ADVANCED C PROGRAMMING – MODULE 3 ASSIGNMENT

1. Which signals are triggered, when the following actions are performed?

1. user press ctrl+C - SIGINT signal.

Explanation: When Ctrl+C key combination is pressed, SIGINT signal is sent. It will terminate the process by default.

1. kill() system call is invoked - SIGTERM

Explanation: when SIGTERM is triggered, it can be blocked, handled or ignored.

1. CPU tried to execute an illegal instruction –SIGILL

Explanation: this signal is triggered when the program is trying to execute an illegal or privileged instruction. This can happen when the executable file is corrupted.

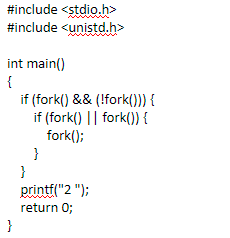
1. When the program access the unassigned memory – SIGSEGV

Explanation: SIGSEGV (signal segmentation violation) is triggered when a program tries to access an inaccessible memory location.

2. List the gdb command for the following operations

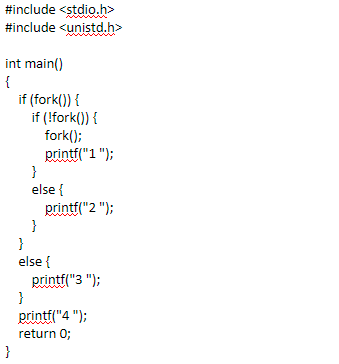
1. To run the current executable file – run [args]
2. To create breakpoints at – break [func\_name]
3. To resume execution once after breakpoint – continue
4. To clear break point created for a function – clear [func\_name]
5. Print the parameters of the function in the backtrace – info args

3. Guess the output for the following program.



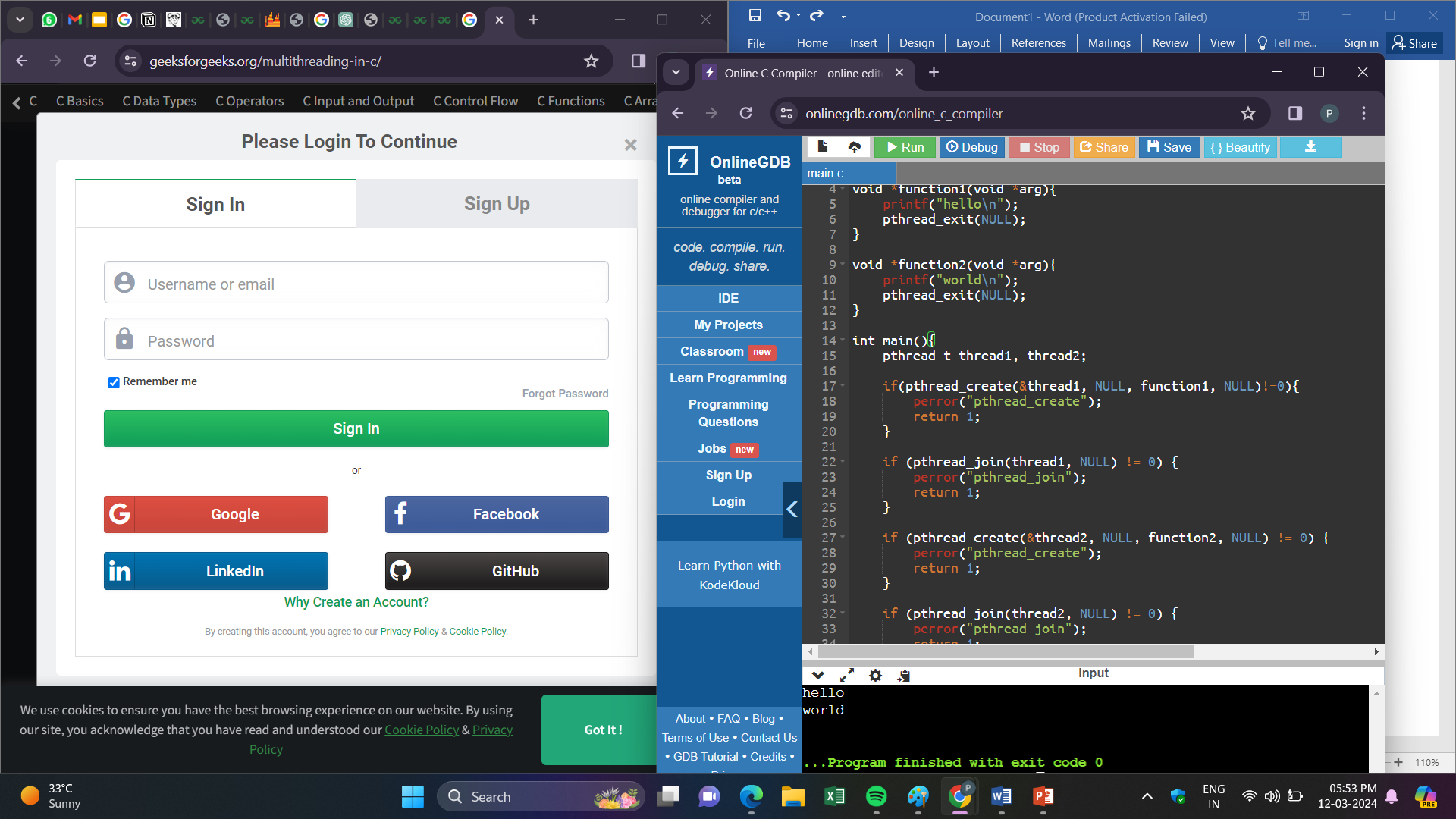
OUTPUT: 2

4. Guess the output for the following program.



OUTPUT: 2 4

5. Create two thread functions to print hello and world separately and create threads for each and execute them one after other in C



6. How to avoid Race conditions and deadlocks?

* To avoid race conditions, we can use mutexes, semaphores etc. Mutex is a locking mechanism used to make sure that only one thread can access a shared resource at a particular time. Semaphores are variables that control and coordinate the activities of multiple threads.
* To avoid deadlocks, we can implement timeout mechanisms. Also we have to avoid circular dependencies.

7. What is the difference between exec and fork?

|  |  |  |
| --- | --- | --- |
| S.No | Exec() | Fork() |
|  | Once exec() is called, a child process is created but the parent process doesn’t exist as it is replaced by the child process. | The fork() creates another child process other than parent process and both exist. |
|  | As the created child process replaces the parent process, both will have same address. | As both parent and child process exist after fork(), they will be stored in different addresses in the memory. |

8. What is the difference between process and threads?

|  |  |  |
| --- | --- | --- |
| S.No | Process | Threads |
|  | When we execute a program, it is termed as a ‘process’. | Threads are small segments of a process. |
|  | Context switching takes more time. | Context switching takes less time. |
|  | Other processes will not be affected if one of the process is obstructed or terminated. | Other threads will be affected if one thread is obstructed. |

9. Write a C program to demonstrate the use of Mutexes in threads synchronization.

