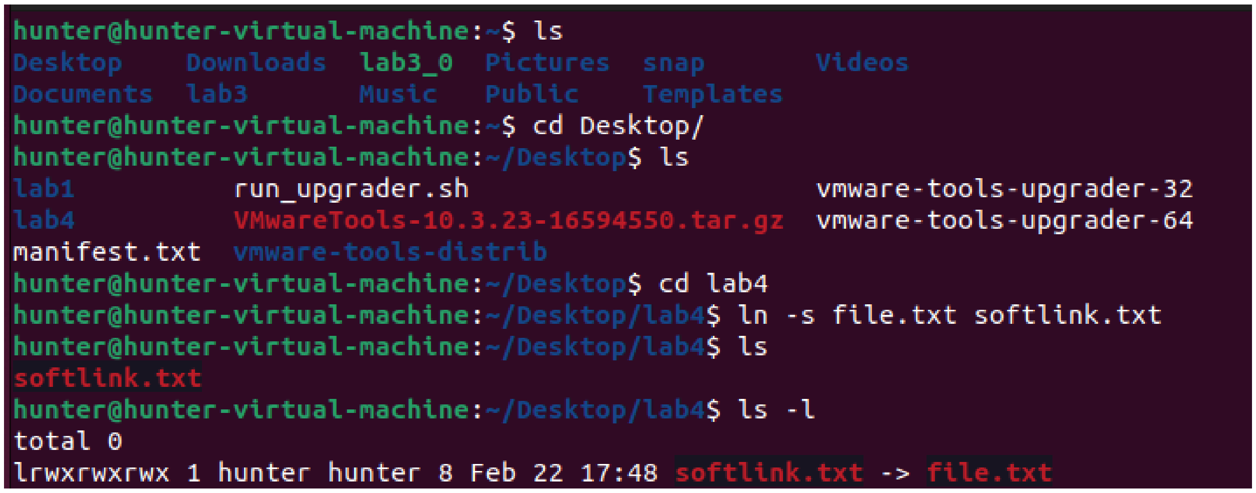
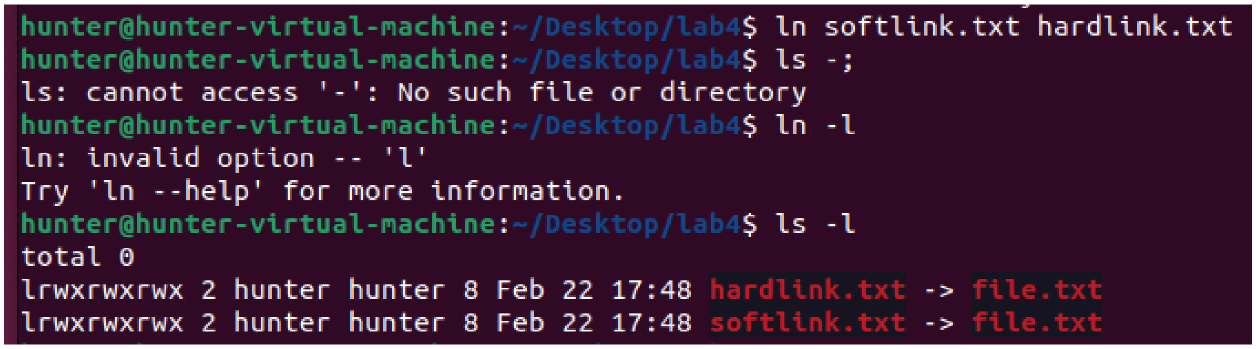
1

