1. Create the Header File (`mathfun.h`):

C

```c
#ifndef MATHFUN_H
#define MATHFUN_H

int check_prime(int data);
int test_digit(int data, int digit); // Accepts digit to check
int test_ascend(int data);
int test_descend(int data);

#endif
```

This header declares the functions John wants to implement in the library, making them accessible to programs that use the library.

2. Implement the Library Functions (`mathfun.c`):

```c
#include "mathfun.h"
#include <stdbool.h> // For boolean data type

bool is_divisible(int num, int divisor) {
    return num % divisor == 0;
}

int check_prime(int data) {
    if (data <= 1) return 0; // 1 or less are not prime
    for (int i = 2; i * i <= data; i++) {
        if (is_divisible(data, i)) return 0; // Divisible by a number > 1, not
prime
    }
    return 1; // Prime number
}

//check for  a particluar digit(eg:3)
int test_digit(int data, int digit) {
    while (data > 0) {
        if (data % 10 == digit) return 1; // Digit found
        data /= 10;
    }
    return 0; // Digit not found
}

//verifying digits are ascending order
```

```c
int test_ascend(int data) {
    int prev_digit = data % 10;
    data /= 10;
    while (data > 0) {
        int curr_digit = data % 10;
        if (curr_digit > prev_digit) return 0; // Digits not ascending
        prev_digit = curr_digit;
        data /= 10;
    }
    return 1; // Digits are ascending
}

//verifying digits are descending order
int test_descend(int data) {
    int prev_digit = data % 10;
    data /= 10;
    while (data > 0) {
        int curr_digit = data % 10;
        if (curr_digit < prev_digit) return 0; // Digits not descending
        prev_digit = curr_digit;
        data /= 10;
    }
    return 1; // Digits are descending
}
```

3. Build the Dynamic Library (`libmathfun.so`):

- **Windows (using MinGW):**

  Bash

  ```
  gcc -c -fpic mathfun.c -o mathfun.o
  gcc -shared -o libmathfun.so mathfun.o
  ```

The `-fPIC` flag ensures position-independent code (PIC) for dynamic linking. The `-shared` flag creates a shared library.

4. Use the Library in Your Program (`PrimeNumberWithLibrary.c`):

C

```c
#include <stdio.h>
#include "mathfun.h" // Include the header file

int main() {
```

```
    int lower = 100, upper = 1000;
    printf("Output: ");
    for (int num = lower; num <= upper; num++) {
        if (check_prime(num) && test_digit(num, 3) && (test_ascend(num) ||
test_descend(num))) {
            printf("%d ", num);
        }
    }
    printf("\n");
    return 0;
}
```

5. Link the Library During Compilation:

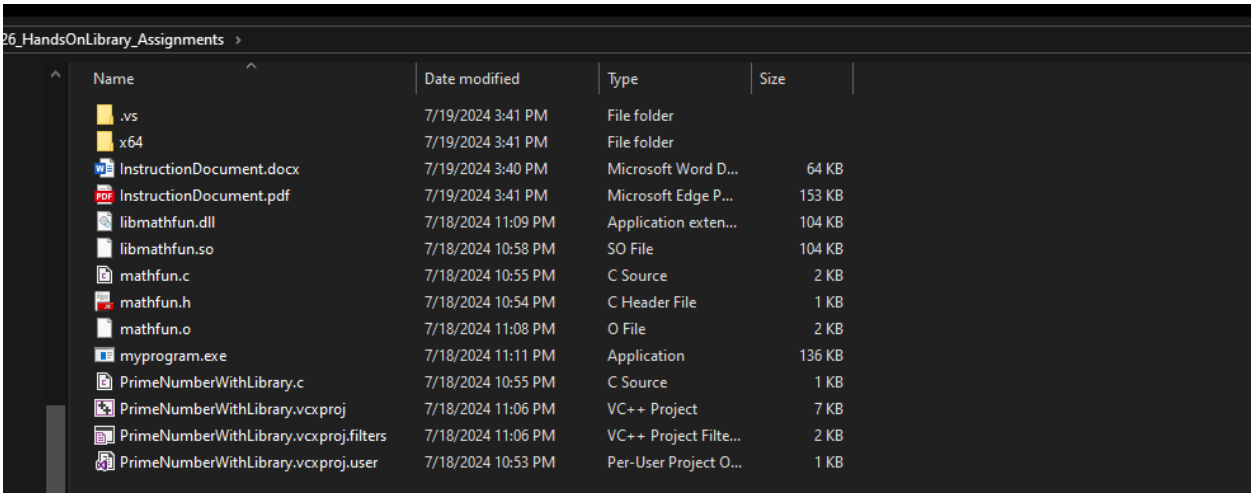
- **Windows (using MinGW):**

  Bash

  ```
  gcc PrimeNumberWithLibrary.c -L. -lmathfun -o
  PrimeNumberWithLibrary.exe
  ```

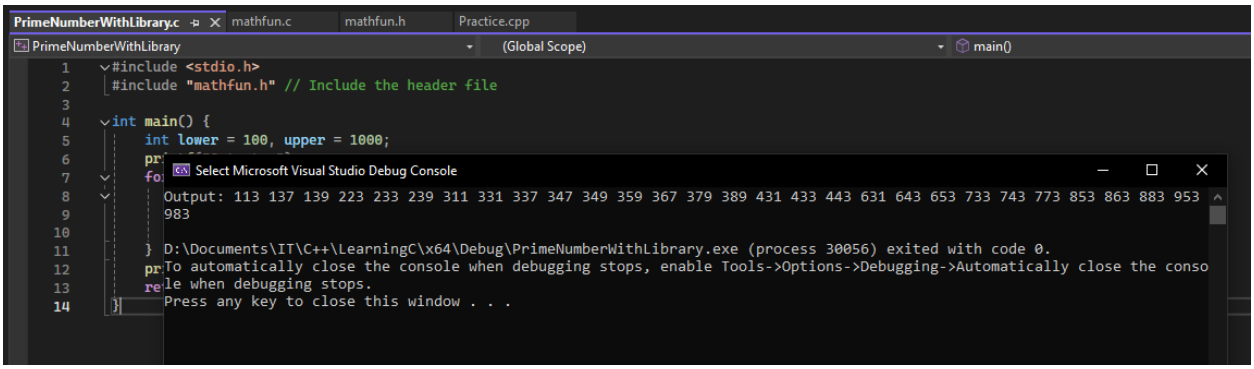
The `-L.` specifies the directory containing the library (`libmathfun.so` or `libmathfun.dll`). The `-lmathfun` links against the library name without the `.so` or `.dll` extension.

Source code after completed



Running the Program:

Compile


PrimeNumberWithLibrary.c
mathfun.c
mathfun.h
Practice.cpp
PrimeNumberWithLibrary
(Global Scope)
main()
#include <stdio.h>
#include "mathfun.h" // Include the header file
int main() {
int lower = 100, upper = 1000;
Select Microsoft Visual Studio Debug Console
Output: 113 137 139 223 233 239 311 331 337 347 349 359 367 379 389 431 433 443 631 643 653 733 743 773 853 863 883 953
983
D:\Documents\IT\C++\LearningC\x64\Debug\PrimeNumberWithLibrary.exe (process 30056) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .