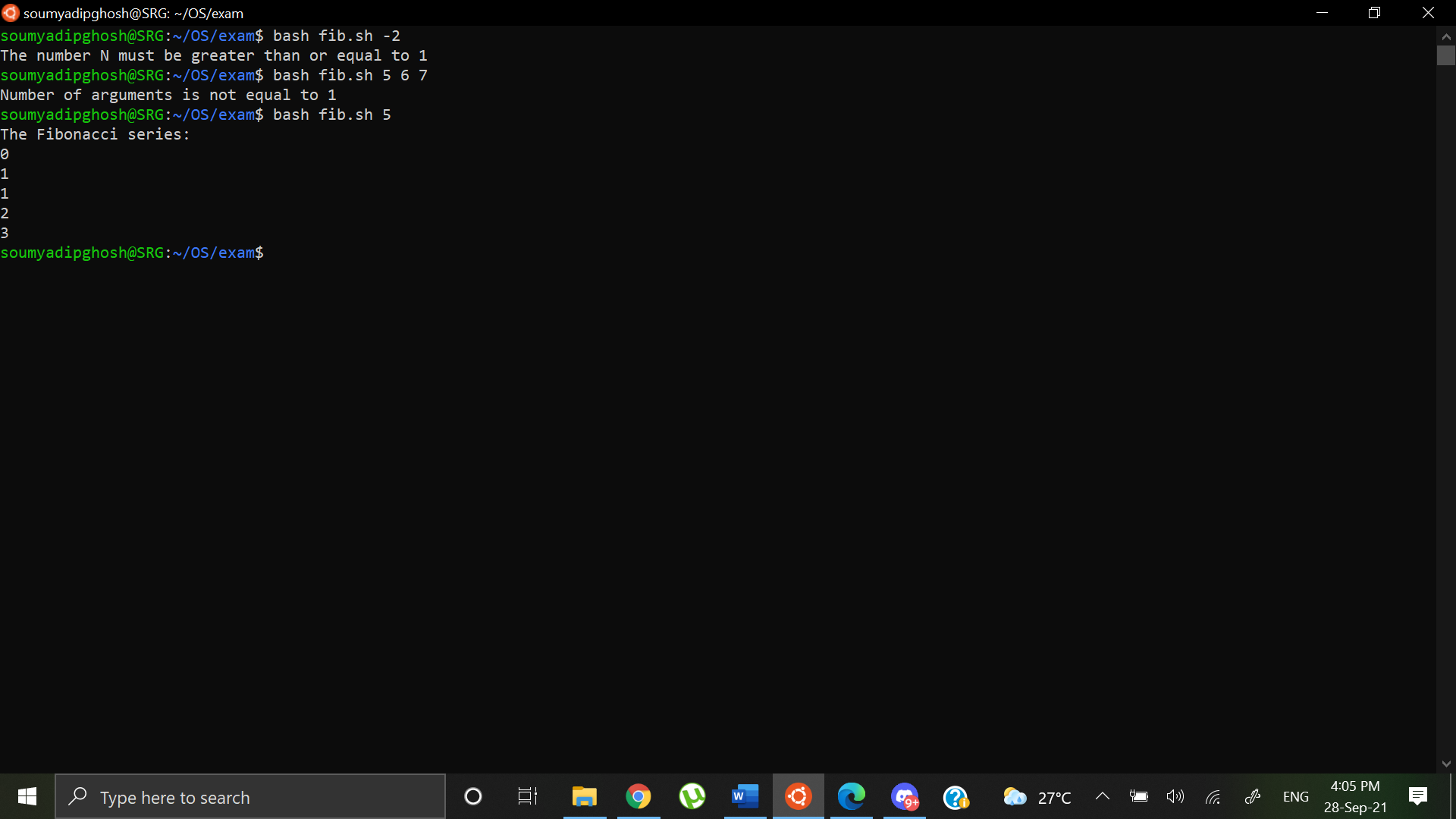
temp=$((t1+t2))

t1=$t2

t2=$temp

done

**Output**



**Question 2**

Write a program that creates exactly three child processes. The first child process should execute a command to show information about the current user, the second child process should execute a command to display the current directory and details of all files within the directory, and the third child process should execute a command to show the current date only. The parent process waits for all the child processes to finish and prints the termination status of the children. Display of information should be in order of first process, followed by second process, and then the third process. Each process should display its own ID, and parent ID and/or child ID.

**Code**

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/wait.h>

#include <sys/types.h>

int main()

{

int pid, pid1, pid2;

int ex1,ex2,ex3;

pid = fork();

if (pid == 0) {

printf("\n1st Child --> Self ID = %d and Parent ID = %d\n",getpid(), getppid());

execlp("whoami","whoami",(char\*)NULL);

exit(1);

}

else {

wait(&ex1);

pid1 = fork();

if (pid1 == 0) {

printf("1st Child has been terminated\n");

printf("\n2nd Child --> Self ID = %d and Parent ID = %d\n",getpid(), getppid());

system("echo `pwd`");

execlp("ls","ls","-l",(char\*)NULL);

exit(2);

}

else {

wait(&ex2);

pid2 = fork();

if (pid2 == 0) {

printf("2nd Child has been terminated\n");

printf("\n3rd Child --> Self ID = %d and Parent ID = %d\n",getpid(), getppid());

execlp("date","date",(char\*)NULL);

exit(3);

}

else {

wait(&ex3);

printf("3rd child has been terminated\n");

printf("\nParent PID = %d\n", getpid());

}

}

}

return 0;

}

**Output**

