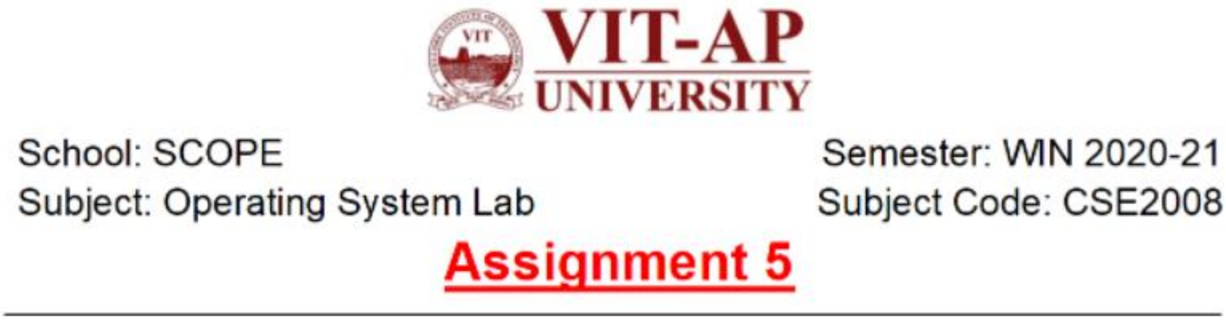


VIT-AP UNIVERSITY

School: SCOPE

Semester: WIN 2020-21

Subject: Operating System Lab

Subject Code: CSE2008

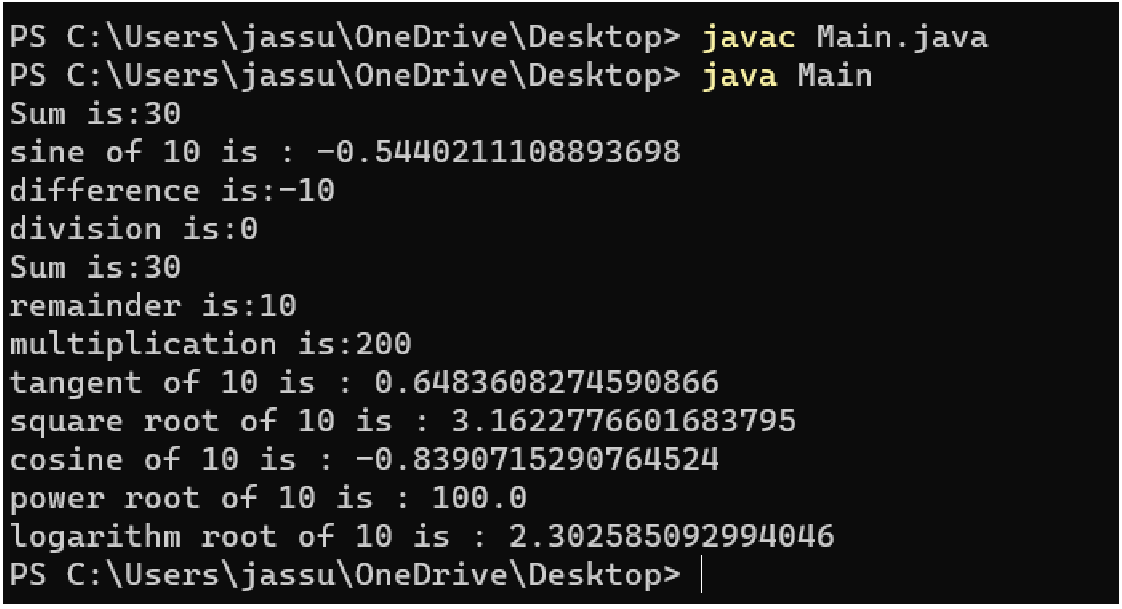
Assignment 5

NAME: MAJJIGA JASWANTH

Registration No: 20BCD7171

Question-1:
Write a Java program to create multiple threads for different calculator operations simultaneously. (calculator operations should includes arithmetic operations (+, - , *, / , %), trigonometric operations (sin, cos, tan) and few other operations (square root, power, log, etc.).
Code:

```
import java.lang.*;
import java.util.*;
class Add extends Thread
{
  void add (int a, int b)
  {
    System.out.println ("Sum is:" + (a + b));
  }
  public void run()
  {
     add(10,20);
  }
}
class Sub extends Thread
{
  void sub (int a, int b)
  {
   System.out.println ("difference is:" + (a - b));
  }
  public void run()
  {
     sub(10,20);
  }
}
class mult extends Thread
{
  void mult (int a, int b)
  {
    System.out.println ("multiplication is:" + (a * b));
  }
  public void run()
  {
      mult(10,20);
  }
}
class div extends Thread
{
  void div (int a, int b)
  {
    System.out.println ("division is:" + (a / b));
  }
  public void run()
  {
      div(10,20);
  }
}
class remainder extends Thread
{
  void remainder(int a, int b)
  {
   System.out.println ("remainder is:" + (a % b));
  }
  public void run()
  {
      remainder(10,20);
  }
}
class trignometricoperations extends Thread
{
  void trignometricoperations (int a)
  {
    System.out.println ("sine of " + a + " is : " + Math.sin (a));
    System.out.println ("tangent of " + a + " is : " + Math.tan(a));
    System.out.println ("cosine of " + a + " is : " + Math.cos(a));


  }
  public void run()
  {
      trignometricoperations(10);
  }
}
class otherfunctions extends Thread
{
  void otherfunctions (int a)
  {
    System.out.println ("square root of " + a + " is : " + Math.sqrt(a));
    System.out.println ("power root of " + a + " is : " + Math.pow(a,2));
   System.out.println ("logarithm root of " + a + " is : " + Math.log(a));
  }
  public void run()
  {
     otherfunctions(10);
  }
}
public class Main
{
   public static void main(String[] args)
   {
      Thread t1=new Add();
      Thread t2=new Sub();
      Thread t3=new mult();
      Thread t4=new div();
      Thread t5=new remainder();
      Thread t6=new trignometricoperations();
      Thread t7=new Add();
      Thread t8=new otherfunctions();
      t1.start();
      t2.start();
      t3.start();
      t4.start();
      t5.start();
      t6.start();
      t7.start();
      t8.start();
   }
}
```



```
PS C:\Users\jassu\OneDrive\Desktop> javac Main.java
PS C:\Users\jassu\OneDrive\Desktop> java Main
Sum is:30
sine of 10 is : -0.5440211108893698
difference is:-10
division is:0
Sum is:30
remainder is:10
multiplication is:200
tangent of 10 is : 0.6483608274590866
square root of 10 is : 3.1622776601683795
cosine of 10 is : -0.8390715290764524
power root of 10 is : 100.0
logarithm root of 10 is : 2.302585092994046
PS C:\Users\jassu\OneDrive\Desktop>
```

Question-2:
Write a program to multiply two matrices that uses threads to divide up the work necessary to compute the product of two matrices. There are several ways to improve the performance using threads. You need to divide the product in row dimension among multiple threads in the computation. That is, if you are computing A * B and A is a 10 x 10 matrix and B is a 10 x 10 matrix, your code should use threads to divide up the computation of the 10 rows of the product

