**Inclass 3**

Q1. A B B

fork() create a child process, duplicate the parent process and run both together at the same time

Q2. A B B A B B

4 processes.

Q3. 1039

It’s the process ID of the child process

Q4. The output is same in every run as the OS handle the program in the same way. The prompt comes back before the input because the parent process terminates before the child process.

Q5. The output is not same. They run child process first then run parent process. The value of ret is 0, it represents the exit status of the child process.

Q6. It starts with new child process then goes to parent process and keep looping till it reaches the starting process.

**Inclass 4**

Q1. Process get terminates.

Q2. Nothing happened. In the program we loaded “new\_action” and passed it to “sigaction” call. However, here we avoided SIG\_IGN specifies that the signal should be ignored.

Q3. Nothing happened.

Q4. Terminated.

Q5. The signal number is 2 and when we press CTRL + C it executes the handler function that we created and print prompt.

Q6. The process got suspended and the name is SIGTSTP.

**Inclass 5**

Q1. This program count number of threads that able to synchronize.

Q2.

* 987
* 995
* 993
* 999
* 985
* 997
* 989
* 993
* 991
* 998

Q3. It occurs when many threads try to run in parallel order. Making their data race each other to finish before each other.

Mutex is used to protect data or other resources from concurrent access, it has attributes, which specify its own characteristics.

Q4.