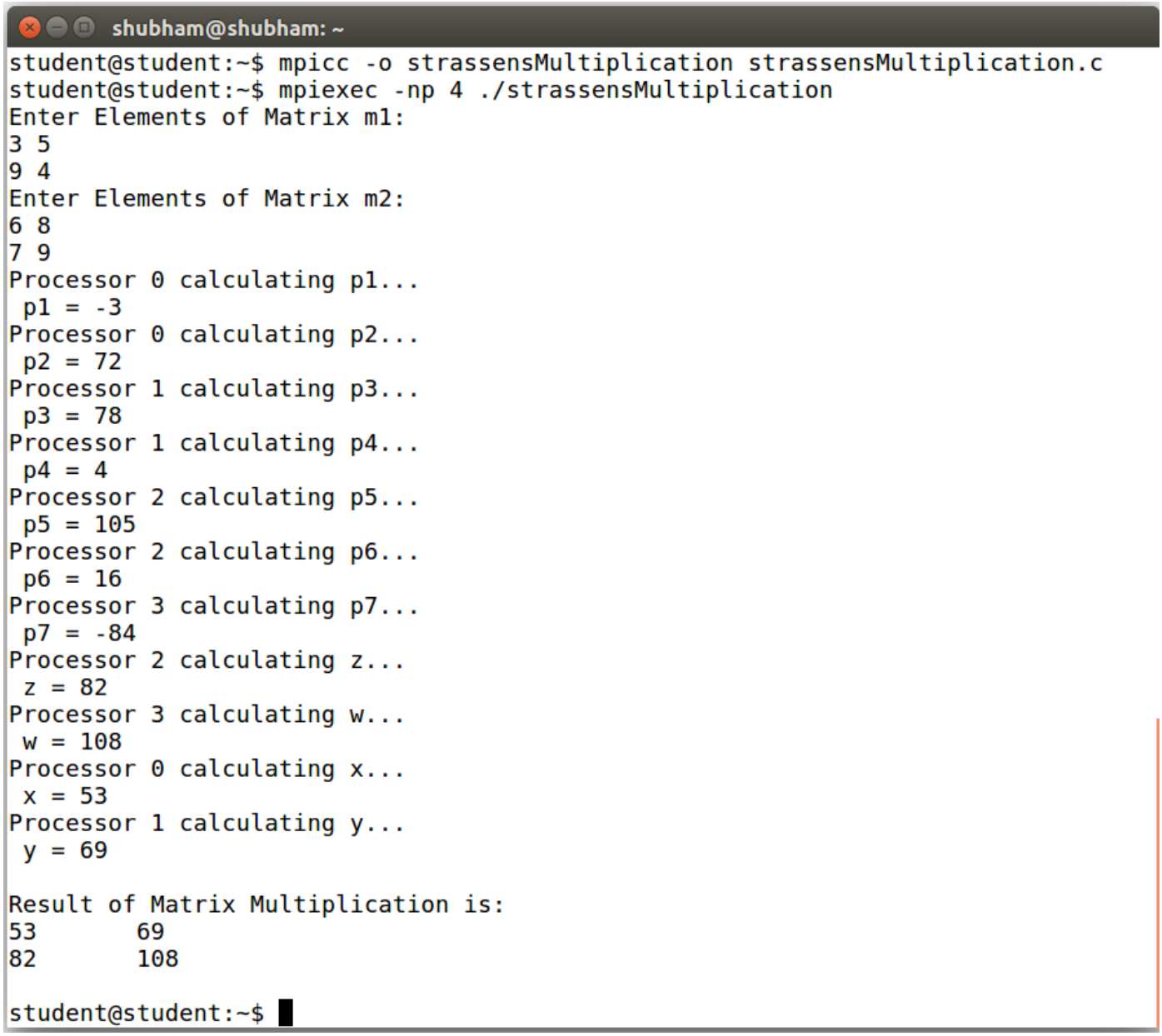
File: strassensMultiplication.c

```
#include<stdio.h>
#include<mpi.h>
int main() {
     MPI_Init(NULL,NULL);
     int rank,size;
     MPI_Comm_rank(MPI_COMM_WORLD,&rank);
     MPI_Comm_size(MPI_COMM_WORLD,&size);
     int p1,p2,p3,p4,p5,p6,p7;
     int x,y,z,w;
     int m1[2][2];
     int m2[2][2];
     int i,j,k;
     MPI_Status stat;

     if(rank==0) {
     // processor 0 takes input from user and send values to all other
processors
          printf("Enter Elements of Matrix m1:\n");
          for(i=0;i<2;i++)
               for(j=0;j<2;j++) {
                    if(scanf("%d",&m1[i][j])==1);
                    MPI_Bcast( &m1[i][j], 1, MPI_INT, 0,
MPI_COMM_WORLD);
               }

          printf("Enter Elements of Matrix m2:\n");
          for(i=0;i<2;i++)
               for(j=0;j<2;j++) {
                    if(scanf("%d",&m2[i][j])==1);
                    MPI_Bcast( &m2[i][j], 1, MPI_INT, 0,
MPI_COMM_WORLD);
               }

          //p0 calculates p1 and p2
          p1=m1[0][0]*(m2[0][1]-m2[1][1]);
          printf("Processor 0 calculating p1...\n p1 = %d\n",p1);
          p2=m2[1][1]*(m1[0][0]+m1[0][1]);
          printf("Processor 0 calculating p2...\n p2 = %d\n",p2);
     }

     else { //Receive matrix m1 and m2
          for(i=0;i<2;i++)
               for(j=0;j<2;j++)
                    MPI_Bcast( &m1[i][j], 1, MPI_INT, 0,
MPI_COMM_WORLD);

          for(i=0;i<2;i++)
               for(j=0;j<2;j++)
                    MPI_Bcast( &m2[i][j], 1, MPI_INT, 0,
MPI_COMM_WORLD);
     }

     if(rank==1) { //p1 calculates p3 and p4
          p3=m2[0][0]*(m1[1][0]+m1[1][1]);
          printf("Processor 1 calculating p3...\n p3 = %d\n",p3);
          p4=m1[1][1]*(m2[1][0]-m2[0][0]);
          printf("Processor 1 calculating p4...\n p4 = %d\n",p4);
```

```
    }

    if(rank==2) { //p2 calculates p5 and p6
        p5=(m1[0][0]+m1[1][1])*(m2[0][0]+m2[1][1]);
        printf("Processor 2 calculating p5...\n p5 = %d\n",p5);
        p6=(m1[0][1]-m1[1][1])*(m2[1][0]+m2[1][1]);
        printf("Processor 2 calculating p6...\n p6 = %d\n",p6);
    }

    if(rank==3) { //p3 calculates p7
        p7=(m1[0][0]-m1[1][0])*(m2[0][0]+m2[0][1]);
        printf("Processor 3 calculating p7...\n p7 = %d\n",p7);
    }

    //Broadcast values p1 to p7 to all other processors
    MPI_Bcast( &p1, 1, MPI_INT, 0, MPI_COMM_WORLD);
    MPI_Bcast( &p2, 1, MPI_INT, 0, MPI_COMM_WORLD );
    MPI_Bcast( &p3, 1, MPI_INT, 1, MPI_COMM_WORLD );
    MPI_Bcast( &p4, 1, MPI_INT, 1, MPI_COMM_WORLD );
    MPI_Bcast( &p5, 1, MPI_INT, 2, MPI_COMM_WORLD );
    MPI_Bcast( &p6, 1, MPI_INT, 2, MPI_COMM_WORLD );
    MPI_Bcast( &p7, 1, MPI_INT, 3, MPI_COMM_WORLD );

    //make sure all processors have identical values of p1 to p7
    MPI_Barrier(MPI_COMM_WORLD);

    if(rank==0) {
        x=p4+p5+p6-p2;
        printf("Processor 0 calculating x...\n x = %d\n",x);
        MPI_Send(&x,1,MPI_INT,3,1,MPI_COMM_WORLD);
    }

    if(rank==1) {
        y=p1+p2;
        printf("Processor 1 calculating y...\n y = %d\n",y);
        MPI_Send(&y,1,MPI_INT,3,1,MPI_COMM_WORLD);
    }

    if(rank==2) {
        z=p3+p4;
        printf("Processor 2 calculating z...\n z = %d\n",z);
        MPI_Send(&z,1,MPI_INT,3,1,MPI_COMM_WORLD);
    }

    if(rank==3) {
        w=p1+p5-p3-p7;
        printf("Processor 3 calculating w...\n w = %d\n",w);
        MPI_Recv(&y,1,MPI_INT,1,1,MPI_COMM_WORLD,&stat);
        MPI_Recv(&z,1,MPI_INT,2,1,MPI_COMM_WORLD,&stat);
        MPI_Recv(&x,1,MPI_INT,0,1,MPI_COMM_WORLD,&stat);
        printf("\nResult of Matrix Multiplication is: \n%d\t %d\n%d\t
%d\n\n",x,y,z,w);
    }

    MPI_Finalize();
    return 0;
}
```

#OUTPUT:


shubham@shubham: ~
student@student:~$ mpicc -o strassensMultiplication strassensMultiplication.c
student@student:~$ mpiexec -np 4 ./strassensMultiplication
Enter Elements of Matrix m1:
3 5
9 4
Enter Elements of Matrix m2:
6 8
7 9
Processor 0 calculating p1...
p1 = -3
Processor 0 calculating p2...
p2 = 72
Processor 1 calculating p3...
p3 = 78
Processor 1 calculating p4...
p4 = 4
Processor 2 calculating p5...
p5 = 105
Processor 2 calculating p6...
p6 = 16
Processor 3 calculating p7...
p7 = -84
Processor 2 calculating z...
z = 82
Processor 3 calculating w...
w = 108
Processor 0 calculating x...
x = 53
Processor 1 calculating y...
y = 69
Result of Matrix Multiplication is:
53 69
82 108
student@student:~$