Soft link



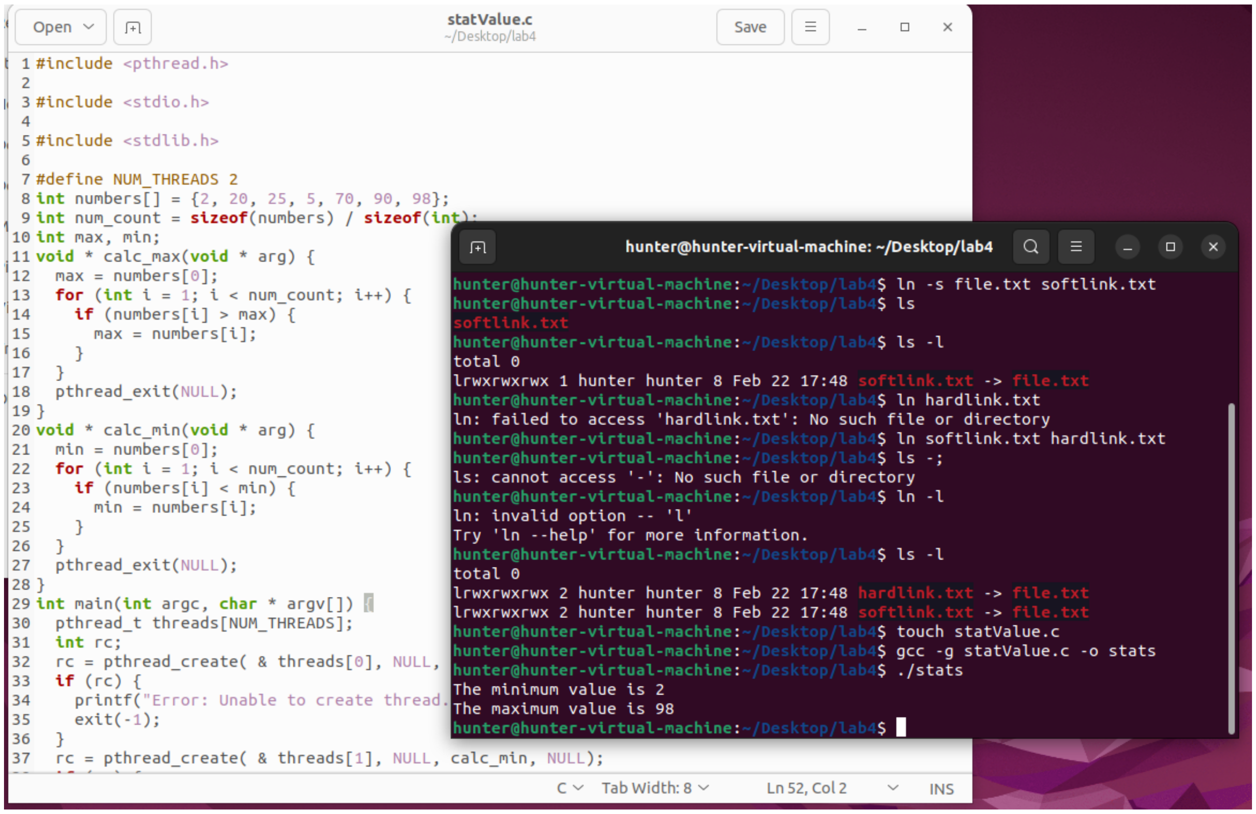
```
hunter@hunter-virtual-machine:~$ ls
Desktop     Downloads  lab3_0  Pictures  snap       Videos
Documents   lab3       Music   Public    Templates
hunter@hunter-virtual-machine:~$ cd Desktop/
hunter@hunter-virtual-machine:~/Desktop$ ls
lab1          run_upgrader.sh                    vmware-tools-upgrader-32
lab4          VMwareTools-10.3.23-16594550.tar.gz  vmware-tools-upgrader-64
manifest.txt  vmware-tools-distrib
hunter@hunter-virtual-machine:~/Desktop$ cd lab4
hunter@hunter-virtual-machine:~/Desktop/lab4$ ln -s file.txt softlink.txt
hunter@hunter-virtual-machine:~/Desktop/lab4$ ls
softlink.txt
hunter@hunter-virtual-machine:~/Desktop/lab4$ ls -l
total 0
lrwxrwxrwx 1 hunter hunter 8 Feb 22 17:48 softlink.txt -> file.txt
```

Hard link



```
hunter@hunter-virtual-machine:~/Desktop/lab4$ ln softlink.txt hardlink.txt
hunter@hunter-virtual-machine:~/Desktop/lab4$ ls -;
ls: cannot access '-': No such file or directory
hunter@hunter-virtual-machine:~/Desktop/lab4$ ln -l
ln: invalid option -- 'l'
Try 'ln --help' for more information.
hunter@hunter-virtual-machine:~/Desktop/lab4$ ls -l
total 0
lrwxrwxrwx 2 hunter hunter 8 Feb 22 17:48 hardlink.txt -> file.txt
lrwxrwxrwx 2 hunter hunter 8 Feb 22 17:48 softlink.txt -> file.txt
```

