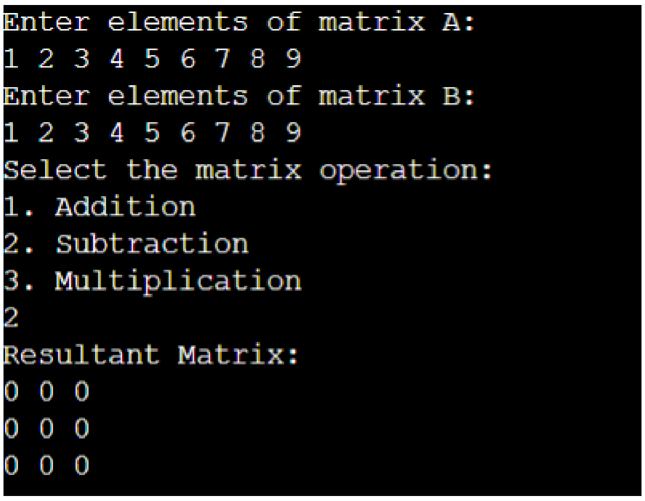
```
//Write a program to Implement multithreading for Matrix Operations using
//Pthreads in C language
#include <stdio.h>
#include <pthread.h>

#define SIZE 3

int A[SIZE][SIZE];
int B[SIZE][SIZE];
int C[SIZE][SIZE];

struct v {
   int i;
   int j;
};

void *matrix_add(void *data) {
   struct v *cell = (struct v *)data;
   C[cell->i][cell->j] = A[cell->i][cell->j] + B[cell->i][cell->j];
   pthread_exit(0);
}

void *matrix_subtract(void *data) {
   struct v *cell = (struct v *)data;
   C[cell->i][cell->j] = A[cell->i][cell->j] - B[cell->i][cell->j];
   pthread_exit(0);
}

void *matrix_multiply(void *data) {
   struct v *cell = (struct v *)data;
   int sum = 0;
```

```
    for (int k = 0; k < SIZE; k++) {
        sum += A[cell->i][k] * B[k][cell->j];
    }

    C[cell->i][cell->j] = sum;

    pthread_exit(0);
}

int main() {
    int choice;
    pthread_t threads[SIZE][SIZE];
    struct v data[SIZE][SIZE];

    printf("Enter elements of matrix A:\n");
    for (int i = 0; i < SIZE; i++) {
        for (int j = 0; j < SIZE; j++) {
            scanf("%d", &A[i][j]);
        }
    }

    printf("Enter elements of matrix B:\n");
    for (int i = 0; i < SIZE; i++) {
        for (int j = 0; j < SIZE; j++) {
            scanf("%d", &B[i][j]);
        }
    }

    printf("Select the matrix operation:\n");
    printf("1. Addition\n2. Subtraction\n3. Multiplication\n");
```

```
scanf("%d", &choice);

for (int i = 0; i < SIZE; i++) {
    for (int j = 0; j < SIZE; j++) {
        data[i][j].i = i;
        data[i][j].j = j;
        if (choice == 1) {
            pthread_create(&threads[i][j], NULL, matrix_add, (void *)&data[i][j]);
        } else if (choice == 2) {
            pthread_create(&threads[i][j], NULL, matrix_subtract, (void *)&data[i][j]);
        } else if (choice == 3) {
            pthread_create(&threads[i][j], NULL, matrix_multiply, (void *)&data[i][j]);
        }
    }
}

for (int i = 0; i < SIZE; i++) {
    for (int j = 0; j < SIZE; j++) {
        pthread_join(threads[i][j], NULL);
    }
}

printf("Resultant Matrix:\n");
for (int i = 0; i < SIZE; i++) {
    for (int j = 0; j < SIZE; j++) {
        printf("%d ", C[i][j]);
    }
    printf("\n");
}

return 0;
```

}

Output-



```
Enter elements of matrix A:
1 2 3 4 5 6 7 8 9
Enter elements of matrix B:
1 2 3 4 5 6 7 8 9
Select the matrix operation:
1. Addition
2. Subtraction
3. Multiplication
2
Resultant Matrix:
0 0 0
0 0 0
0 0 0
```