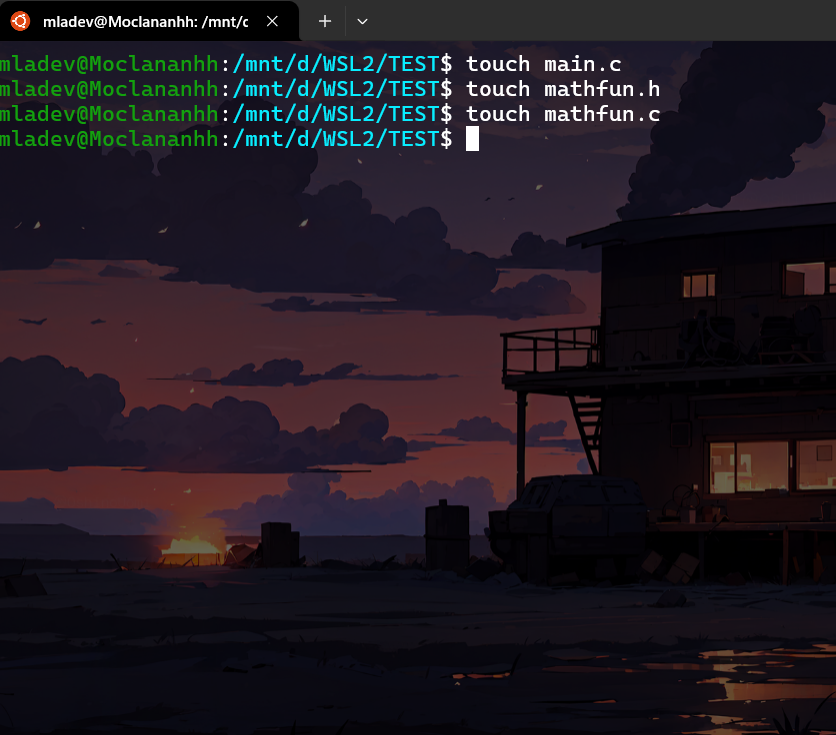
### Hands on Library Author: ThanhTH10

## Create file



## mathfun.h

#ifndef MATHFUN\_H

#define MATHFUN\_H

int check\_prime(int data);

int test\_digit(int data);

int test\_ascend(int data);

int test\_descend(int data);

#endif

## Mathfun.c

#include "mathfun.h"

#include <stdbool.h>

int check\_prime(int data)

{

    if (data <= 1)

        return 0; // 1 and below are not prime

    for (int i = 2; i \* i <= data; i++)

    {

        if (data % i == 0)

            return 0; // If divisible, not prime

    }

    return 1; // If no divisors found, it's prime

}

int test\_digit(int data)

{

    while (data > 0)

    {

        if (data % 10 == 3)

            return 1; // If any digit is 3, return true

        data /= 10;

    }

    return 0; // If no 3 found, return false

}

int test\_descend(int data)

{

    int last\_digit = -1; // Start with -1 as no digit can be smaller

    while (data > 0)

    {

        int current\_digit = data % 10;

        if (current\_digit < last\_digit)

            return 0; // If not descending or equal, return false

        last\_digit = current\_digit;

        data /= 10;

    }

    return 1; // If all digits were descending or equal, return true

}

int test\_ascend(int data)

{

    int last\_digit = 10; // Start with 10 as no digit can be larger

    while (data > 0)

    {

        int current\_digit = data % 10;

        if (current\_digit > last\_digit)

            return 0; // If not ascending or equal, return false

        last\_digit = current\_digit;

        data /= 10;

    }

    return 1; // If all digits were ascending or equal, return true

}

## Main.c

int main()

{

    printf("Prime numbers between 100 and 1000 that contain the digit 3 and have digits in non-descending or non-ascending order:\n");

    int first = 1;

    for (int i = 100; i <= 1000; i++)

    {

        if (check\_prime(i) && test\_digit(i) && (test\_ascend(i) || test\_descend(i)))

        {

            if (!first)

            {

                printf(",");

            }

            printf("%d", i);

            first = 0;

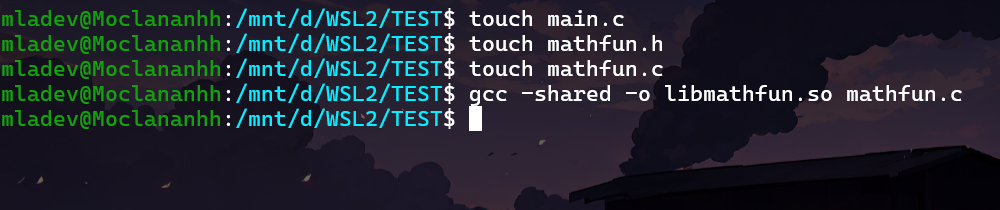
        }

    }

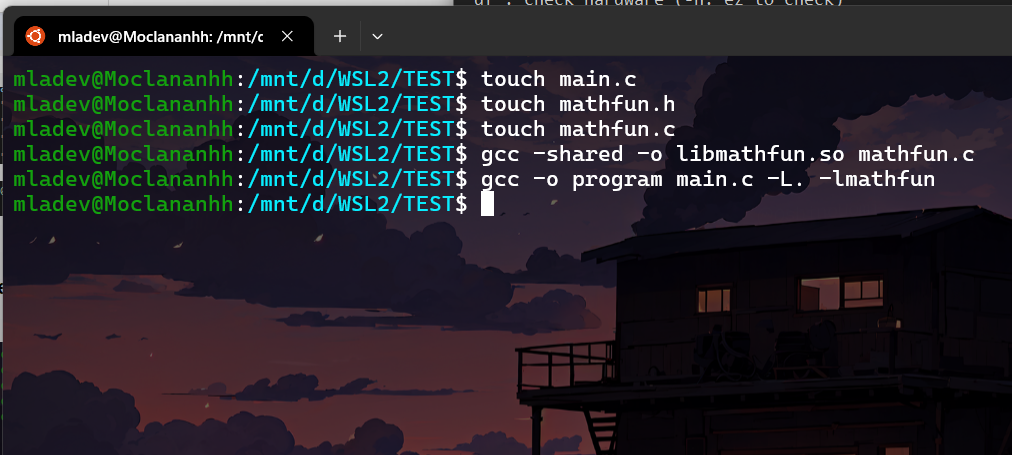
    return 0;

}

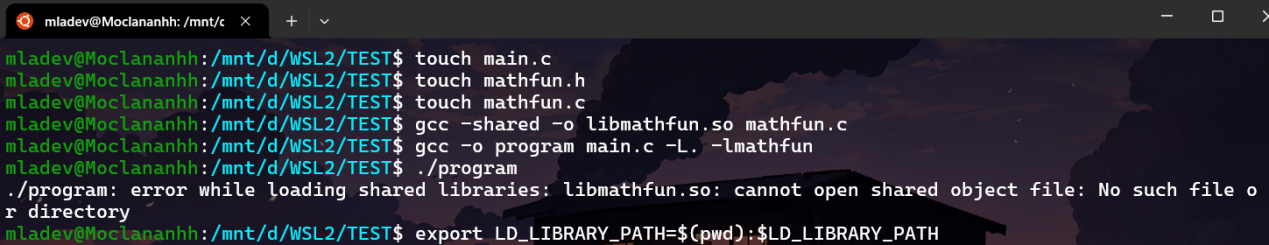
## Compile the lib.c file into a shared library:



## Compile the main.c file and link it with the shared library



## Export Loading share library



## Run program:

