PDEU PANDIT DEENDAYAL ENERGY UNIVERSITY

SOT SCHOOL OF TECHNOLOGY

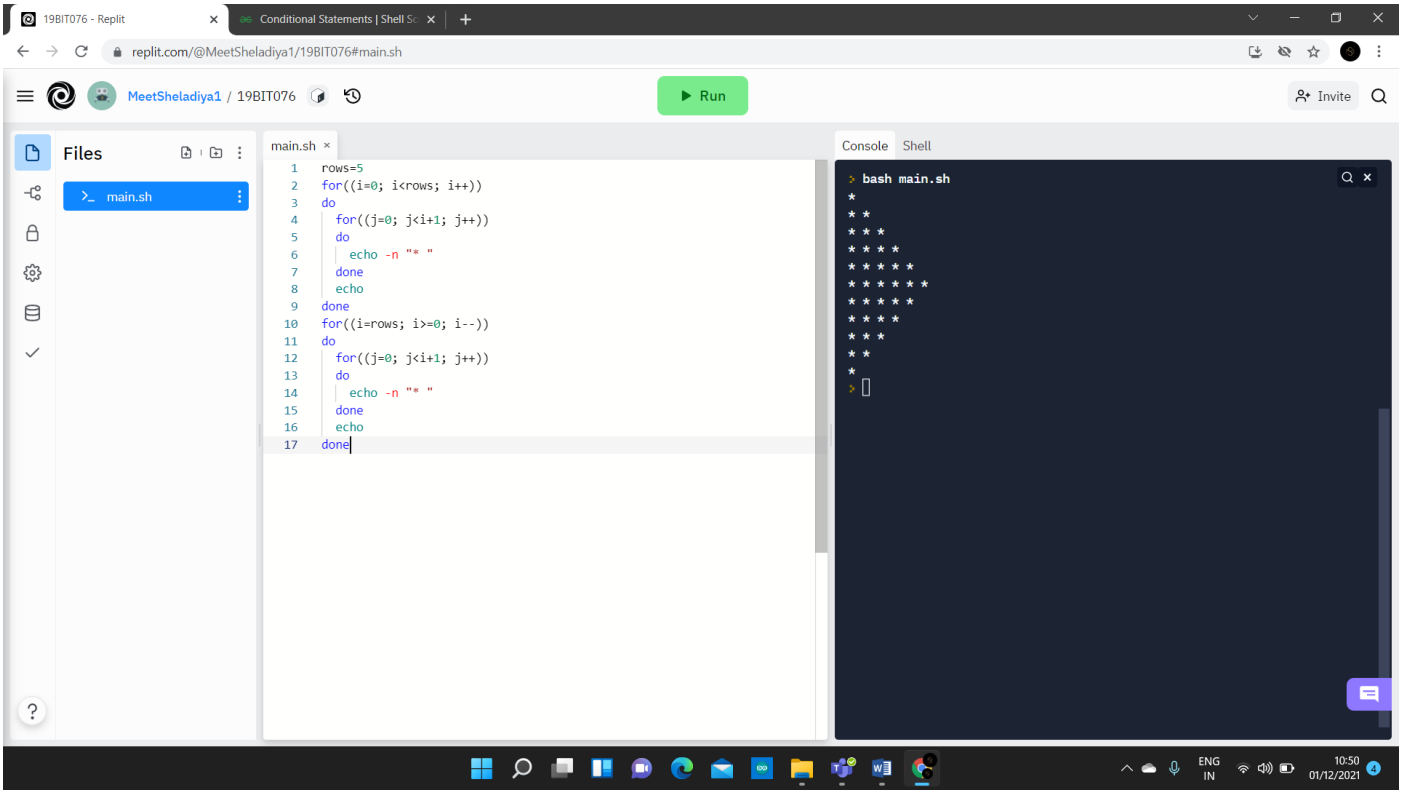
NAME	**:**	**MEET SHELADIYA**
ROLL NUMBER	**:**	**19BIT076**
SUBJECT	**:**	**OPERATING SYSTEM LAB (20IC301P)**
BRANCH	**:**	**INFORMATION AND COMMUNICATION TECHNOLOGY (I.C.T.) DIVISION-2**

1.a) Write a shell script to print the following pattern

Code:

```
rows=5
for((i=0; i<rows; i++))
do
  for((j=0; j<i+1; j++))
  do
    echo -n "* "
  done
  echo
done
for((i=rows; i>=0; i--))
do
  for((j=0; j<i+1; j++))
  do
    echo -n "* "
  done
  echo
done
```

Output:

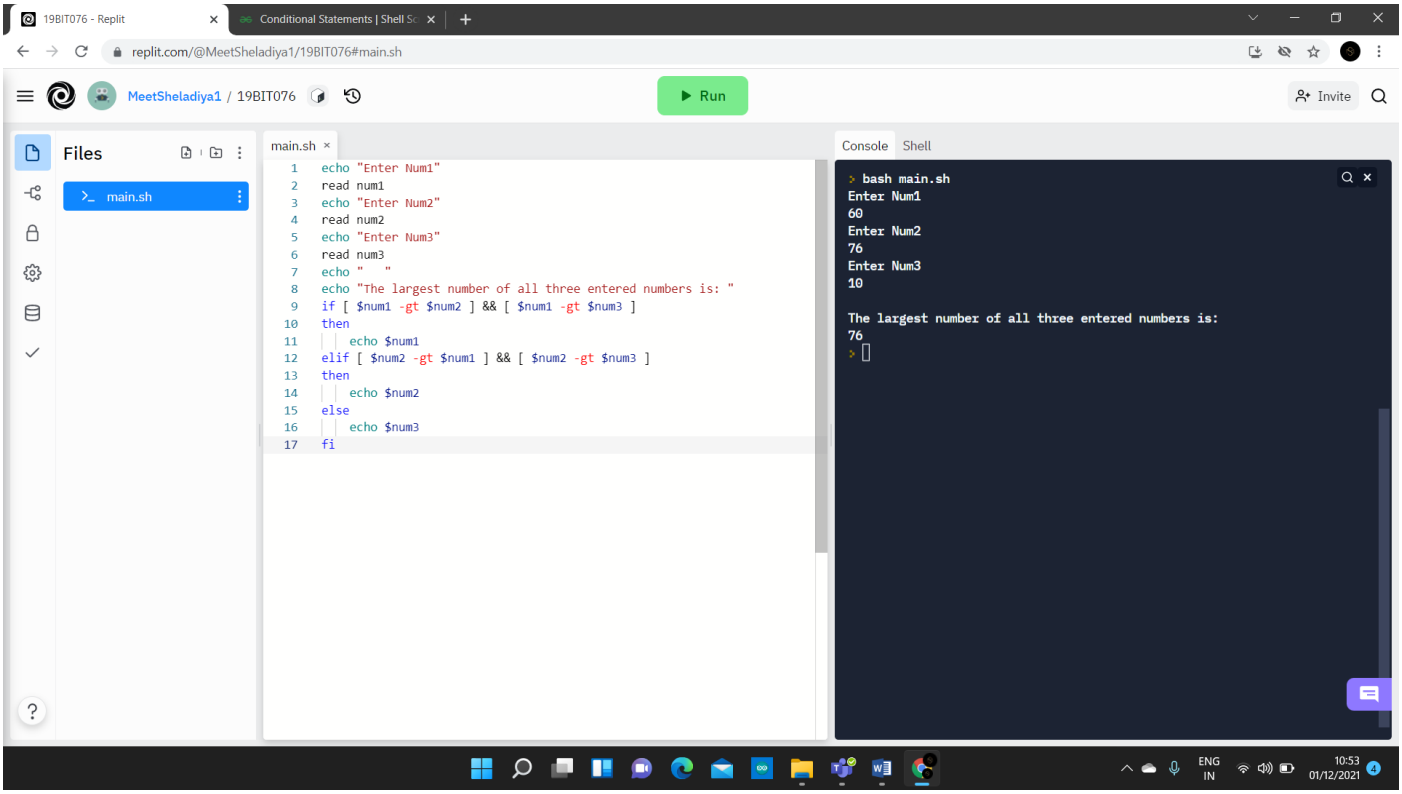


1.b) Write a shell script to find the largest of three numbers.

Code:

```
echo "Enter Num1"
read num1
echo "Enter Num2"
read num2
echo "Enter Num3"
read num3
echo "   "
echo "The largest number of all three entered numbers is: "
if [ $num1 -gt $num2 ] && [ $num1 -gt $num3 ]
then
   echo $num1
elif [ $num2 -gt $num1 ] && [ $num2 -gt $num3 ]
then
   echo $num2
else
   echo $num3
fi
```

Output:



2. Implement the Dining philosophers problem using semaphore.

Code:

```
#include<stdio.h>
#include<stdlib.h>
#include<pthread.h>
#include<semaphore.h>
#include<unistd.h>
sem_t chopstick[5];
void * philos(void *);
void eat(int);
int main()
 {
        int i,n[5];
        pthread_t T[5];
        for(i=0;i<5;i++)
        sem_init(&chopstick[i],0,1);
        for(i=0;i<5;i++){
                n[i]=i;
                pthread_create(&T[i],NULL,philos,(void *)&n[i]);
                }
        for(i=0;i<5;i++)
                pthread_join(T[i],NULL);
 }
void * philos(void * n)
 {
        int ph=*(int *)n;
        printf("Philosopher %d wants to eat\n",ph);
        printf("Philosopher %d tries to pick left chopstick\n",ph);
        sem_wait(&chopstick[ph]);
        printf("Philosopher %d picks the left chopstick\n",ph);
        printf("Philosopher %d tries to pick the right chopstick\n",ph);
        sem_wait(&chopstick[(ph+1)%5]);
        printf("Philosopher %d picks the right chopstick\n",ph);
        eat(ph);
        sleep(2);
        printf("Philosopher %d has finished eating\n",ph);
```

```
        sem_post(&chopstick[(ph+1)%5]);
        printf("Philosopher %d leaves the right chopstick\n",ph);
        sem_post(&chopstick[ph]);
        printf("Philosopher %d leaves the left chopstick\n",ph);
}
void eat(int ph)
{
        printf("Philosopher %d begins to eat\n",ph);
}
```

Output: