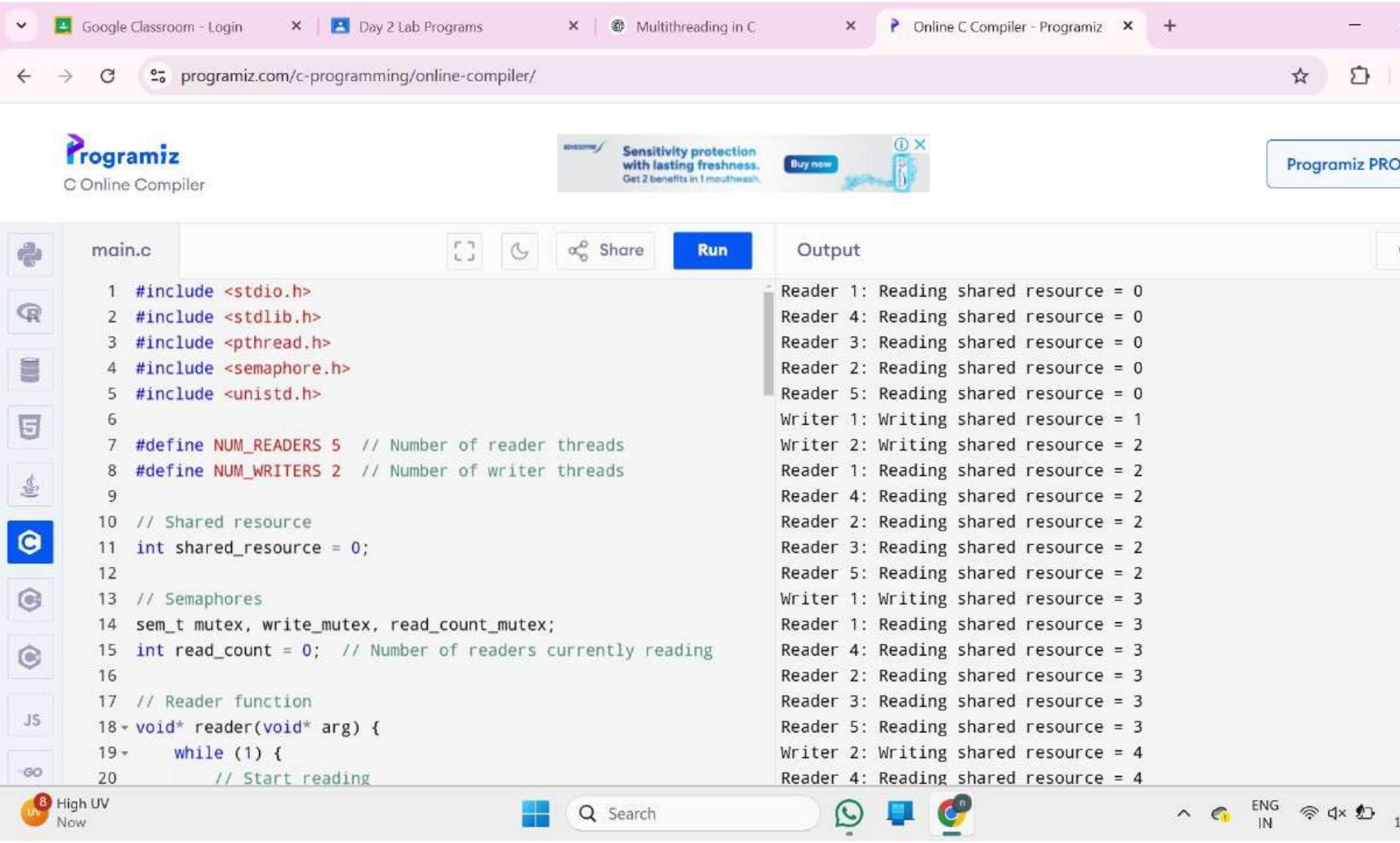
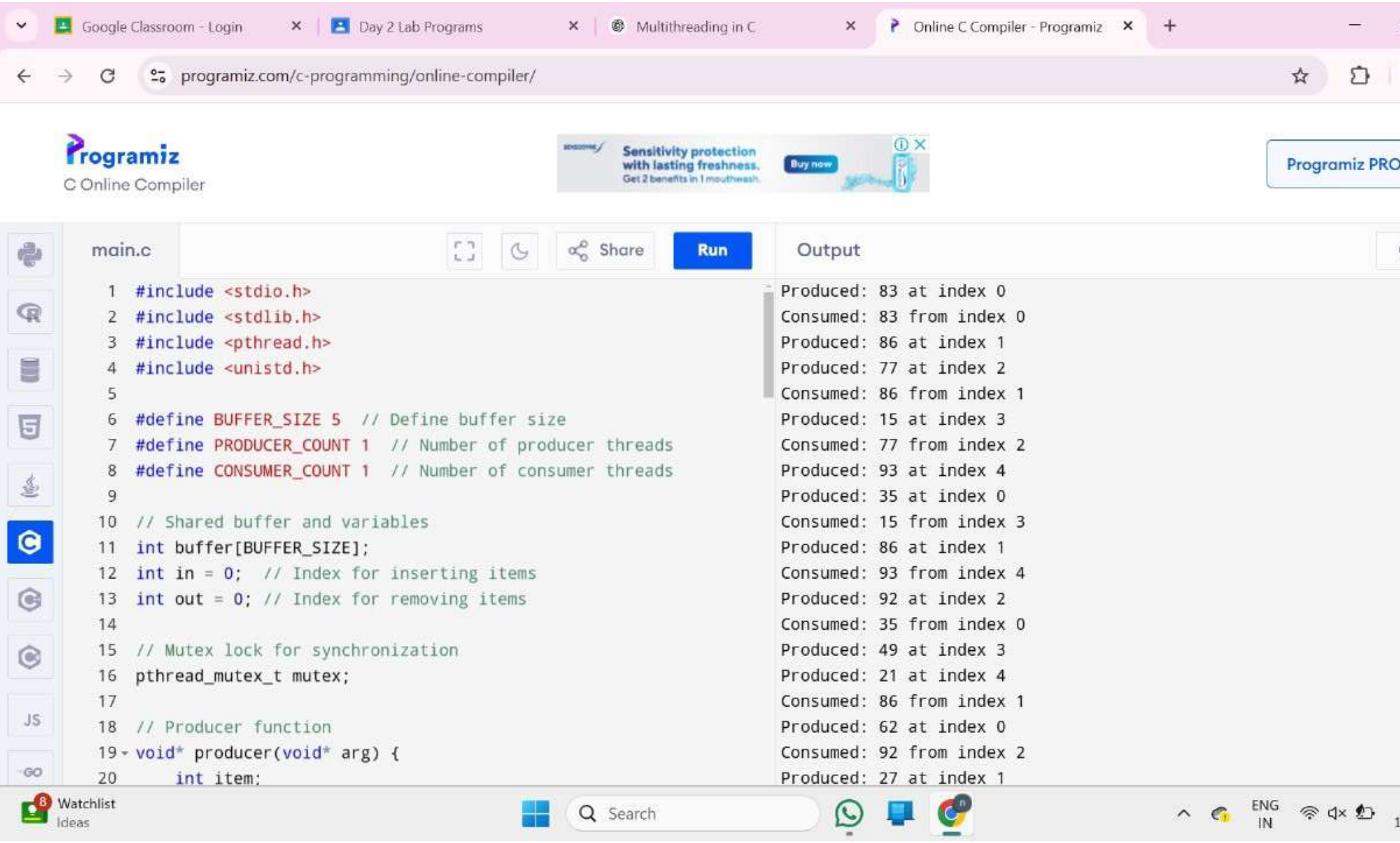

Google Classroom - Login
Day 2 Lab Programs
Multithreading in C
Online C Compiler - Programiz
programiz.com/c-programming/online-compiler/
Programiz
C Online Compiler
Sensitivity protection with lasting freshness.
Get 2 benefits in 1 mouthwash.
Buy now
Programiz PRO
main.c
Share
Run
Output
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <semaphore.h>
#include <unistd.h>
#define NUM_READERS 5 // Number of reader threads
#define NUM_WRITERS 2 // Number of writer threads
// Shared resource
int shared_resource = 0;
// Semaphores
sem_t mutex, write_mutex, read_count_mutex;
int read_count = 0; // Number of readers currently reading
// Reader function
void* reader(void* arg) {
while (1) {
// Start reading
Reader 1: Reading shared resource = 0
Reader 4: Reading shared resource = 0
Reader 3: Reading shared resource = 0
Reader 2: Reading shared resource = 0
Reader 5: Reading shared resource = 0
Writer 1: Writing shared resource = 1
Writer 2: Writing shared resource = 2
Reader 1: Reading shared resource = 2
Reader 4: Reading shared resource = 2
Reader 2: Reading shared resource = 2
Reader 3: Reading shared resource = 2
Reader 5: Reading shared resource = 2
Writer 1: Writing shared resource = 3
Reader 1: Reading shared resource = 3
Reader 4: Reading shared resource = 3
Reader 2: Reading shared resource = 3
Reader 3: Reading shared resource = 3
Reader 5: Reading shared resource = 3
Writer 2: Writing shared resource = 4
Reader 4: Reading shared resource = 4
High UV
Now
Search
ENG
IN


Google Classroom - Login
Day 2 Lab Programs
Multithreading in C
Online C Compiler - Programiz
programiz.com/c-programming/online-compiler/
Programiz
C Online Compiler
Sensitivity protection with lasting freshness.
Get 2 benefits in 1 mouthwash.
Buy now
Programiz PRO
main.c
Share
Run
Output
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <unistd.h>
#define BUFFER_SIZE 5 // Define buffer size
#define PRODUCER_COUNT 1 // Number of producer threads
#define CONSUMER_COUNT 1 // Number of consumer threads
// Shared buffer and variables
int buffer[BUFFER_SIZE];
int in = 0; // Index for inserting items
int out = 0; // Index for removing items
// Mutex lock for synchronization
pthread_mutex_t mutex;
// Producer function
void* producer(void* arg) {
int item;
Produced: 83 at index 0
Consumed: 83 from index 0
Produced: 86 at index 1
Produced: 77 at index 2
Consumed: 86 from index 1
Produced: 15 at index 3
Consumed: 77 from index 2
Produced: 93 at index 4
Produced: 35 at index 0
Consumed: 15 from index 3
Produced: 86 at index 1
Consumed: 93 from index 4
Produced: 92 at index 2
Consumed: 35 from index 0
Produced: 49 at index 3
Produced: 21 at index 4
Consumed: 86 from index 1
Produced: 62 at index 0
Consumed: 92 from index 2
Produced: 27 at index 1
Watchlist
Ideas
Search
ENG
IN

Google Classroom - Login | Day 2 Lab Programs | Multithreading in C | Online C Compiler - Programiz

programiz.com/c-programming/online-compiler/

Programiz
C Online Compiler

main.c

Share | Run

Output

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

#define BUFFER_SIZE 5  // Define buffer size
#define PRODUCER_COUNT 1  // Number of producer threads
#define CONSUMER_COUNT 1  // Number of consumer threads

// Shared buffer and variables
int buffer[BUFFER_SIZE];
int in = 0;  // Index for inserting items
int out = 0; // Index for removing items

// Semaphores
sem_t empty, full, mutex;

// Producer function
void* producer(void* arg) {
```

```
Produced: 83 at index 0
Consumed: 83 from index 0
Produced: 86 at index 1
Consumed: 86 from index 1
Produced: 77 at index 2
Produced: 15 at index 3
Consumed: 77 from index 2
Produced: 93 at index 4
Produced: 35 at index 0
Consumed: 15 from index 3
Produced: 86 at index 1
Produced: 92 at index 2
Consumed: 93 from index 4
Produced: 49 at index 3
```

Google Classroom - Login | Day 2 Lab Programs | Multithreading in C | Online C Compiler - Programiz

programiz.com/c-programming/online-compiler/

Programiz
C Online Compiler

Programiz PRO

main.c

Share | Run

Output

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

#define P 5  // Number of processes
#define R 3  // Number of resource types

// Function to calculate if the system is in a safe state
bool is_safe(int available[], int max[][R], int allot[][R],
     int need[][R]) {
     int work[R];
     bool finish[P] = {false};
     int safeSeq[P];
     int count = 0;

     // Initialize work[] as available[]
     for (int i = 0; i < R; i++) {
         work[i] = available[i];
     }

     // Try to find a safe sequence
```

```
A module you have imported isn't available at the moment. It wi
    available soon.
```

Google Classroom - Login | Day 2 Lab Programs | Multithreading in C | Online C Compiler - Programiz

programiz.com/c-programming/online-compiler/

Programiz
C Online Compiler

Programiz PRO

main.c | Share | Run | Output

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

#define FILENAME "day 4 programs.pdf"

// Define the employee structure
typedef struct {
    int id;
    char name[50];
    float salary;
} Employee;

// Function to add a new employee record to the file
void add_employee(FILE *file) {
    Employee emp;

    // Get employee details
    printf("Enter employee ID: ");
    scanf("%d", &emp.id);
    getchar(); // Consume newline character left by scanf
```

```
Unable to open file.


=== Code Exited With Errors ===
```

NIFTY -0.11%

Search

ENG IN

Google Classroom - Login | Day 2 Lab Programs | Multithreading in C | Online C Compiler - Programiz

programiz.com/c-programming/online-compiler/

Programiz
C Online Compiler

Programiz PRO

main.c

Share | Run

```
140             subdir_name[strcspn(subdir_name, "\n")] = '\0'
                    ;  // Remove newline character
141             printf("Enter filename to delete: ");
142             fgets(filename, sizeof(filename), stdin);
143             filename[strcspn(filename, "\n")] = '\0';  //
                    Remove newline character
144             delete_file_from_subdir(&dir, subdir_name,
                    filename);
145             break;
146         case 4:
147             printf("Enter subdirectory name to display: "
                    );
148             fgets(subdir_name, sizeof(subdir_name), stdin
                    );
149             subdir_name[strcspn(subdir_name, "\n")] = '\0'
                    ;  // Remove newline character
150             display_subdir(&dir, subdir_name);
151             break;
152         case 5:
153             printf("Exiting the program.\n");
```

Output

```
--- Two-Level Directory Management ---
1. Add subdirectory
2. Add file to subdirectory
3. Delete file from subdirectory
4. Display subdirectory contents
5. Exit
Enter your choice: 4
Enter subdirectory name to display: ghy
Subdirectory 'ghy' not found.

--- Two-Level Directory Management ---
1. Add subdirectory
2. Add file to subdirectory
3. Delete file from subdirectory
4. Display subdirectory contents
5. Exit
Enter your choice: 3
Enter subdirectory name to delete file from: xyz
Enter filename to delete: sdg
```

BSE smlcap -0.81%

Search

ENG IN

Google Classroom - Login | Day 2 Lab Programs | Multithreading in C | Online C Compiler - Programiz

programiz.com/c-programming/online-compiler/

Programiz
C Online Compiler

Programiz PRO

main.c | Share | Run | Output

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

#define MAX_FILES 10
#define FILENAME_LENGTH 50

// Structure to represent the directory
typedef struct {
    char filenames[MAX_FILES][FILENAME_LENGTH];
    int file_count;
} Directory;

// Function to initialize the directory
void init_directory(Directory* dir) {
    dir->file_count = 0;
}

// Function to add a file to the directory
void add_file(Directory* dir, const char* filename) {
    if (dir->file_count < MAX_FILES) {
```

```
--- Directory Management ---
1. Add file
2. Delete file
3. Display directory
4. Exit
Enter your choice: 2
Enter file name to delete: xyz
File 'xyz' not found in the directory.

--- Directory Management ---
1. Add file
2. Delete file
3. Display directory
4. Exit
Enter your choice:
```

500312
+1.11%

Search

ENG
IN

Google Classroom - Login | Day 2 Lab Programs | Multithreading in C | Online C Compiler - Programiz

programiz.com/c-programming/online-compiler/

Programiz
C Online Compiler

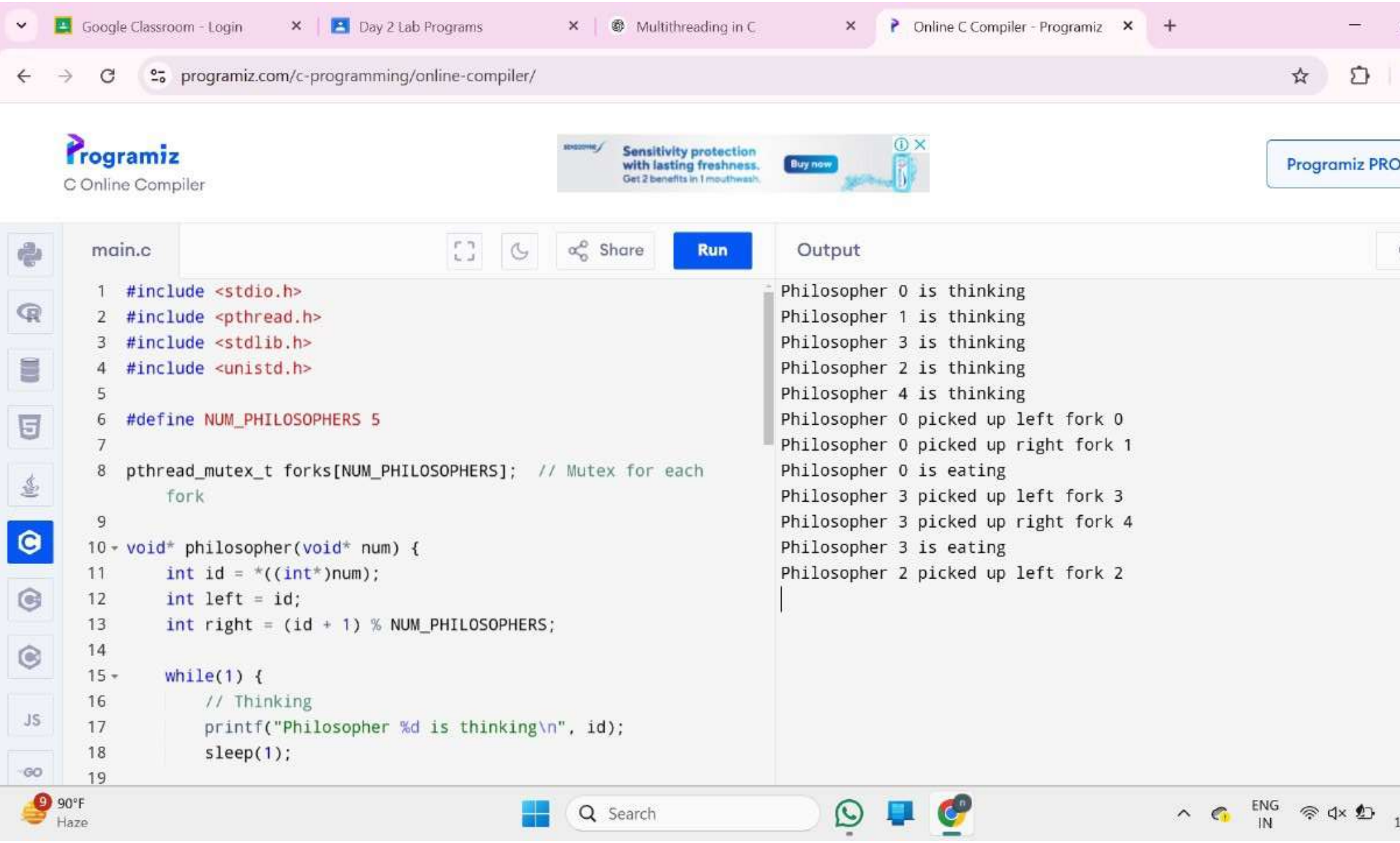
Programiz PRO

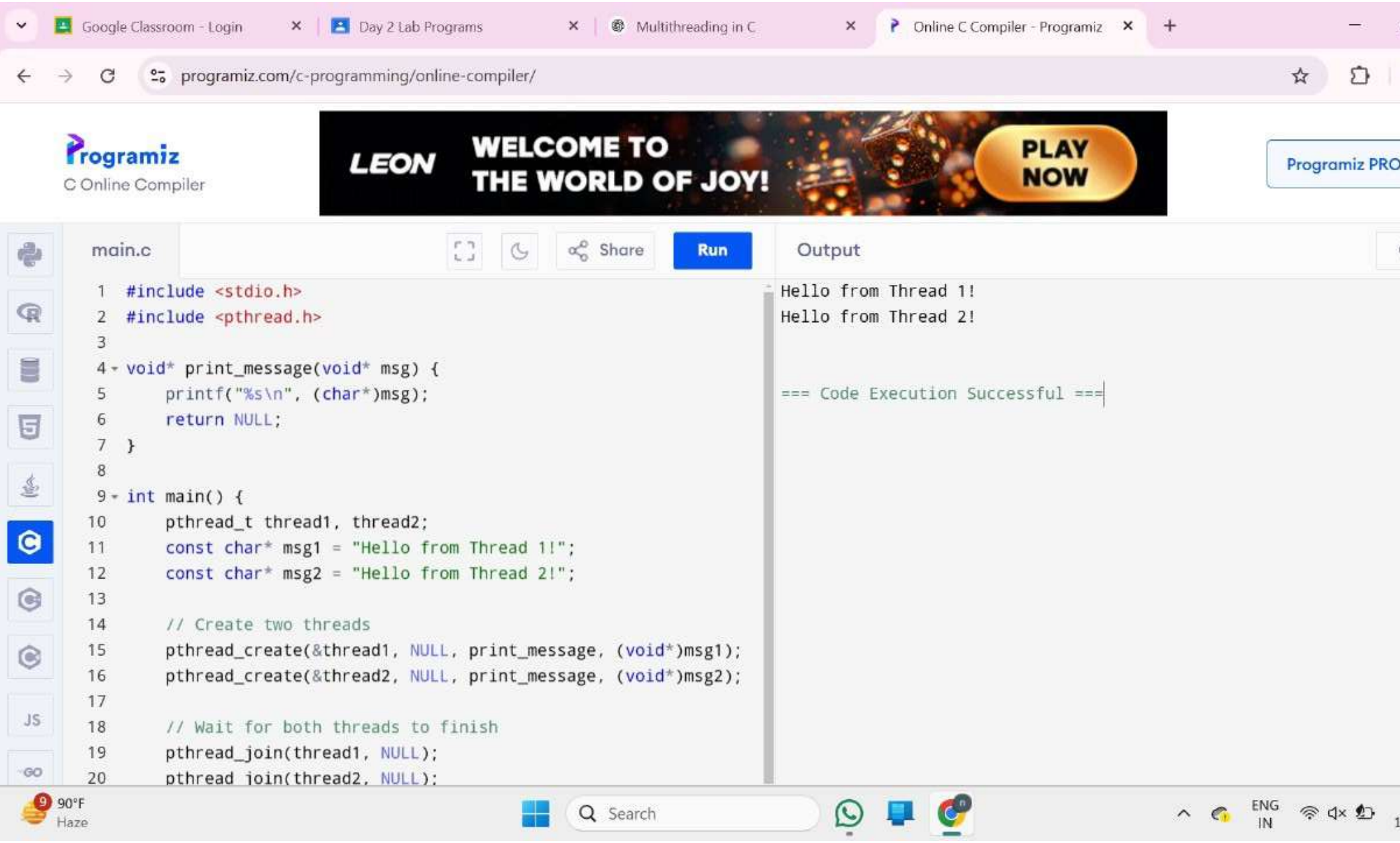
main.c | Share | Run | Output

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

#define MAX_BLOCKS 10

// Memory block structure
typedef struct {
    int size;  // Size of the block
    int is_free; // 1 if the block is free, 0 if it is allocated
} MemoryBlock;

// Memory pool
MemoryBlock memory[MAX_BLOCKS];

// Initialize memory pool
void initialize_memory() {
    for (int i = 0; i < MAX_BLOCKS; i++) {
        memory[i].size = rand() % 100 + 1;  // Assign random sizes
```

```
Memory Pool:
Block # | Size | Status
1       84      Free
2       87      Free
3       78      Free
4       16      Free
5       94      Free
6       36      Free
7       87      Free
8       93      Free
9       50      Free
10      22      Free


Memory allocation requests:
First Fit: Allocated block 1 of size 84
Memory Pool:
Block # | Size | Status
1       84      Allocated
2       87      Free
```

90°F Haze | Search | ENG IN


Google Classroom - Login
Day 2 Lab Programs
Multithreading in C
Online C Compiler - Programiz
programiz.com/c-programming/online-compiler/
Programiz
C Online Compiler
Sensitivity protection with lasting freshness.
Get 2 benefits in 1 mouthwash.
Buy now
Programiz PRO
main.c
Share
Run
Output
#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>
#include <unistd.h>
#define NUM_PHILOSOPHERS 5
pthread_mutex_t forks[NUM_PHILOSOPHERS]; // Mutex for each fork
void* philosopher(void* num) {
int id = *((int*)num);
int left = id;
int right = (id + 1) % NUM_PHILOSOPHERS;
while(1) {
// Thinking
printf("Philosopher %d is thinking\n", id);
sleep(1);
Philosopher 0 is thinking
Philosopher 1 is thinking
Philosopher 3 is thinking
Philosopher 2 is thinking
Philosopher 4 is thinking
Philosopher 0 picked up left fork 0
Philosopher 0 picked up right fork 1
Philosopher 0 is eating
Philosopher 3 picked up left fork 3
Philosopher 3 picked up right fork 4
Philosopher 3 is eating
Philosopher 2 picked up left fork 2
90°F
Haze
Search
ENG
IN


Google Classroom - Login
Day 2 Lab Programs
Multithreading in C
Online C Compiler - Programiz
programiz.com/c-programming/online-compiler/
Programiz
C Online Compiler
LEON
WELCOME TO
THE WORLD OF JOY!
PLAY
NOW
Programiz PRO
main.c
Share
Run
Output
#include <stdio.h>
#include <pthread.h>
void* print_message(void* msg) {
printf("%s\n", (char*)msg);
return NULL;
}
int main() {
pthread_t thread1, thread2;
const char* msg1 = "Hello from Thread 1!";
const char* msg2 = "Hello from Thread 2!";
// Create two threads
pthread_create(&thread1, NULL, print_message, (void*)msg1);
pthread_create(&thread2, NULL, print_message, (void*)msg2);
// Wait for both threads to finish
pthread_join(thread1, NULL);
pthread_join(thread2, NULL);
Hello from Thread 1!
Hello from Thread 2!
=== Code Execution Successful ===
90°F
Haze
Search
ENG
IN