

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

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):

```
#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
```

```

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

```

#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

| 26_HandsOnLibrary_Assignments > | | | | |
|--|--------------------|-----------------------|--------|--|
| Name | Date modified | Type | Size | |
| .vs | 7/19/2024 3:41 PM | File folder | | |
| x64 | 7/19/2024 3:41 PM | File folder | | |
| InstructionDocument.docx | 7/19/2024 3:40 PM | Microsoft Word D... | 64 KB | |
| InstructionDocument.pdf | 7/19/2024 3:41 PM | Microsoft Edge P... | 153 KB | |
| libmathfun.dll | 7/18/2024 11:09 PM | Application exten... | 104 KB | |
| libmathfun.so | 7/18/2024 10:58 PM | SO File | 104 KB | |
| mathfun.c | 7/18/2024 10:55 PM | C Source | 2 KB | |
| mathfun.h | 7/18/2024 10:54 PM | C Header File | 1 KB | |
| mathfun.o | 7/18/2024 11:08 PM | O File | 2 KB | |
| myprogram.exe | 7/18/2024 11:11 PM | Application | 136 KB | |
| PrimeNumberWithLibrary.c | 7/18/2024 10:55 PM | C Source | 1 KB | |
| PrimeNumberWithLibrary.vcxproj | 7/18/2024 11:06 PM | VC++ Project | 7 KB | |
| PrimeNumberWithLibrary.vcxproj.filters | 7/18/2024 11:06 PM | VC++ Project Filte... | 2 KB | |
| PrimeNumberWithLibrary.vcxproj.user | 7/18/2024 10:53 PM | Per-User Project O... | 1 KB | |

Running the Program:

Compile

```
1  #include <stdio.h>
2  #include "mathfun.h" // Include the header file
3
4  int main() {
5      int lower = 100, upper = 1000;
6      printf("Primes between %d and %d are: ", lower, upper);
7      for (int i = lower; i <= upper; i++) {
8          if (isPrime(i)) {
9              printf("%d ", i);
10             if (i % 10 == 0) printf("\n");
11         }
12     }
13     printf("\n");
14 }
```

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\Debug\PrimeNumberWithLibrary.exe (process 30056) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .