

Contents:

- A. The Task
 - B. Sample Output
 - C. Important Points / Hints
 - D. Submission Guidelines
-

A. The Task

We have studied in the class how to use the `write()` system call API to print strings on the terminal. In this assignment, you are required to develop a very simple C program that asks the user to enter two integers and then prints their sum on the screen. However, you are not allowed to use any library calls in the program. You are allowed to call only two built-in functions:

1. The `read()` system call API to get user inputs
2. The `write()` system call API to print output on terminal

B. Sample Output

```
Enter the first number: 12
Enter the second number: 13
The sum of 12 and 13 is 25
```

C. Important Points / Hints

CAREFULLY READ THE FOLLOWING POINTS BEFORE SUBMISSION

1. The input / output must follow the format shown in the sample output above.
2. You are required to provide all the code inside the `main()` function. You are not allowed to define any other functions.
3. You are allowed to call only `read()` and `write()` system call APIs – no other functions can be called whatsoever.
4. With `read()`, you will receive input in a char-type buffer (array). This function also returns the number of characters actually entered by the user. Under normal circumstances, we use the library call `atoi()` to convert a string to an integer. However, you are not allowed to use this function. You will have to write the code for this conversion yourselves.
5. With `write()`, you provide output in the form of a char-type buffer. In this program, we need to build a formatted string containing the three integer values to display the final output. Under normal circumstances, we use the library call `sprintf()` to build such strings. However, you are not allowed to use this function. You will have to write the code for this conversion yourselves.
6. For your convenience, a simple C program is provided below. It shows how to `echo` the input received through the keyboard. Note that this program is only intended to show you how to use the `read()` and `write()` calls.

Helper Code

```
#include <unistd.h>
int main()
{
    char buf[100];
    write(STDOUT_FILENO, "Enter something: ", 17);
    int n = read(STDIN_FILENO, buf, 100); // n = number of characters entered
    write(STDOUT_FILENO, "You entered: ", 13);
    write(STDOUT_FILENO, buf, n);
    return 0;
}
```

Sample Output

```
Enter something: Hello World
You entered: Hello World
```

D. Submission Guidelines

Submission Deadline:

Tuesday, 17th February 2026 before midnight

Submission will be closed after the deadline.

CAREFULLY READ THE FOLLOWING POINTS BEFORE SUBMISSION

Important:

1. The assignment **must be uploaded** to Assignment 1 created in your **respective Google Classroom** (GCR).
 2. You are required to upload your source file, which must be named as "**RollNo_Name_Program_Section.c**", e.g., "**24P-1234_UsmanAliShah_BSCS_1A.c**". Non-compliance with the naming convention or file format will result in the rejection of your assignment.
 3. Make sure that the source file is completely free of errors. No marks will be awarded if your source file fails to compile with gcc in Ubuntu.
 4. No submission other than GCR will be accepted. **Sending assignments via email and Whatsapp messages is strictly prohibited.**
 5. The source file must begin with comments mentioning your full name, roll number and section.
 6. The source file must be well-documented using appropriate comments explaining each major operation performed in the source code.
 7. **Cheating/copying/sharing in any form, including help from tools like ChatGPT, will result in the rejection of your assignment.**
 8. Assignment **MUST** be uploaded before the deadline. A penalty of **Two Marks Per Day** will be awarded with each day over the deadline.
-