TE IT

Name : Omkar Gurav

Roll no : 8048

Implement multithreading for Matrix Multiplication using pthreads.

```
#include<stdio.h>
#include<stdlib.h>
#include<pthread.h>

Int
row_A,col_A,row_B,col_B,matrix_A[10][10],matrix_B[10][10],matrix_Result[10][10],s
um=0,cnt=0;

struct index
{
 int i,j;
};

void* multi_thread(void* arg)
{
 struct index* index=arg;
 sum=0;

 for(int k=0;k<row_B;k++)
 sum+=matrix_A[index->i][k]*matrix_B[k][index->j];

 matrix_Result[index->i][index->j]=sum;
 pthread_exit(0);
```

```
}

int main()
{
 printf("\nMatrix A");
 printf("\nEnter the number of rows in matrix A: ");
 scanf("%d",&row_A);
 printf("Enter the number of columns in matrix A: ");
 scanf("%d",&col_A);

 printf("\nMatrix B");
 printf("\nEnter the number of rows in matrix B: ");
 scanf("%d",&row_B);
 printf("Enter the number of columns in matrix B: ");
 scanf("%d",&col_B);

 if(col_A==row_B)
 {
  printf("\n\nEnter elements of matrix A\n");
  for(int i=0;i<row_A;i++)
  for(int j=0;j<col_A;j++)
  scanf("%d",&matrix_A[i][j]);

  printf("\nEnter elements of matrix B\n");
  for(int i=0;i<row_B;i++)
  for(int j=0;j<col_B;j++)
  scanf("%d", &matrix_B[i][j]);

  pthread_t tid[30];
```

```
for(int i=0;i<row_A;i++)
for(int j=0;j<col_B;j++)
{
 struct index* data=(struct index*)malloc(sizeof(struct index));
 data->i=i;
 data->j=j;

 pthread_create(&tid[cnt],NULL,multi_thread,data);
 pthread_join(tid[cnt],NULL);
 cnt++;
}

printf("\nResultant matrix\n");
for(int i=0;i<row_A;i++)
{
 for(int j=0;j<col_B;j++)
 printf("%d ",matrix_Result[i][j]);

 printf("\n");
}
 }

 else
 printf("\nMatrices cannot be multiplied, please try again\n");

 return 0;
}
```

Output:

Matrix A

Enter the number of rows in matrix A: 3

Enter the number of columns in matrix A: 3

Matrix B

Enter the number of rows in matrix B: 3

Enter the number of columns in matrix B: 3

Enter elements of matrix A

1

2

3

4

5

6

7

8

9

Enter elements of matrix B

1

2

1

2

4

6

7

2

5

Resultant matrix

26 16 28

56 40 64

86 64 100