

COMP281 2021-22 – Assignment 1

14 Feb 2022

- In the following, you will find the problems that constitute Assignment 1. They will be also available on the *online judge (OJ)* system available at <https://student.csc.liv.ac.uk/JudgeOnline>. Please note that it is possible some of the problem details on Online Judge vary slightly from those provided in this document. In such cases, follow the details from this document.
- You need to write a C program (not C++ or C#) that solves each problem – it must read the input, as specified in the problem description then print the solution to the given problem for that input.
 - Note that code is “correct” only if it **correctly implements a solution to the problem stated** in the assignment, not “if online judge accepts it”.
 - That is, even if OJ accepts your code, it could be wrong. Read the problems carefully.
- Input is read from the standard input, in the same way that you read input from the keyboard as shown in lectures (e.g., using `scanf`). Output is also printed to the standard output, as you have seen (e.g., using `printf`).
- For this set of problems, you must not use C’s string handling library `string.h`
- When you are satisfied that your C programs work correctly, you must submit them through the departmental submission system, SAM.
 - Even if the program is not correct, still submit whatever you have! You can still earn points if certain parts of the code are done right.
- You must also include a brief report describing your solutions to the problems. This should be maximum two sides of A4 paper and should give a description of how each of your solutions works. This should include describing the algorithm used to reach the solution, describing your use of any C language features (that were not discussed in lectures) and identifying any resources that you have used to help you solve the problems.
- This assignment is worth 50% of the total mark for COMP281.
 - All five problems in this assignment are weighted equally.
 - For each problem, you can earn a total of 20 points
 - 10 points for “Functionality and Correctness” awarded for programs that **correctly** solve the problem for all test cases.
 - 8 points for “Programming style, use of comments, indentation and identifiers” awarded depending on the style, comments, efficiency of the solution and use of appropriately named variables etc.
 - 2 points for the quality and depth of the accompanying report
 - The final grade results from normalising the earned points to a scale of 100.
 - See separate “comp281-detailed-marking-guidelines.pdf” for more details.

Submission Instructions

- Create a folder, and name it using your Student ID and User ID, e.g. 201234567_sgpj6
- In the folder, there should be 6 files:
 - 1 report file, in PDF format. Name it with your Student ID and User ID, e.g. 201234567_sgpj6.pdf
 - 5 source code files. Name each using the Problem Number, e.g. 1014.c
 - In your source code, include your Student Info and Problem Info, e.g.:

```
/*  
 * Student ID: 201234567  
 * Student Name: Phil Jimmieson  
 * Email: phil.jimmieson@student.liverpool.ac.uk  
 *  
 * User: sgpj6  
 *  
 * Problem ID: 1014  
 * RunID: 22456  
 * Result: Accepted  
 */
```
 - The OJ provides a RunID, which is different from the Problem ID.
 - The Result is one of the following: Accepted, Wrong Answer, Presentation Error, Time Limit Exceeded, Memory Limit Exceeded, Output Limit Exceeded, Runtime Error, Compile Error.
- Compress the folder into a single zip file, and name it as, e.g. 201234567_sgpj6.zip
 - Use the standard zip file format:
which is supported by winzip, winrar, etc. on Windows. On macOS use 'compress'. Use 'zip' on Linux
 - Test your zip file before submitting to ensure you have submitted all the relevant files.
- Submit this zip file using the departmental submission system at https://sam.csc.liv.ac.uk/COMP/CW_Submissions.pl
Only the file submitted through this link will be marked.
- The **deadline** for this assignment submission is **25-Feb-2022 at 12:00**.
- Penalties for late submission apply in accordance with departmental policy as set out in the student handbook, which can be found at: <http://intranet.csc.liv.ac.uk/student/ug-handbook.pdf>

1. Problem 1014

Title: Area and circumference of circles

Description

In this exercise you have to compute the area and circumference of a series of circles and output their sum. Specifically, the program will take the radius of two circles as input ($r_1 \leq r_2$, both integers) and will output the sum of the areas and the circumferences of all circles starting with r_1 and increasing at each step the radius by '1' until radius r_2 has been reached. As an example, suppose ' $r_2 - r_1 = 2$ ' then the program has to compute the sum of the areas and circumferences of three circles with radii $r_1, r_1 + 1, r_2$.

Remember that the area of a circle equals $\pi * r^2$ and the circumference equals $2\pi * r$.

Set π to 3.14

Input

Two integers r_1 and r_2 with $r_1 \leq r_2$.

Output

Two floats, sumofareas and sumofcircumferences.

The result should be to 3 digits precision.

Sample Input

```
3 4
```

Sample Output

```
78.500
```

```
43.960
```

2. Problem 1018

Title: Reverse String

Description

Reverse a string, without using an array for storage purposes. The length of the string is at most 100. Do not use any functions from string.h

Input

One string (letters and spaces).

Output

Reversed string

Sample Input

```
I am a student
```

Sample Output

```
tneduts a ma I
```

HINT

Recursion!

3. Problem 1022

Title: Count characters in a string

Description

Input a string. Count the numbers of 1) English characters; 2) digits; 3) spaces; 4) other characters. Note: Do not use the functions provided in string.h

Input

String

Output

number_of_english_characters number_of_digits number_of_spaces number_of_other_characters

Sample Input

```
aklsjflj123 sadf918u324 asdf91u32oasdf/.';123
```

Sample Output

```
23 16 2 4
```

4. Problem 1030

Title: Precise division

Description

$8/13=0.615384615384615384615384\dots$

For $8/13$, the 6-th digit after the decimal point is 4.

Given three positive integers a , b , and n (all at most 60000), you are asked to compute a/b and print out the n -th digit after the decimal point.

Input

a b n

Output

The n -th digit after the decimal point of a/b .

Sample Input

```
8 13 6
```

Sample Output

```
4
```

5. Problem 1032

Title: String Search

Description

Input three strings s_1 , s_2 , and s_3 (one for each line) of max length 50 characters. They may contain whitespace. Output the number of times s_2 and s_3 occur as a substring of s_1 .

For example, if the input strings are "hello world", "europe" and "wor", then the output should be "0 1". As another example, if the input strings are "hello world hello", "hello world" and "hel", then the output should be "1 2". If the input strings are "hello world", "hello" and "helo", then the output should be "1 0".

You are not allowed to use system functions defined in `string.h`.

Input

s_1 , s_2 , s_3 (each of max length 50 chars).

Output

Two integers.

Sample Input

```
Hello World! Hello
Hello
World
```

Sample Output

```
2 1
```