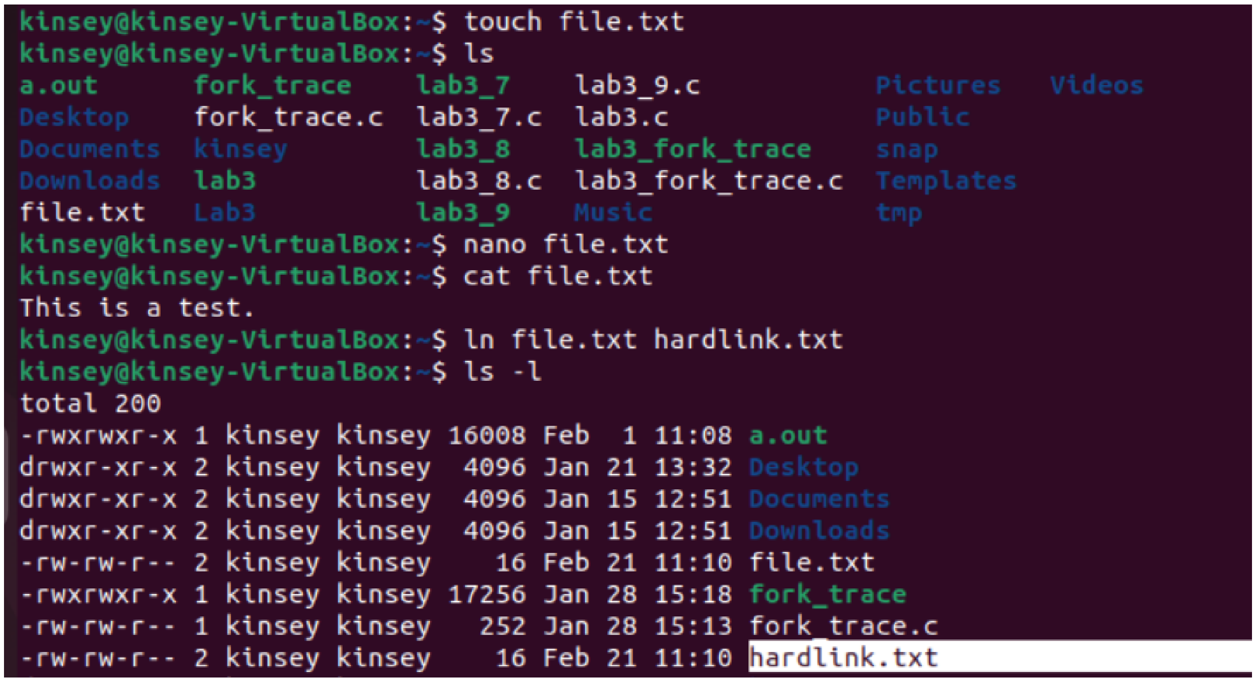
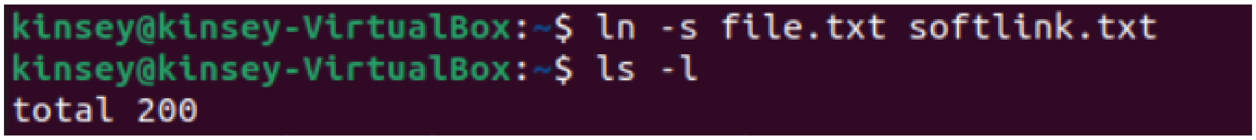
1. Creating hard link file:

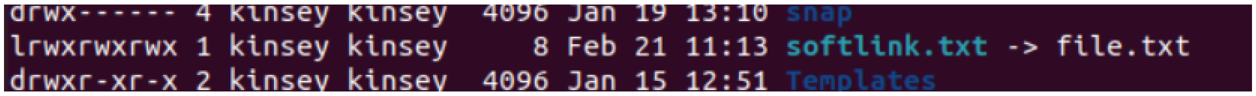


```
kinsey@kinsey-VirtualBox:~$ touch file.txt
kinsey@kinsey-VirtualBox:~$ ls
a.out      fork_trace     lab3_7    lab3_9.c          Pictures  Videos
Desktop    fork_trace.c   lab3_7.c  lab3.c            Public
Documents  kinsey         lab3_8    lab3_fork_trace   snap
Downloads  lab3           lab3_8.c  lab3_fork_trace.c Templates
file.txt   Lab3           lab3_9    Music             tmp
kinsey@kinsey-VirtualBox:~$ nano file.txt
kinsey@kinsey-VirtualBox:~$ cat file.txt
This is a test.
kinsey@kinsey-VirtualBox:~$ ln file.txt hardlink.txt
kinsey@kinsey-VirtualBox:~$ ls -l
total 200
-rwxrwxr-x 1 kinsey kinsey 16008 Feb  1 11:08 a.out
drwxr-xr-x 2 kinsey kinsey  4096 Jan 21 13:32 Desktop
drwxr-xr-x 2 kinsey kinsey  4096 Jan 15 12:51 Documents
drwxr-xr-x 2 kinsey kinsey  4096 Jan 15 12:51 Downloads
-rw-rw-r-- 2 kinsey kinsey    16 Feb 21 11:10 file.txt
-rwxrwxr-x 1 kinsey kinsey 17256 Jan 28 15:18 fork_trace
-rw-rw-r-- 1 kinsey kinsey   252 Jan 28 15:13 fork_trace.c
-rw-rw-r-- 2 kinsey kinsey    16 Feb 21 11:10 hardlink.txt
```

Creating soft link file:

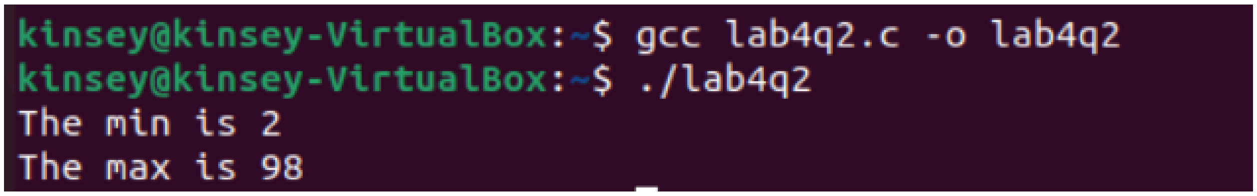


```
kinsey@kinsey-VirtualBox:~$ ln -s file.txt softlink.txt
kinsey@kinsey-VirtualBox:~$ ls -l
total 200
```



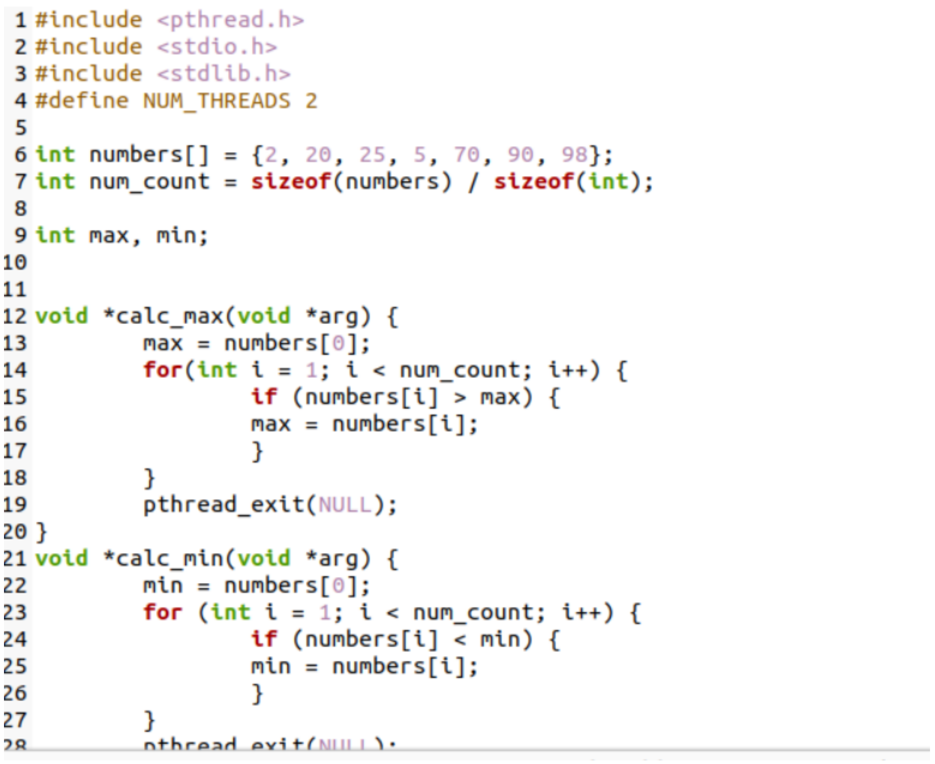
```
drwx------ 4 kinsey kinsey  4096 Jan 19 13:10 snap
lrwxrwxrwx 1 kinsey kinsey     8 Feb 21 11:13 softlink.txt -> file.txt
drwxr-xr-x 2 kinsey kinsey  4096 Jan 15 12:51 Templates
```

2. Output:

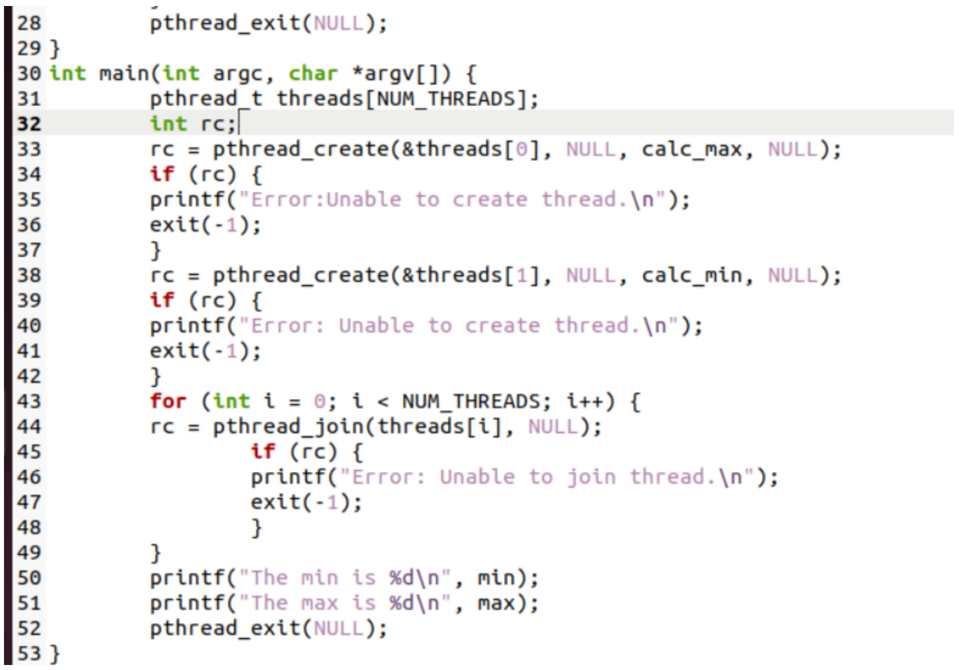


```
kinsey@kinsey-VirtualBox:~$ gcc lab4q2.c -o lab4q2
kinsey@kinsey-VirtualBox:~$ ./lab4q2
The min is 2
The max is 98
```

C program:

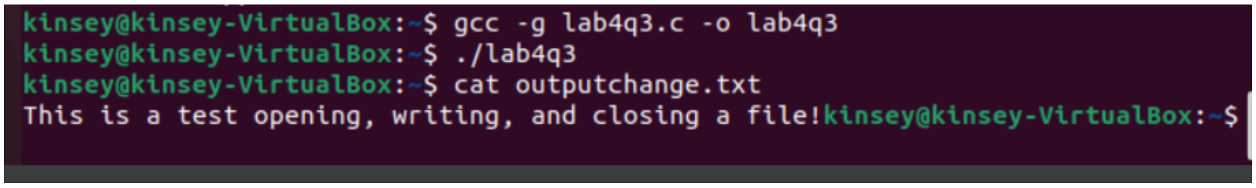


```c
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#define NUM_THREADS 2

int numbers[] = {2, 20, 25, 5, 70, 90, 98};
int num_count = sizeof(numbers) / sizeof(int);

int max, min;


void *calc_max(void *arg) {
        max = numbers[0];
        for(int i = 1; i < num_count; i++) {
                if (numbers[i] > max) {
                max = numbers[i];
                }
        }
        pthread_exit(NULL);
}
void *calc_min(void *arg) {
        min = numbers[0];
        for (int i = 1; i < num_count; i++) {
                if (numbers[i] < min) {
                min = numbers[i];
                }
        }
        pthread_exit(NULL);
```



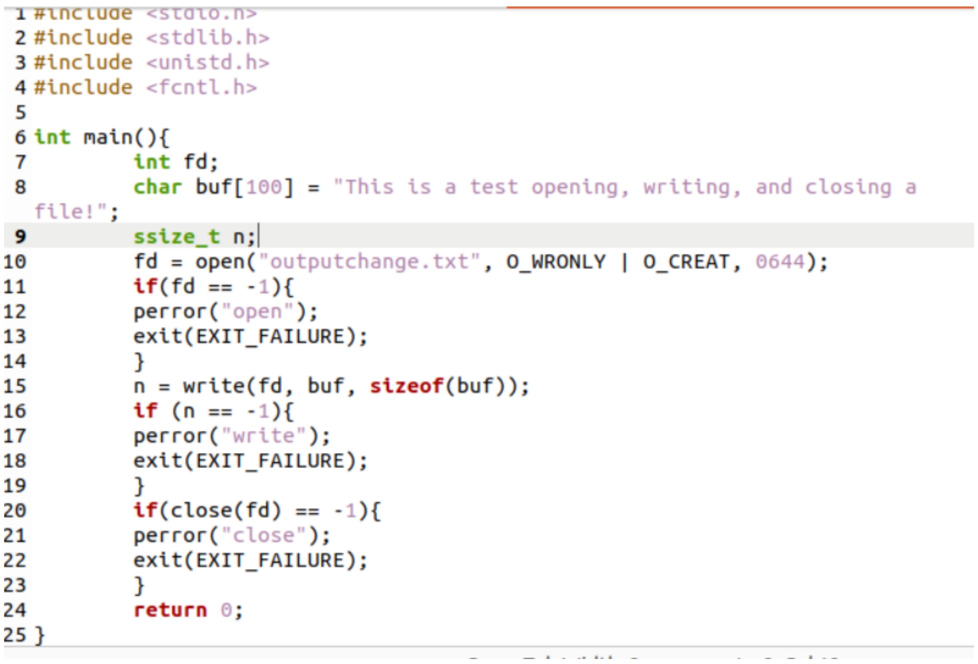
```c
        pthread_exit(NULL);
}
int main(int argc, char *argv[]) {
        pthread_t threads[NUM_THREADS];
        int rc;
        rc = pthread_create(&threads[0], NULL, calc_max, NULL);
        if (rc) {
        printf("Error:Unable to create thread.\n");
        exit(-1);
        }
        rc = pthread_create(&threads[1], NULL, calc_min, NULL);
        if (rc) {
        printf("Error: Unable to create thread.\n");
        exit(-1);
        }
        for (int i = 0; i < NUM_THREADS; i++) {
        rc = pthread_join(threads[i], NULL);
                if (rc) {
                printf("Error: Unable to join thread.\n");
                exit(-1);
                }
        }
        printf("The min is %d\n", min);
        printf("The max is %d\n", max);
        pthread_exit(NULL);
}
```