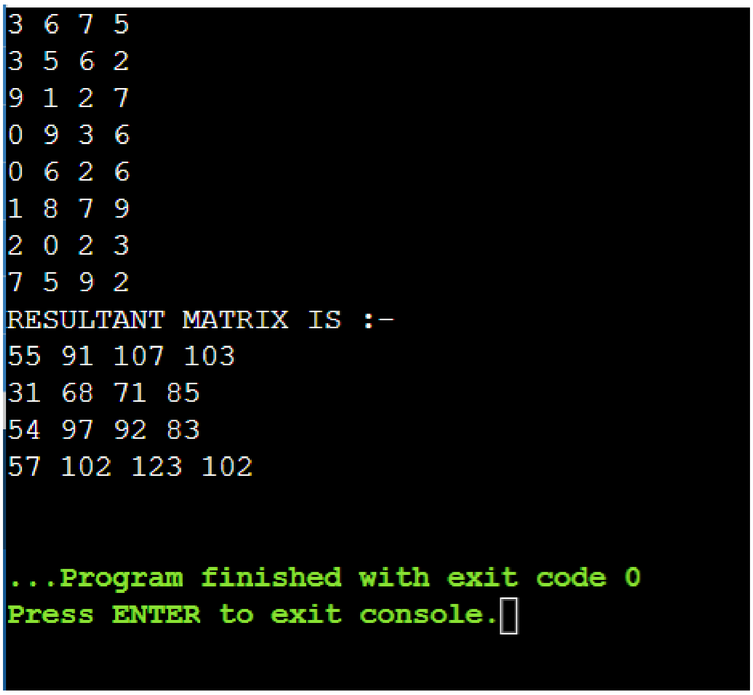
which is a 10 x 10 matrix. If you were to use 5 threads, then rows 0 and 1 of the product would be computed by thread 0, rows 2 and 3 would be computed by thread 1,...and rows 8 and 9 would be computed by thread 4

Code:

```
#include<stdio.h>
#include<pthread.h>
#include<unistd.h>
#include<stdlib.h>
#define MAX 4
void *mult(void* arg)
{
    int *data = (int *)arg;
    int k = 0, i = 0;

    int x = data[0];
    for (i = 1; i <= x; i++)
           k += data[i]*data[i+x];

    int *p = (int*)malloc(sizeof(int));
         *p = k;

    pthread_exit(p);
}
int main()
{

    int matA[MAX][MAX];
    int matB[MAX][MAX];


    int r1=MAX,c1=MAX,r2=MAX,c2=MAX,i,j,k;
    for (i = 0; i < r1; i++)
            for (j = 0; j < c1; j++)
                   matA[i][j] = rand() % 10;
    for (i = 0; i < r1; i++)
            for (j = 0; j < c1; j++)
                   matB[i][j] = rand() % 10;
    for (i = 0; i < r1; i++){
        for(j = 0; j < c1; j++)
            printf("%d ",matA[i][j]);
        printf("\n");
    }
    for (i = 0; i < r2; i++){
        for(j = 0; j < c2; j++)
            printf("%d ",matB[i][j]);
        printf("\n");
    }


    int max = r1*c2;
    pthread_t *threads;
    threads = (pthread_t*)malloc(max*sizeof(pthread_t));
```

```
  int count = 0;
  int* data = NULL;
  for (i = 0; i < r1; i++)
    for (j = 0; j < c2; j++)
        {
      data = (int *)malloc((20)*sizeof(int));
      data[0] = c1;

      for (k = 0; k < c1; k++)
        data[k+1] = matA[i][k];

      for (k = 0; k < r2; k++)
        data[k+c1+1] = matB[k][j];

        pthread_create(&threads[count++], NULL,
                mult, (void*)(data));

          }

  printf("RESULTANT MATRIX IS :- \n");
  for (i = 0; i < max; i++)
  {
   void *k;
   pthread_join(threads[i], &k);


     int *p = (int *)k;
   printf("%d ",*p);
   if ((i + 1) % c2 == 0)
     printf("\n");
  }
 return 0;
}
```

Output:



```
3 6 7 5
3 5 6 2
9 1 2 7
0 9 3 6
0 6 2 6
1 8 7 9
2 0 2 3
7 5 9 2
RESULTANT MATRIX IS :-
55 91 107 103
31 68 71 85
54 97 92 83
57 102 123 102


...Program finished with exit code 0
Press ENTER to exit console.
```