2



```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);
}
int main(int argc, char * argv[]) {
  pthread_t threads[NUM_THREADS];
  int rc;
  rc = pthread_create( & threads[0], NULL,
  if (rc) {
    printf("Error: Unable to create thread.
    exit(-1);
  }
  rc = pthread_create( & threads[1], NULL, calc_min, NULL);
```

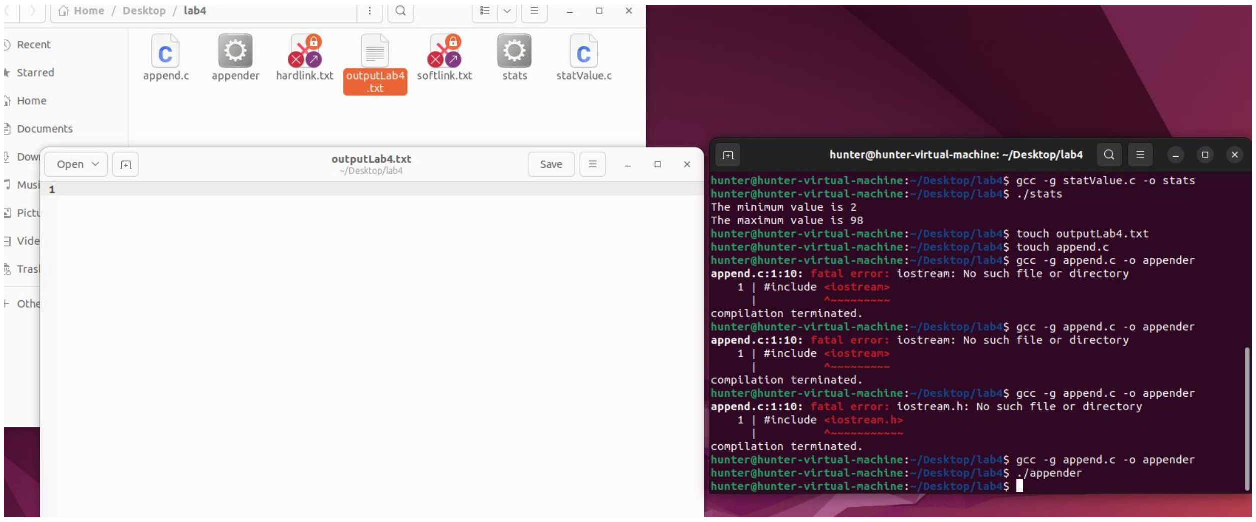
```
hunter@hunter-virtual-machine:~/Desktop/lab4$ ln -s file.txt softlink.txt
hunter@hunter-virtual-machine:~/Desktop/lab4$ ls
softlink.txt
hunter@hunter-virtual-machine:~/Desktop/lab4$ ls -l
total 0
lrwxrwxrwx 1 hunter hunter 8 Feb 22 17:48 softlink.txt -> file.txt
hunter@hunter-virtual-machine:~/Desktop/lab4$ ln hardlink.txt
ln: failed to access 'hardlink.txt': No such file or directory
hunter@hunter-virtual-machine:~/Desktop/lab4$ ln softlink.txt hardlink.txt
hunter@hunter-virtual-machine:~/Desktop/lab4$ ls -;
ls: cannot access '-': No such file or directory
hunter@hunter-virtual-machine:~/Desktop/lab4$ ln -l
ln: invalid option -- 'l'
Try 'ln --help' for more information.
hunter@hunter-virtual-machine:~/Desktop/lab4$ ls -l
total 0
lrwxrwxrwx 2 hunter hunter 8 Feb 22 17:48 hardlink.txt -> file.txt
lrwxrwxrwx 2 hunter hunter 8 Feb 22 17:48 softlink.txt -> file.txt
hunter@hunter-virtual-machine:~/Desktop/lab4$ touch statValue.c
hunter@hunter-virtual-machine:~/Desktop/lab4$ gcc -g statValue.c -o stats
hunter@hunter-virtual-machine:~/Desktop/lab4$ ./stats
The minimum value is 2
The maximum value is 98
hunter@hunter-virtual-machine:~/Desktop/lab4$
```

```
"#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);
}
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) {
```

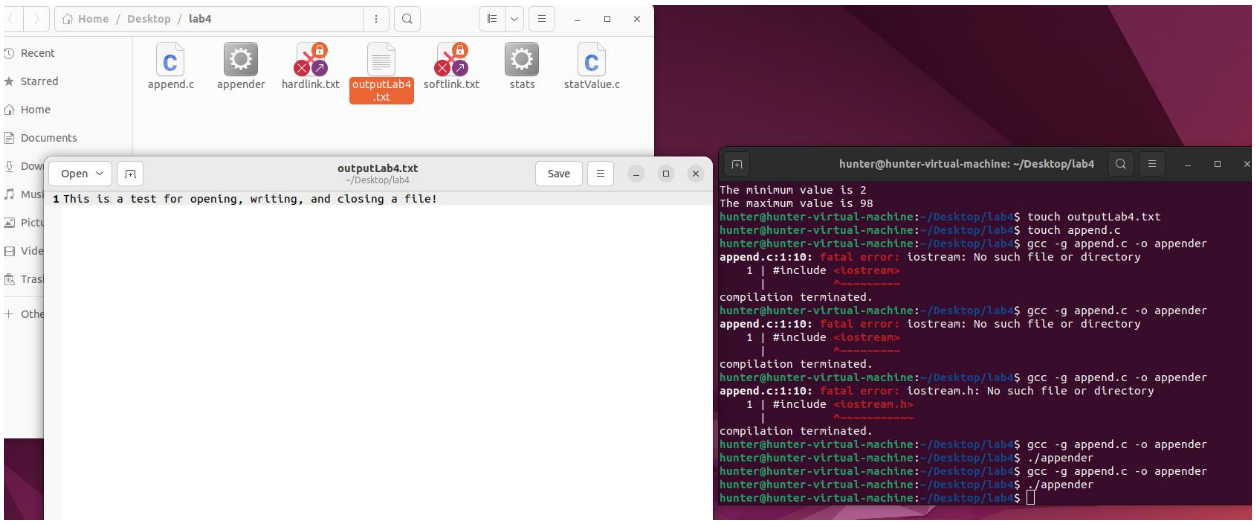
```
 printf("Error: Unable to join thread.\n");
 exit(-1);
 }
 }
 printf("The minimum value is %d\n", min);
 printf("The maximum value is %d\n", max);
 pthread_exit(NULL);
}”
```

The code defines two functions, calc_max and calc_min, that calculate the maximum and minimum of an array of numbers. The main function creates two threads, one for each function, and passes them the array as an argument. Then it waits for the threads to finish their work by calling pthread_join. Finally, it prints the results and exits.

3



```
hunter@hunter-virtual-machine:~/Desktop/lab4$ gcc -g statValue.c -o stats
hunter@hunter-virtual-machine:~/Desktop/lab4$ ./stats
The minimum value is 2
The maximum value is 98
hunter@hunter-virtual-machine:~/Desktop/lab4$ touch outputLab4.txt
hunter@hunter-virtual-machine:~/Desktop/lab4$ touch append.c
hunter@hunter-virtual-machine:~/Desktop/lab4$ gcc -g append.c -o appender
append.c:1:10: fatal error: iostream: No such file or directory
    1 | #include <iostream>
      |          ^~~~~~~~~~
compilation terminated.
hunter@hunter-virtual-machine:~/Desktop/lab4$ gcc -g append.c -o appender
append.c:1:10: fatal error: iostream: No such file or directory
    1 | #include <iostream>
      |          ^~~~~~~~~~
compilation terminated.
hunter@hunter-virtual-machine:~/Desktop/lab4$ gcc -g append.c -o appender
append.c:1:10: fatal error: iostream.h: No such file or directory
    1 | #include <iostream.h>
      |          ^~~~~~~~~~~~
compilation terminated.
hunter@hunter-virtual-machine:~/Desktop/lab4$ gcc -g append.c -o appender
hunter@hunter-virtual-machine:~/Desktop/lab4$ ./appender
hunter@hunter-virtual-machine:~/Desktop/lab4$ 
```



```
This is a test for opening, writing, and closing a file!
```

```
The minimum value is 2
The maximum value is 98
hunter@hunter-virtual-machine:~/Desktop/lab4$ touch outputLab4.txt
hunter@hunter-virtual-machine:~/Desktop/lab4$ touch append.c
hunter@hunter-virtual-machine:~/Desktop/lab4$ gcc -g append.c -o appender
append.c:1:10: fatal error: iostream: No such file or directory
    1 | #include <iostream>
      |          ^~~~~~~~~~
compilation terminated.
hunter@hunter-virtual-machine:~/Desktop/lab4$ gcc -g append.c -o appender
append.c:1:10: fatal error: iostream: No such file or directory
    1 | #include <iostream>
      |          ^~~~~~~~~~
compilation terminated.
hunter@hunter-virtual-machine:~/Desktop/lab4$ gcc -g append.c -o appender
append.c:1:10: fatal error: iostream.h: No such file or directory
    1 | #include <iostream.h>
      |          ^~~~~~~~~~~~
compilation terminated.
hunter@hunter-virtual-machine:~/Desktop/lab4$ gcc -g append.c -o appender
hunter@hunter-virtual-machine:~/Desktop/lab4$ ./appender
hunter@hunter-virtual-machine:~/Desktop/lab4$ gcc -g append.c -o appender
hunter@hunter-virtual-machine:~/Desktop/lab4$ ./appender
hunter@hunter-virtual-machine:~/Desktop/lab4$ 
```

```
#include <stdio.h>


int main()

{

    // Create a FILE pointer

    FILE *file;


    // Open "outputLab4.txt" for writing

    file = fopen("outputLab4.txt", "w");
```

```
   // Check if the file is opened successfully
   if (file != NULL)
   {
      // Write to the file
      fprintf(file, "This is a test for opening, writing, and closing a file!\n");

      // Close the file
      fclose(file);
   }
   else
   {
      // Display an error message
      printf("Unable to open file\n");
   }

   return 0;
}
```

includes the stdio.h header file, which provides functions for input and output operations

main function, which is the entry point of a C program

    declares a FILE pointer, which is used to access a file

    fopen to open a file named “outputLab4.txt”

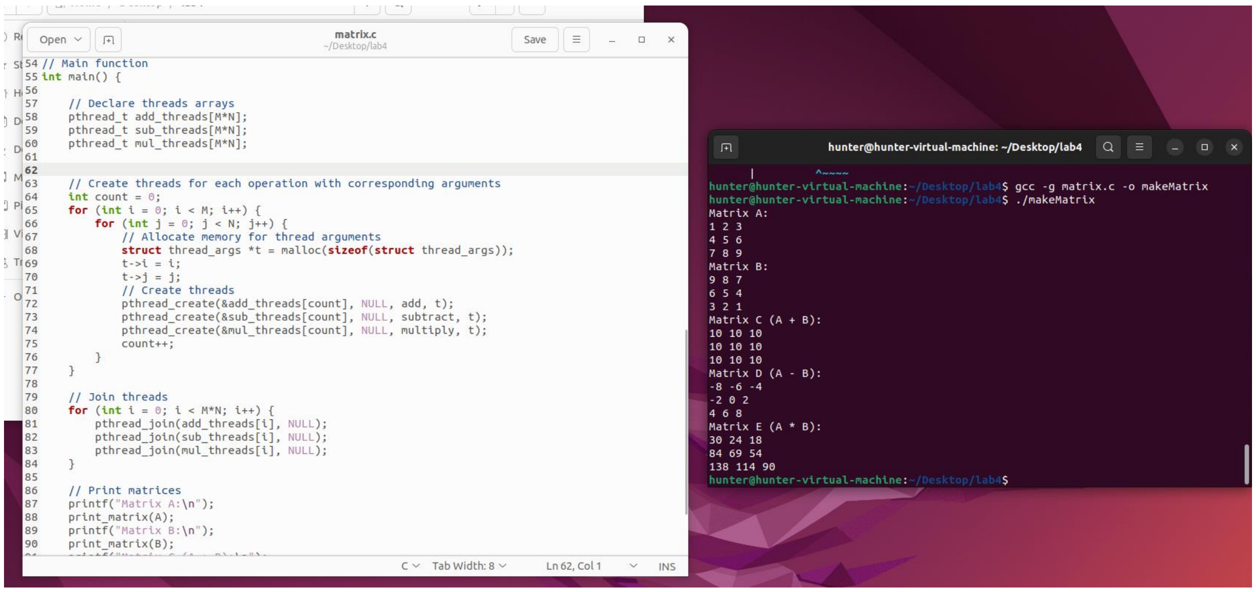
    if fopen succeeds, it returns a valid pointer to the file; otherwise, it returns NULL

checks if the pointer is not NULL before using fprintf to write a string to the file

fclose to close the file and free any resources

return 0

4



```c
// Main function
int main() {

    // Declare threads arrays
    pthread_t add_threads[M*N];
    pthread_t sub_threads[M*N];
    pthread_t mul_threads[M*N];


    // Create threads for each operation with corresponding arguments
    int count = 0;
    for (int i = 0; i < M; i++) {
        for (int j = 0; j < N; j++) {
            // Allocate memory for thread arguments
            struct thread_args *t = malloc(sizeof(struct thread_args));
            t->i = i;
            t->j = j;
            // Create threads
            pthread_create(&add_threads[count], NULL, add, t);
            pthread_create(&sub_threads[count], NULL, subtract, t);
            pthread_create(&mul_threads[count], NULL, multiply, t);
            count++;
        }
    }

    // Join threads
    for (int i = 0; i < M*N; i++) {
        pthread_join(add_threads[i], NULL);
        pthread_join(sub_threads[i], NULL);
        pthread_join(mul_threads[i], NULL);
    }

    // Print matrices
    printf("Matrix A:\n");
    print_matrix(A);
    printf("Matrix B:\n");
    print_matrix(B);
```

```
      |          ^~~~~
hunter@hunter-virtual-machine:~/Desktop/lab4$ gcc -g matrix.c -o makeMatrix
hunter@hunter-virtual-machine:~/Desktop/lab4$ ./makeMatrix
Matrix A:
1 2 3
4 5 6
7 8 9
Matrix B:
9 8 7
6 5 4
3 2 1
Matrix C (A + B):
10 10 10
10 10 10
10 10 10
Matrix D (A - B):
-8 -6 -4
-2 0 2
4 6 8
Matrix E (A * B):
30 24 18
84 69 54
138 114 90
hunter@hunter-virtual-machine:~/Desktop/lab4$
```

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

// Define matrix size
#define M 3
#define N 3

// Declare global matrices
int A[M][N] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
int B[M][N] = {{9, 8, 7}, {6, 5, 4}, {3, 2, 1}};
int C[M][N]; // Result of addition
int D[M][N]; // Result of subtraction
int E[M][N]; // Result of multiplication

// Define thread arguments structure
struct thread_args {
  int i; // Row index
  int j; // Column index
};

// Define thread functions for each operation
void *add(void *args) {
  struct thread_args *t = (struct thread_args *)args;
  C[t->i][t->j] = A[t->i][t->j] + B[t->i][t->j];
  pthread_exit(NULL);
```

```
}

void *subtract(void *args) {
   struct thread_args *t = (struct thread_args *)args;
   D[t->i][t->j] = A[t->i][t->j] - B[t->i][t->j];
   pthread_exit(NULL);
}

void *multiply(void *args) {
   struct thread_args *t = (struct thread_args *)args;
   E[t->i][t->j] = 0;
   for (int k = 0; k < N; k++) {
      E[t->i][t->j] += A[t->i][k] * B[k][t->j];
   }
   pthread_exit(NULL);
}

// Define a function to print a matrix
void print_matrix(int mat[M][N]) {
   for (int i = 0; i < M; i++) {
      for (int j = 0; j < N; j++) {
         printf("%d ", mat[i][j]);
      }
      printf("\n");
   }
}

// Main function
int main() {

   // Declare threads arrays
   pthread_t add_threads[M*N];
   pthread_t sub_threads[M*N];
   pthread_t mul_threads[M*N];


   // Create threads for each operation with corresponding arguments
   int count = 0;
   for (int i = 0; i < M; i++) {
      for (int j = 0; j < N; j++) {
         // Allocate memory for thread arguments
         struct thread_args *t = malloc(sizeof(struct thread_args));
         t->i = i;
         t->j = j;
```

```
            // Create threads
            pthread_create(&add_threads[count], NULL, add, t);
            pthread_create(&sub_threads[count], NULL, subtract, t);
            pthread_create(&mul_threads[count], NULL, multiply, t);
            count++;
        }
    }

    // Join threads
    for (int i = 0; i < M*N; i++) {
        pthread_join(add_threads[i], NULL);
        pthread_join(sub_threads[i], NULL);
        pthread_join(mul_threads[i], NULL);
    }

    // Print matrices
    printf("Matrix A:\n");
    print_matrix(A);
    printf("Matrix B:\n");
    print_matrix(B);
    printf("Matrix C (A + B):\n");
    print_matrix(C);
    printf("Matrix D (A - B):\n");
    print_matrix(D);
    printf("Matrix E (A * B):\n");
    print_matrix(E);



}
```