3. Output:



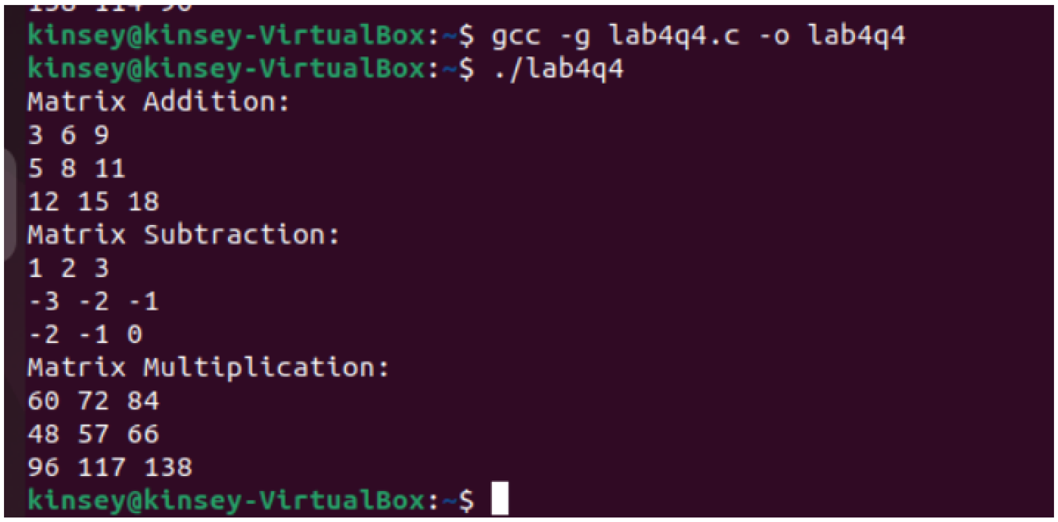
```
kinsey@kinsey-VirtualBox:~$ gcc -g lab4q3.c -o lab4q3
kinsey@kinsey-VirtualBox:~$ ./lab4q3
kinsey@kinsey-VirtualBox:~$ cat outputchange.txt
This is a test opening, writing, and closing a file!kinsey@kinsey-VirtualBox:~$
```

C program:



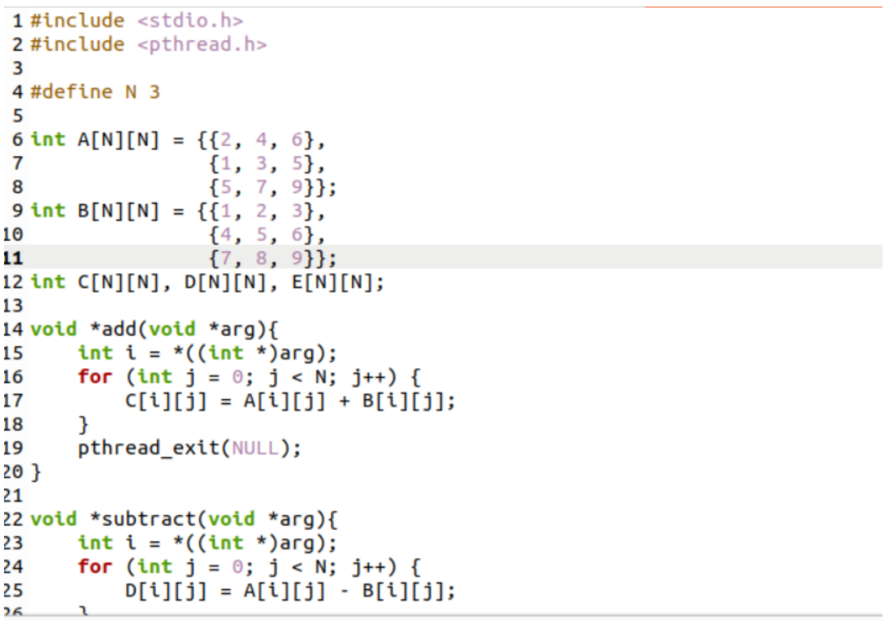
```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>

int main(){
        int fd;
        char buf[100] = "This is a test opening, writing, and closing a 
file!";
        ssize_t n;
        fd = open("outputchange.txt", O_WRONLY | O_CREAT, 0644);
        if(fd == -1){
        perror("open");
        exit(EXIT_FAILURE);
        }
        n = write(fd, buf, sizeof(buf));
        if (n == -1){
        perror("write");
        exit(EXIT_FAILURE);
        }
        if(close(fd) == -1){
        perror("close");
        exit(EXIT_FAILURE);
        }
        return 0;
}
```

4. Output:

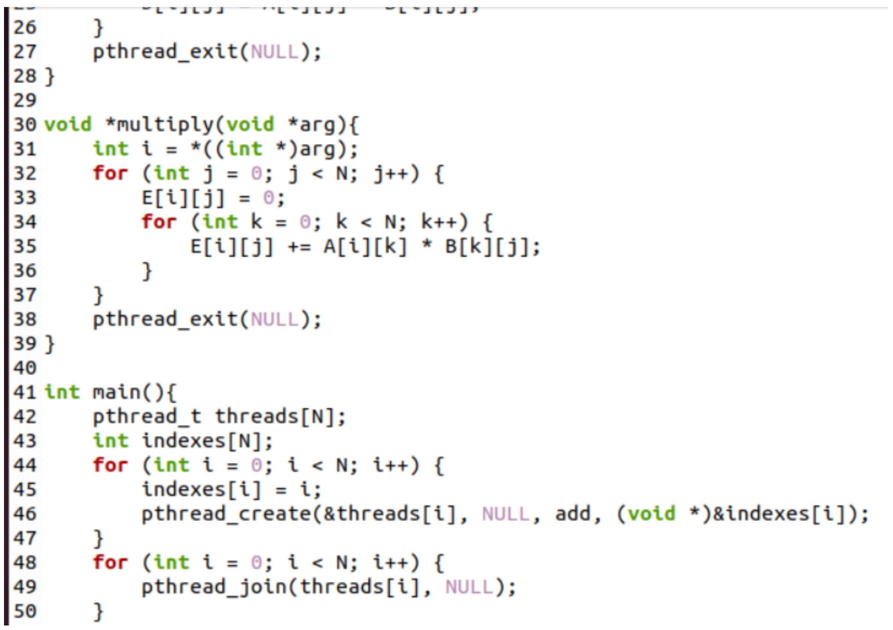


```
kinsey@kinsey-VirtualBox:~$ gcc -g lab4q4.c -o lab4q4
kinsey@kinsey-VirtualBox:~$ ./lab4q4
Matrix Addition:
3 6 9
5 8 11
12 15 18
Matrix Subtraction:
1 2 3
-3 -2 -1
-2 -1 0
Matrix Multiplication:
60 72 84
48 57 66
96 117 138
kinsey@kinsey-VirtualBox:~$
```

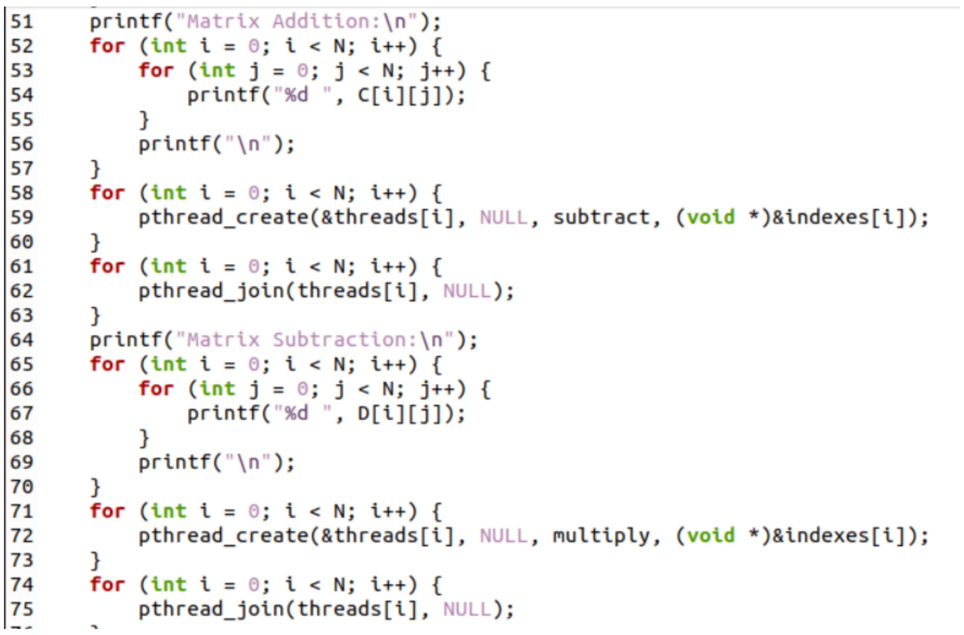
C program:



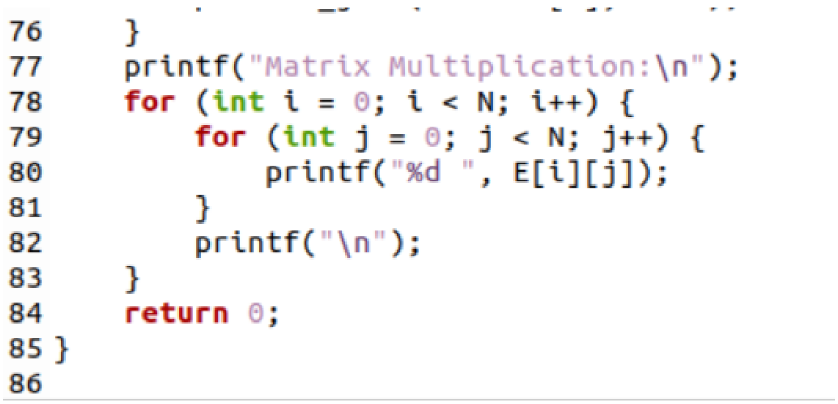
```c
#include <stdio.h>
#include <pthread.h>

#define N 3

int A[N][N] = {{2, 4, 6},
               {1, 3, 5},
               {5, 7, 9}};
int B[N][N] = {{1, 2, 3},
               {4, 5, 6},
               {7, 8, 9}};
int C[N][N], D[N][N], E[N][N];

void *add(void *arg){
    int i = *((int *)arg);
    for (int j = 0; j < N; j++) {
        C[i][j] = A[i][j] + B[i][j];
    }
    pthread_exit(NULL);
}

void *subtract(void *arg){
    int i = *((int *)arg);
    for (int j = 0; j < N; j++) {
        D[i][j] = A[i][j] - B[i][j];
```



```c
    }
    pthread_exit(NULL);
}

void *multiply(void *arg){
    int i = *((int *)arg);
    for (int j = 0; j < N; j++) {
        E[i][j] = 0;
        for (int k = 0; k < N; k++) {
            E[i][j] += A[i][k] * B[k][j];
        }
    }
    pthread_exit(NULL);
}

int main(){
    pthread_t threads[N];
    int indexes[N];
    for (int i = 0; i < N; i++) {
        indexes[i] = i;
        pthread_create(&threads[i], NULL, add, (void *)&indexes[i]);
    }
    for (int i = 0; i < N; i++) {
        pthread_join(threads[i], NULL);
    }
```



```c
    printf("Matrix Addition:\n");
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            printf("%d ", C[i][j]);
        }
        printf("\n");
    }
    for (int i = 0; i < N; i++) {
        pthread_create(&threads[i], NULL, subtract, (void *)&indexes[i]);
    }
    for (int i = 0; i < N; i++) {
        pthread_join(threads[i], NULL);
    }
    printf("Matrix Subtraction:\n");
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            printf("%d ", D[i][j]);
        }
        printf("\n");
    }
    for (int i = 0; i < N; i++) {
        pthread_create(&threads[i], NULL, multiply, (void *)&indexes[i]);
    }
    for (int i = 0; i < N; i++) {
        pthread_join(threads[i], NULL);
```



```c
    }
    printf("Matrix Multiplication:\n");
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            printf("%d ", E[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```