

CSCI2100B Data Structures (Fall 2023)

Lab Assignment #0	
Schedule:	Week 2
<ol style="list-style-type: none">1. To familiarize with the VM platform and the submission system2. To revise basic C programming	

Submission guideline (Online judge)

1. Name your program(s) according to the question (e.g. lab0-q1.c)
2. Write your programs under "Desktop/mywork/lab0"
3. Duplicate your programs before submission
4. Submit your programs under "Desktop/submission/lab0"
5. Check the submission folder for your grading report (Be patience, auto grading need some time)
6. Resubmit your work if you cannot receive full score
7. Visit "Other Locations/Computer/2100share/lab0" in the VM to download the starting programs
8. Version of the gcc compiler in the online judge: C99
9. Deadline: 19th September, 2023 (Tuesday)
10. Lab 0 is worth 1 attendance point

Useful commands

1. Compile your program

```
gcc -o prog labX-q1.c labX-q1-tc1.c -lm -std=c99
gcc -o prog labX-q3.c -lm -std=c99
```
2. Run the program with arguments

```
Test case (.c)
./prog outputfile.txt
Test case (.in)
./prog inputfile.txt outputfile.txt
```

Question 1: Hello World! (1 attendance record)

Write a program, only print a line "Hello World!", to the given file in the argument list. First file is an empty file, which is redundant. Second file is the output file. Name your program as "lab0-q1.c".

Sample Output:

Hello World!

Question 2: Two Sum

Given an array of integers `nums` and an integer `target`, return indices of the two numbers such that they add up to `target`. You may assume that each input would have exactly one solution, and you may not use the same element twice. You can return the answer in any order. Name your program as "lab0-q2.c".

Example 1:

Input:

4

2 7 11 15

9

Output:

0 1

Explanation:

A sequence of 4 numbers is given.

They are 2, 7, 11, and 15.

The target of this sequence is 9.

The indices are 0 and 1.

Constraints:

$2 \leq \text{nums.length} \leq 10^4$

$-10^9 \leq \text{nums}[i] \leq 10^9$

$-10^9 \leq \text{target} \leq 10^9$

Only one valid answer exists.

Follow-up: Can you come up with an algorithm that is less than $O(n^2)$ time complexity?

Question 3: Reverse a number (0%)

Write a program to reverse a non-negative number. For example, the reverse of 12345 is 54321. Your program should read the input from the file, and output the answer to another file. The first argument is the input file name, while the second argument is the output file name. Name your program as "lab0-q3.c".

Input file: A line contains an integer.

Output file: A line contains only 1 integer, the reverse of the input number

Sample Input:

123456

Sample Output:

654321

Question 4: Is Prime Again (0%)

You are going to implement the content for the `isprime()` function prototype defined below. Read the comments carefully. The header file "lab0-q4.h" and the description of the function are as follows.

Your program should not contain `main()`. Name your program as "lab0-q4.c".

```
// lab0-q4.h
// Cannot modify this file

// return 0 for normal use
// return -1 for negative input
// *result is 1, if input is prime
// *result is 0, if input is not prime or abnormal use
int isprime(int input, int *result);
```

Sample Test Case (lab0-q4-tc1.c):

```
// This file is named as lab0-q4-tc1.c
#include "lab0-q4.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <limits.h>

int main(int argc, char *argv[]){

    FILE *fout = fopen(argv[1], "w");
    int value1, value2, value3;
    int ans1, ans3;

    value2 = 0; // input is 0
    ans1 = 0;   // 0 is not negative
    ans3 = 0;   // 0 is not prime

    value1 = isprime(value2, &value3);

    // output "Correct" if isprime() is correctly implemented
    if (value1 != ans1 || value3 != ans3) {
        fprintf(fout, "Prime: Wrong Answer\n");
    } else {
        fprintf(fout, "Prime: Correct\n");
    }
    fclose(fout);

    return 0;
}
```