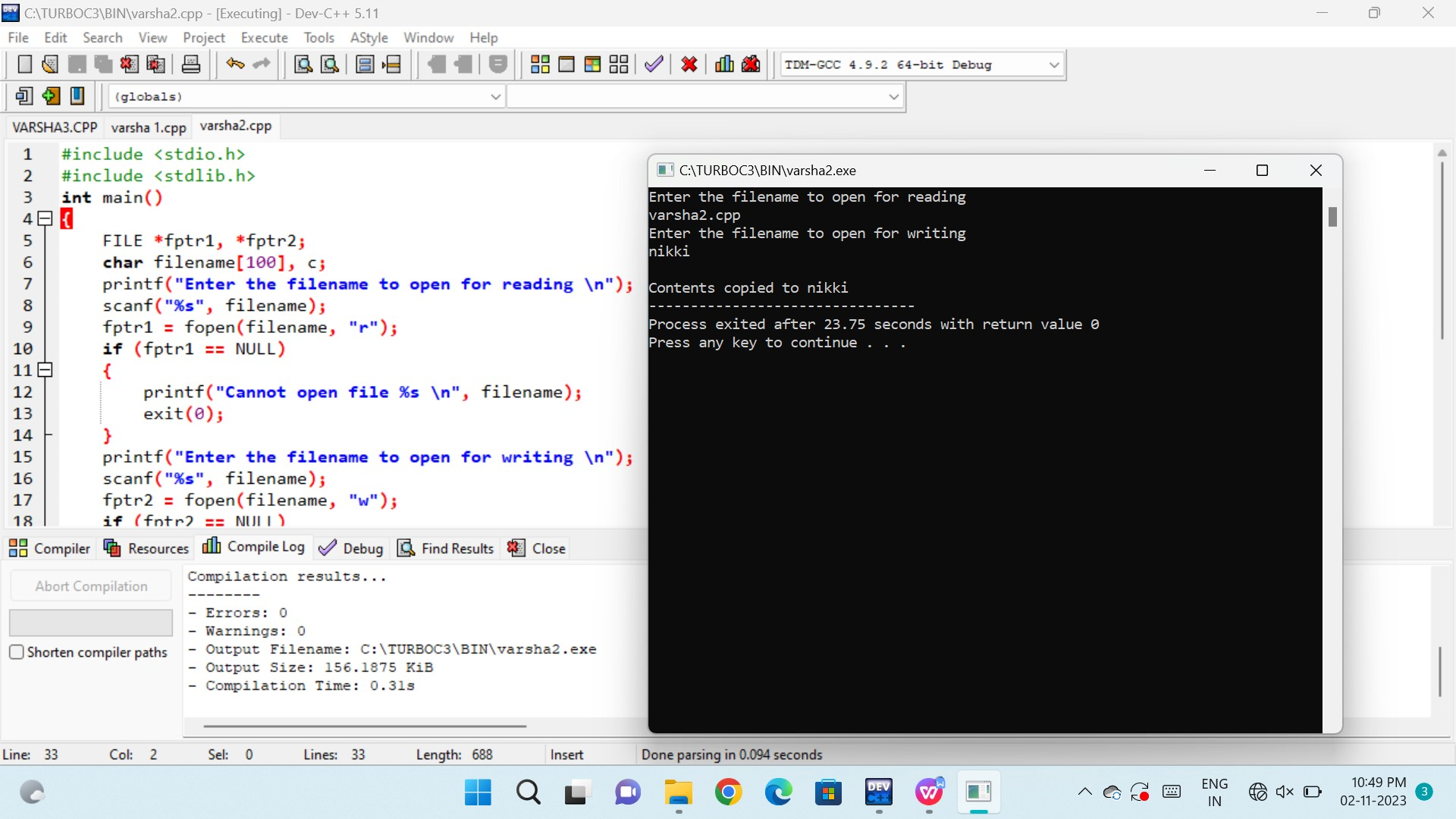
1.Create a new process by invoking the appropriate system call. Get the process identifier of the

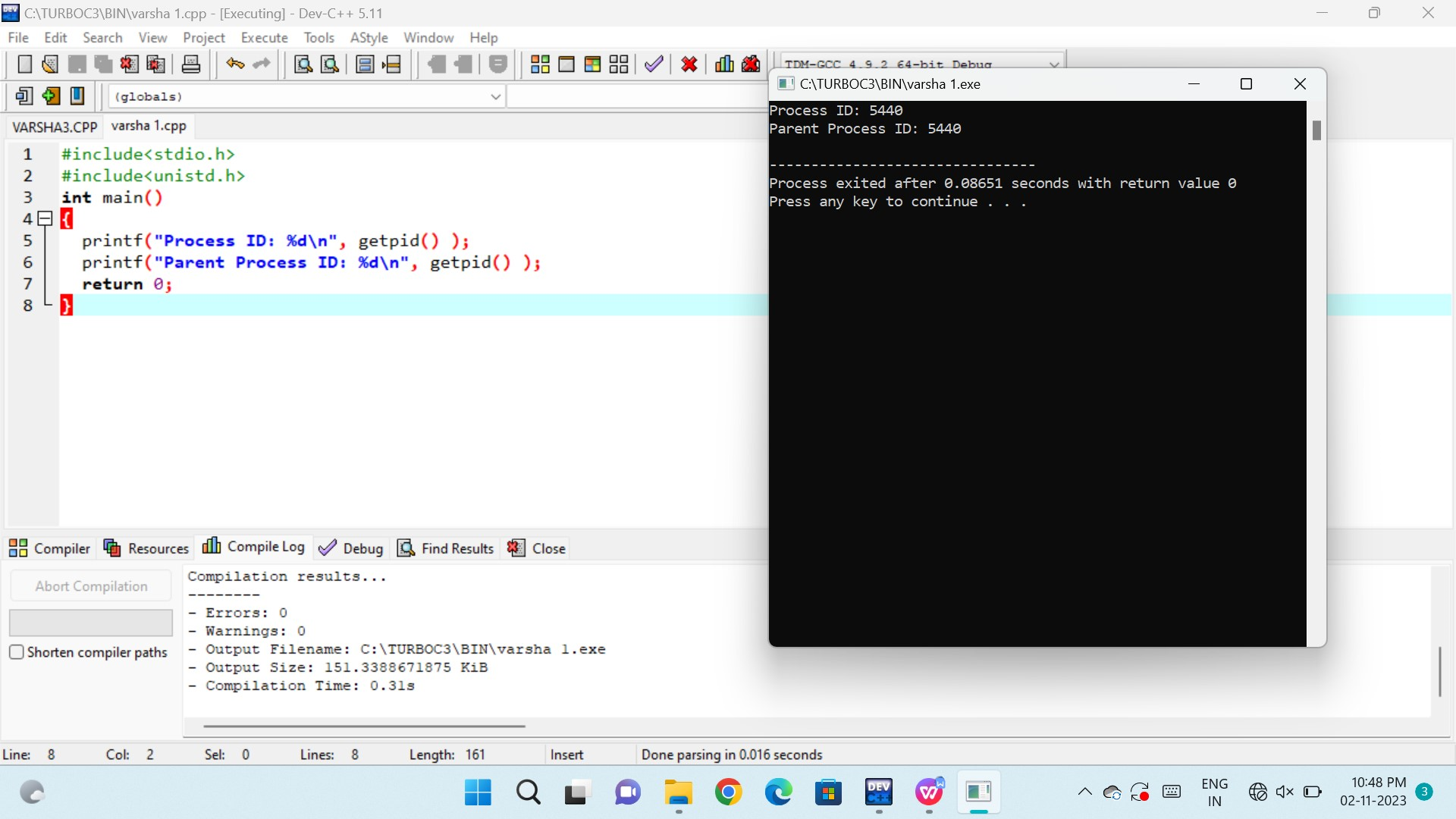
currently running process and its respective parent using system calls and display the same

using a C program.



2.Identify the system calls to copy the content of one file to another and illustrate the same using

a C program.

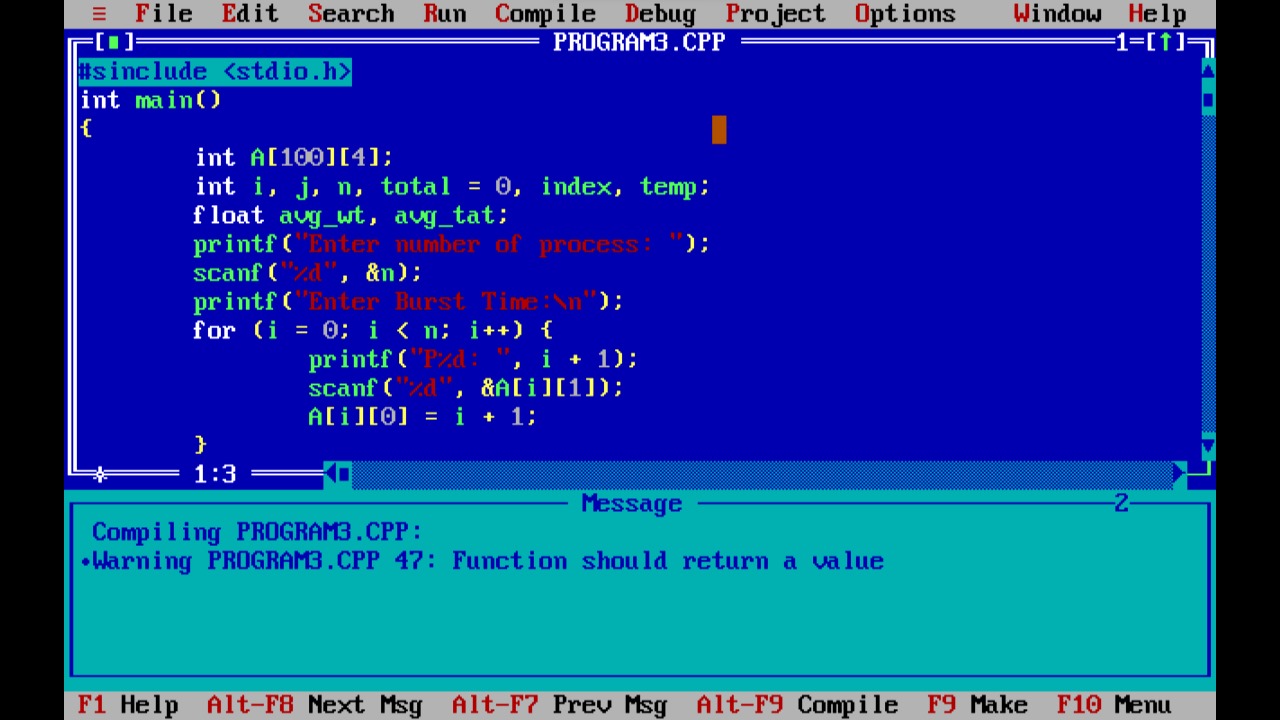


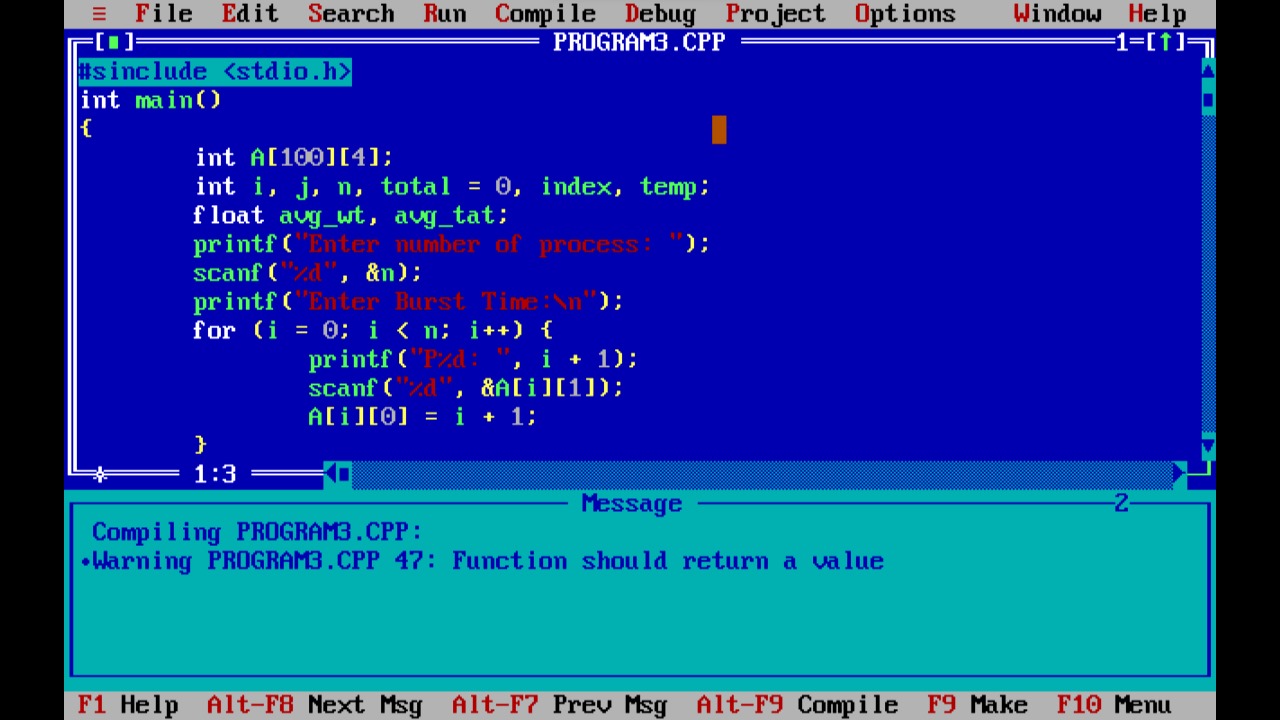
3. Design a CPU scheduling program with C using First Come First Served technique with the

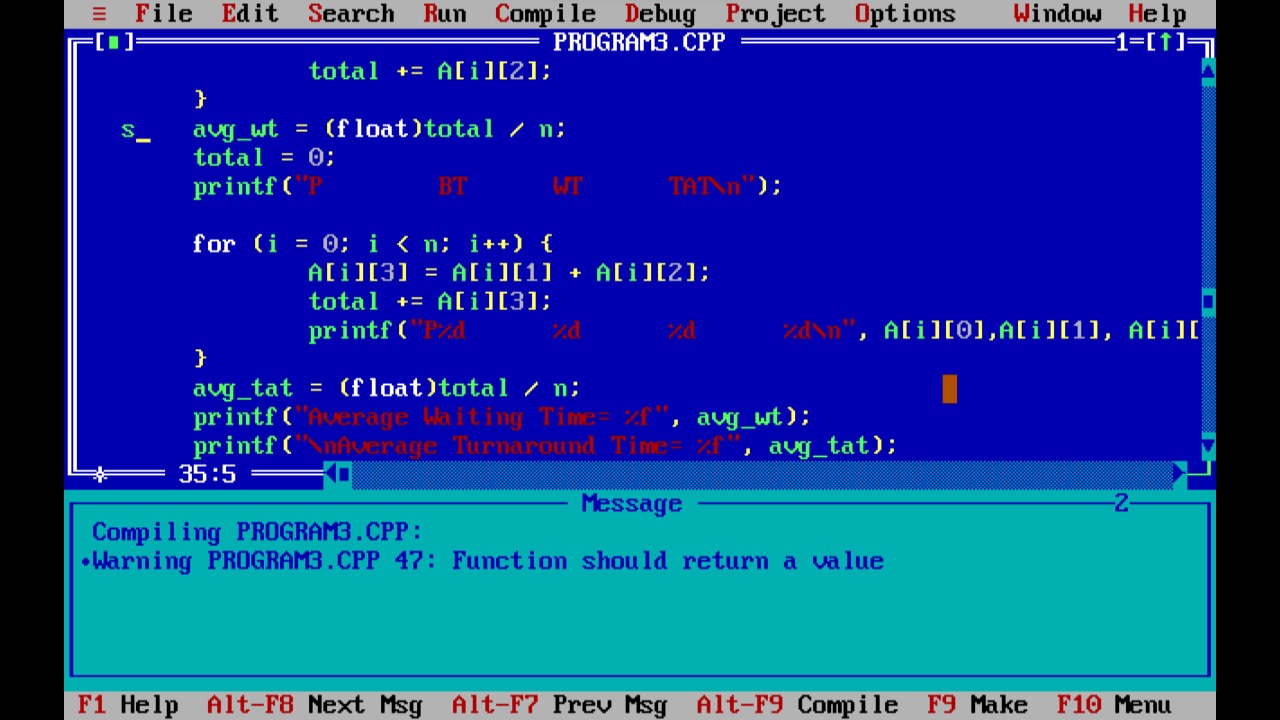
following considerations.

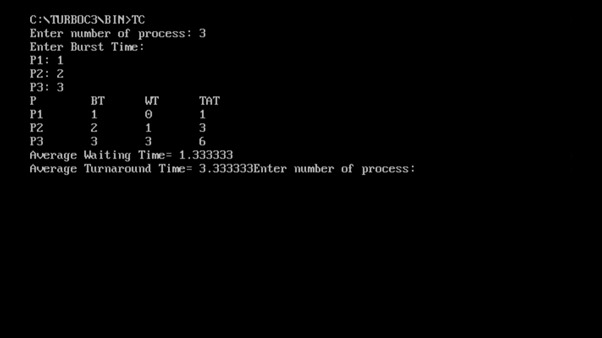
a. All processes are activated at time 0.

b. Assume that no process waits on I/O devices







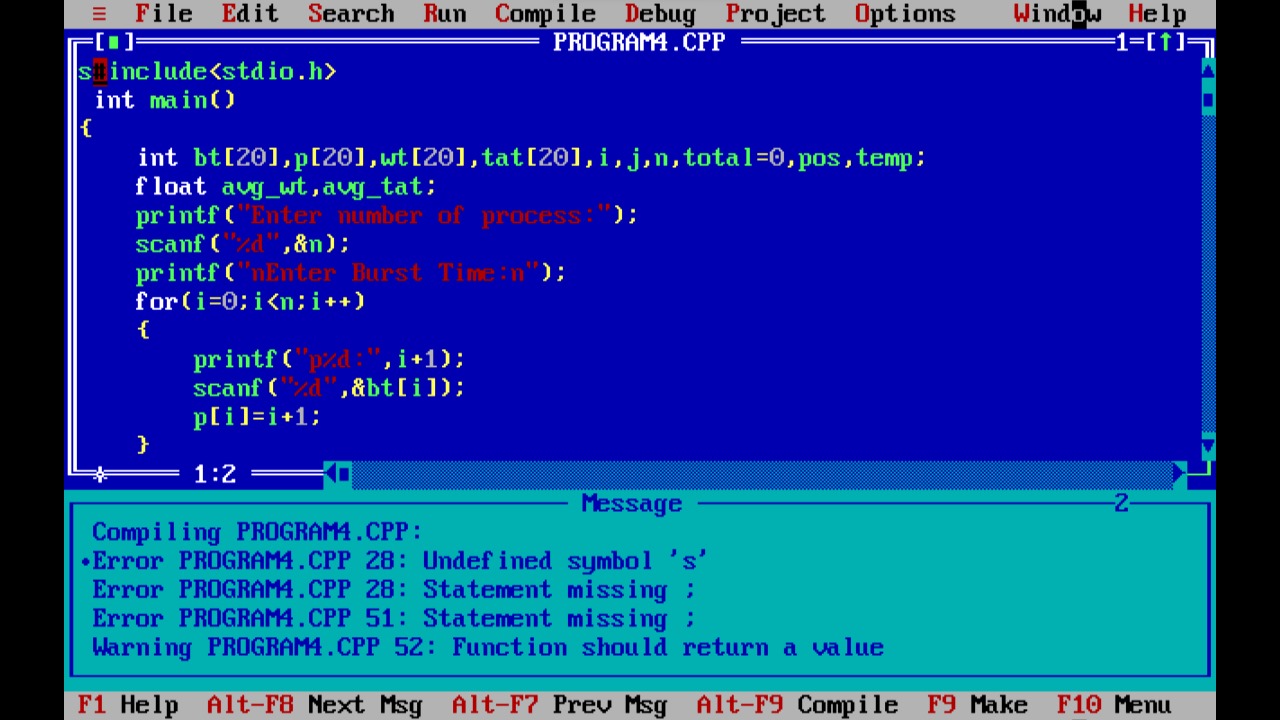


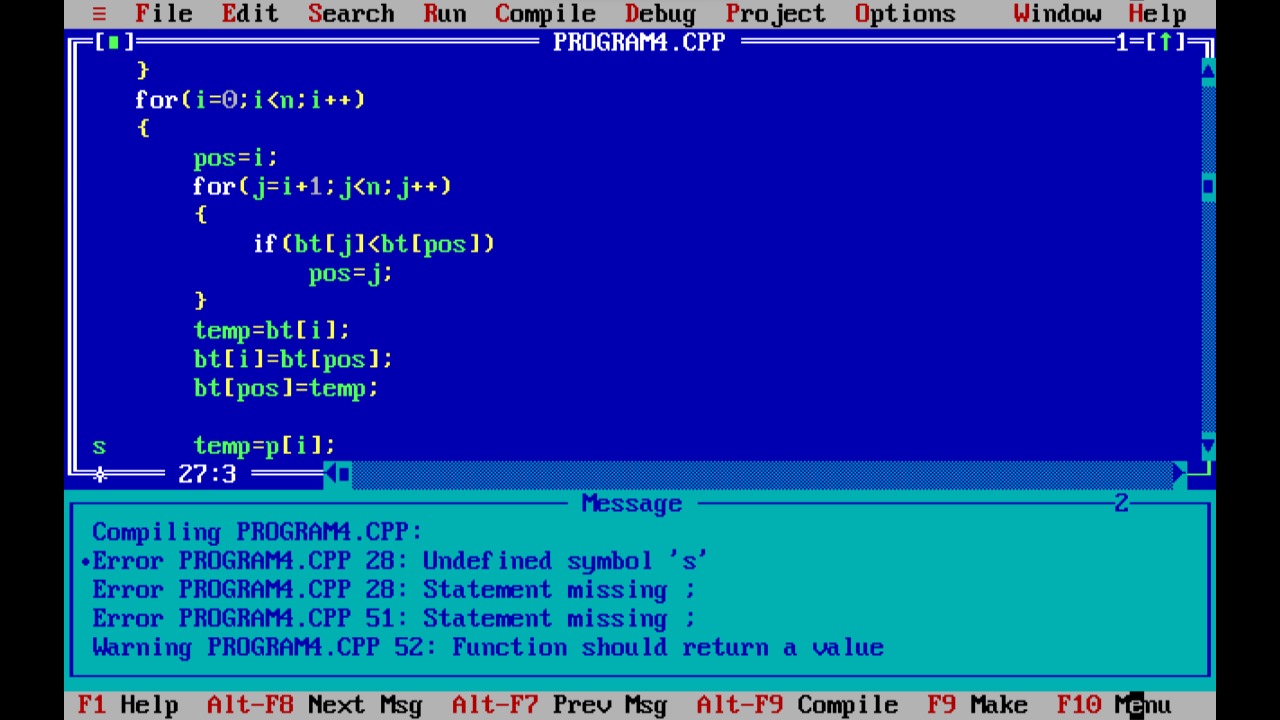
4. Design a CPU scheduling program with C using First Come First Served technique with the

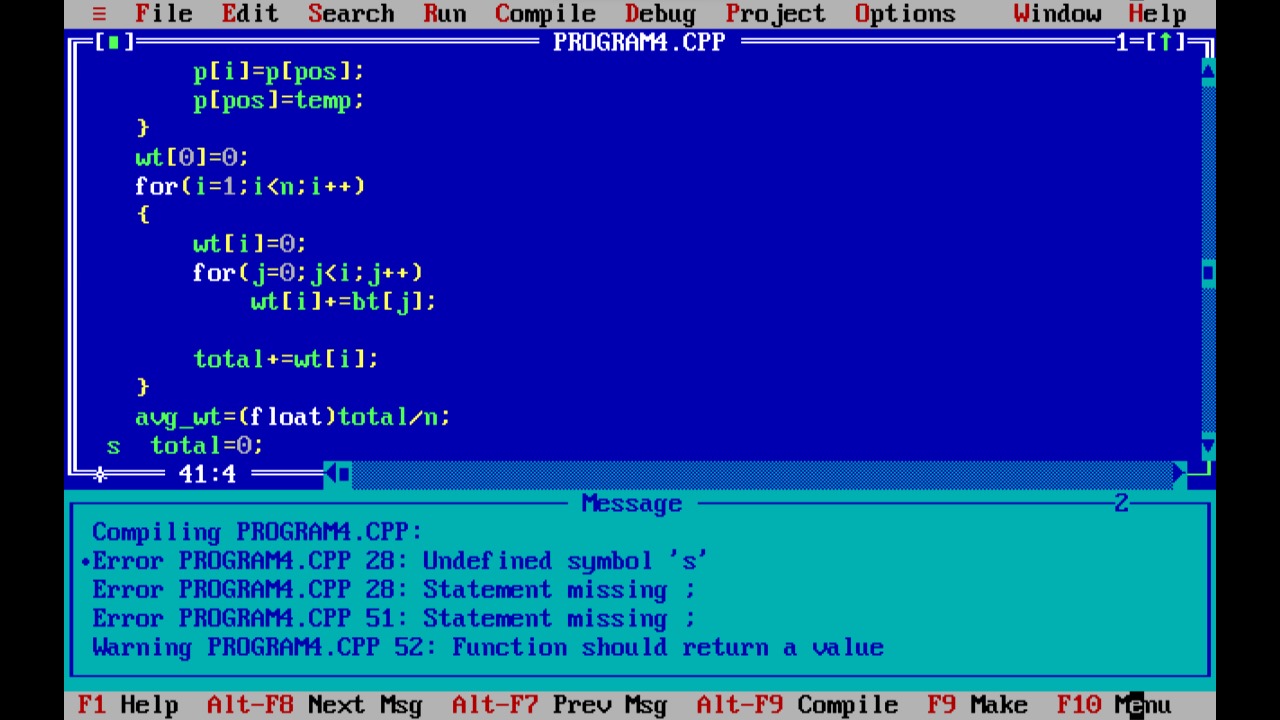
following considerations.

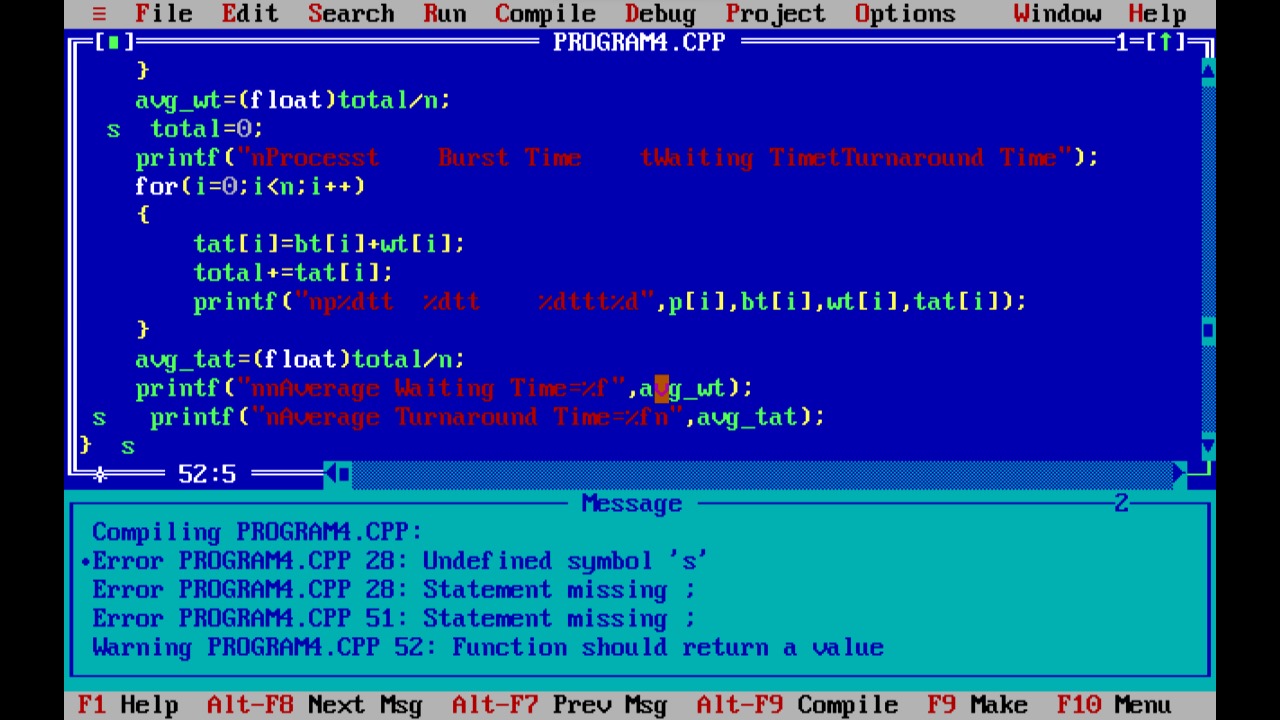
a. All processes are activated at time 0.

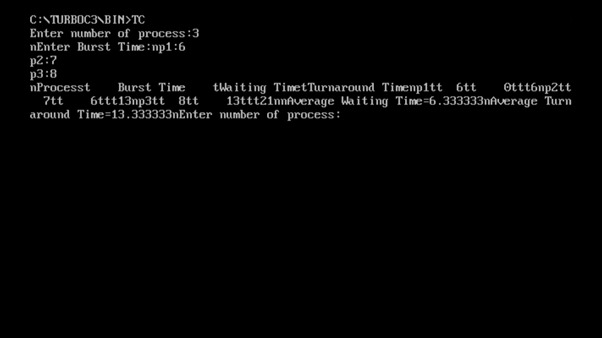
b. Assume that no process waits on I/O devices





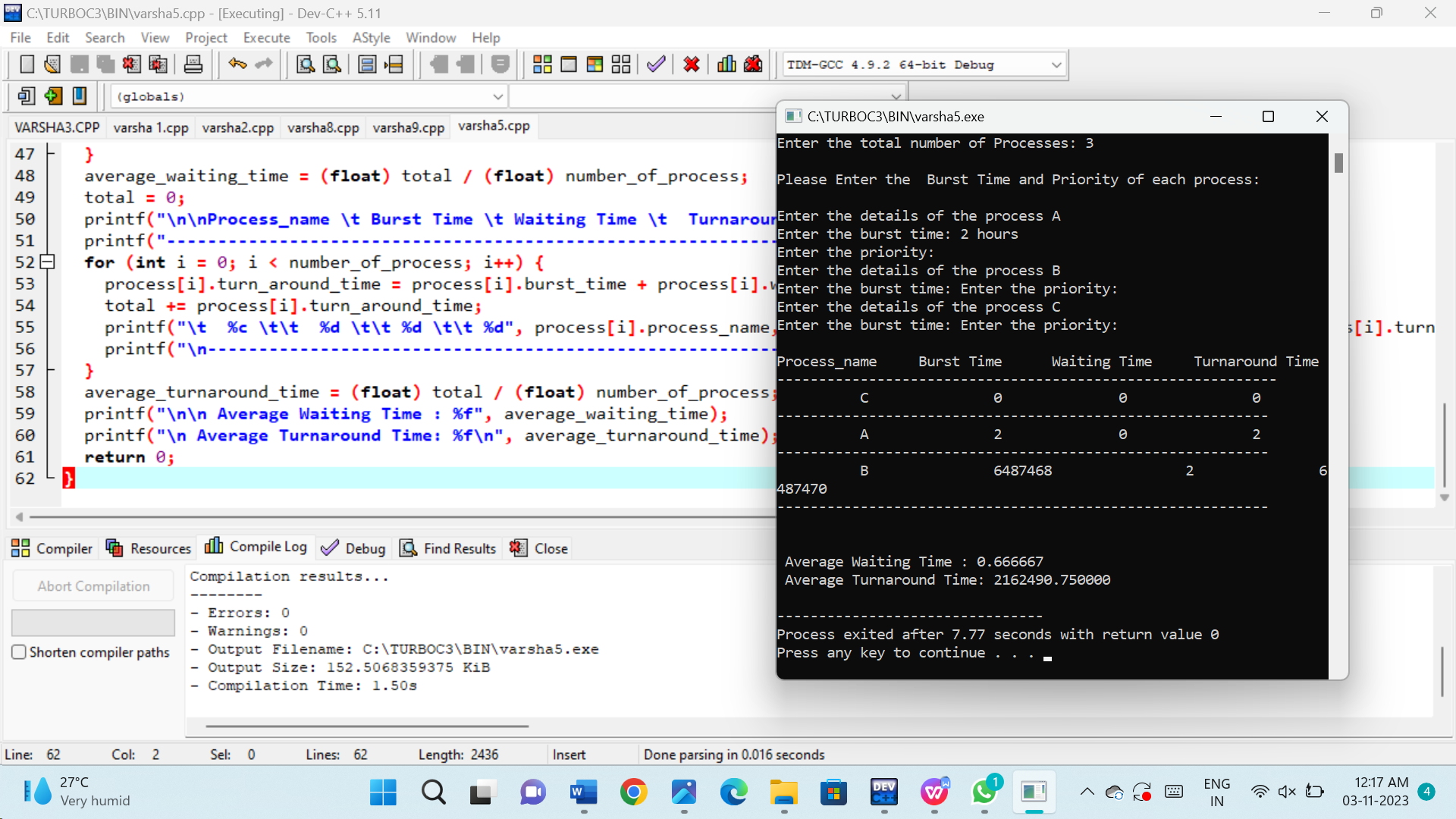




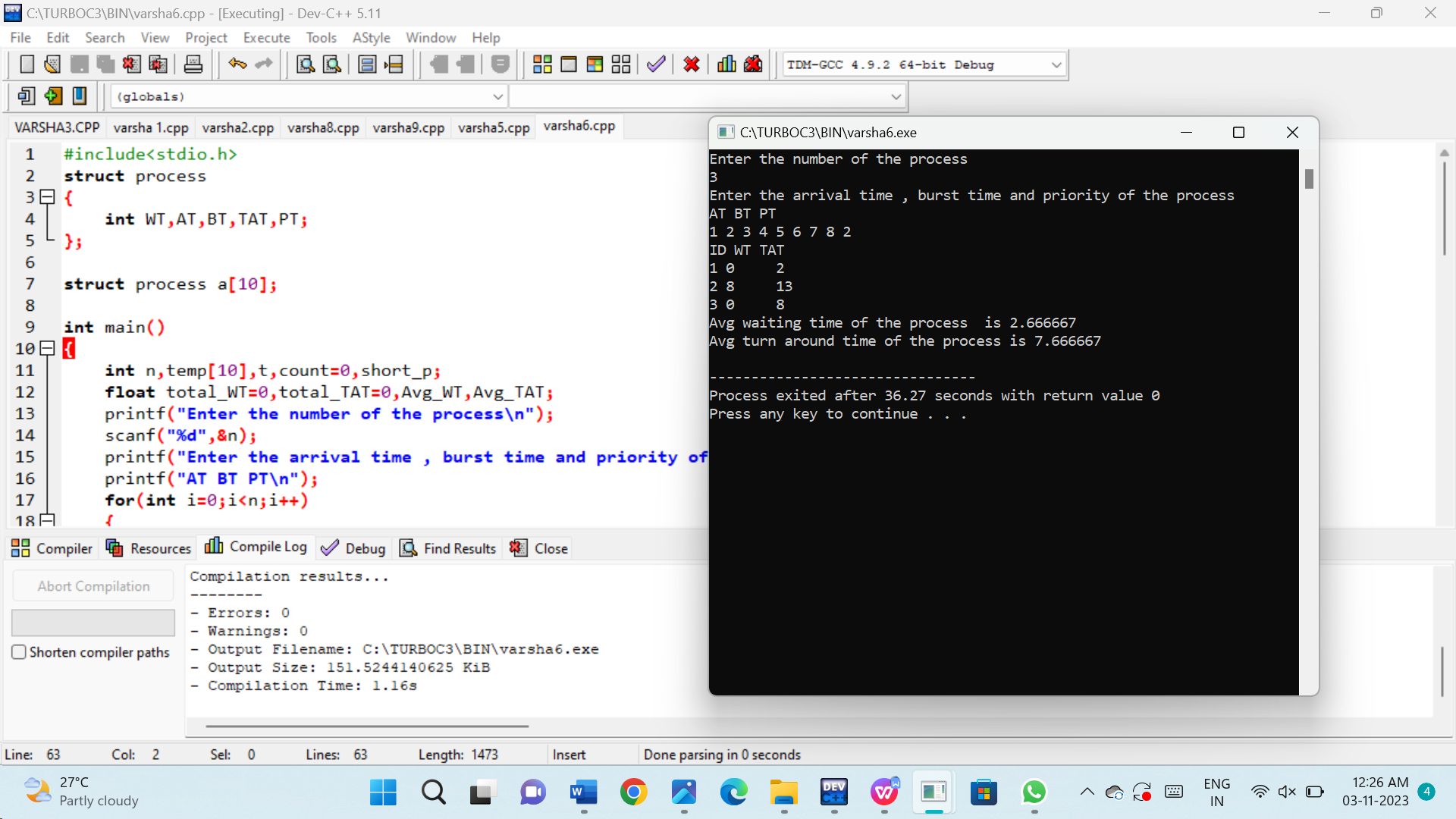


5. Construct a scheduling program with C that selects the waiting process with the highest

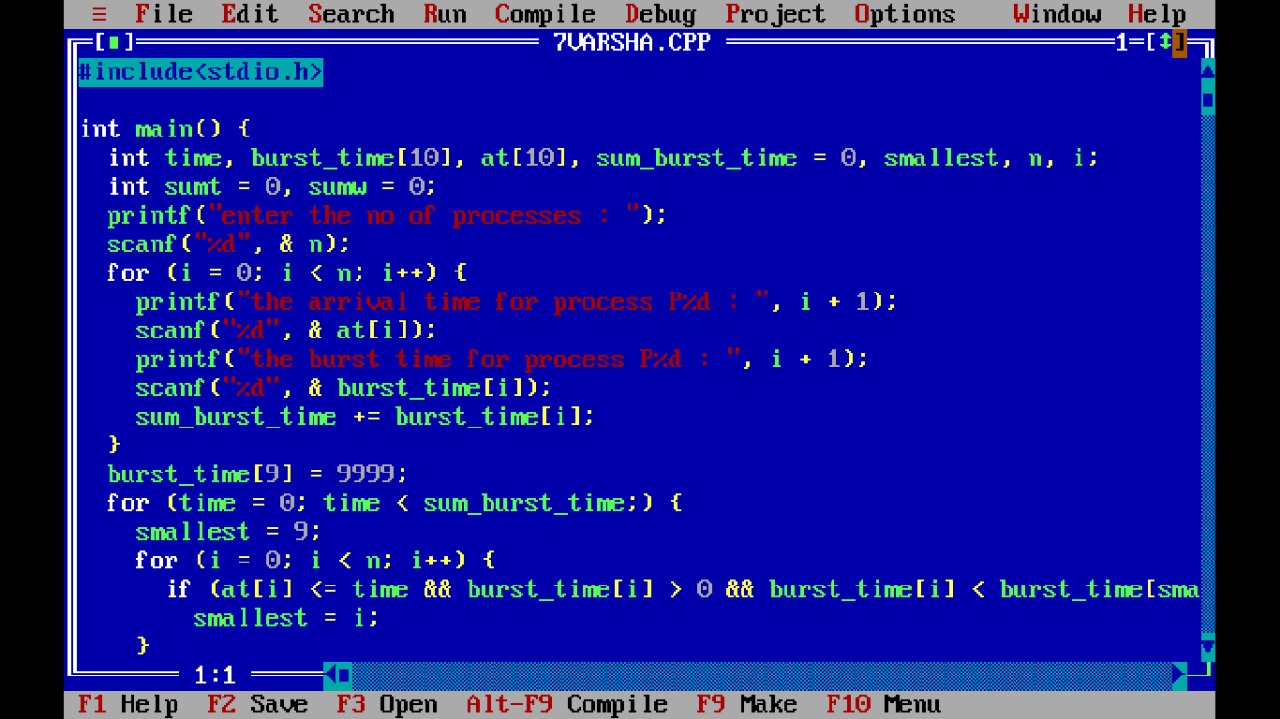
priority to execute next.

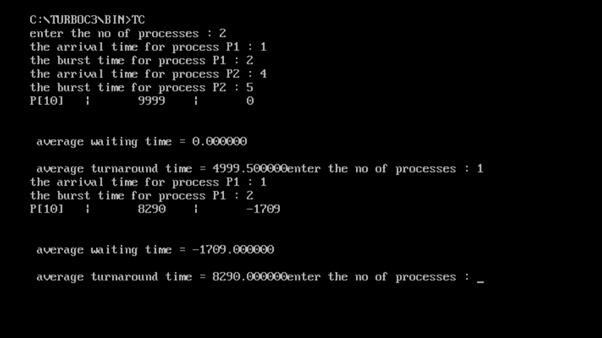


6.Construct a C program to implement pre-emptive priority scheduling algorithm.

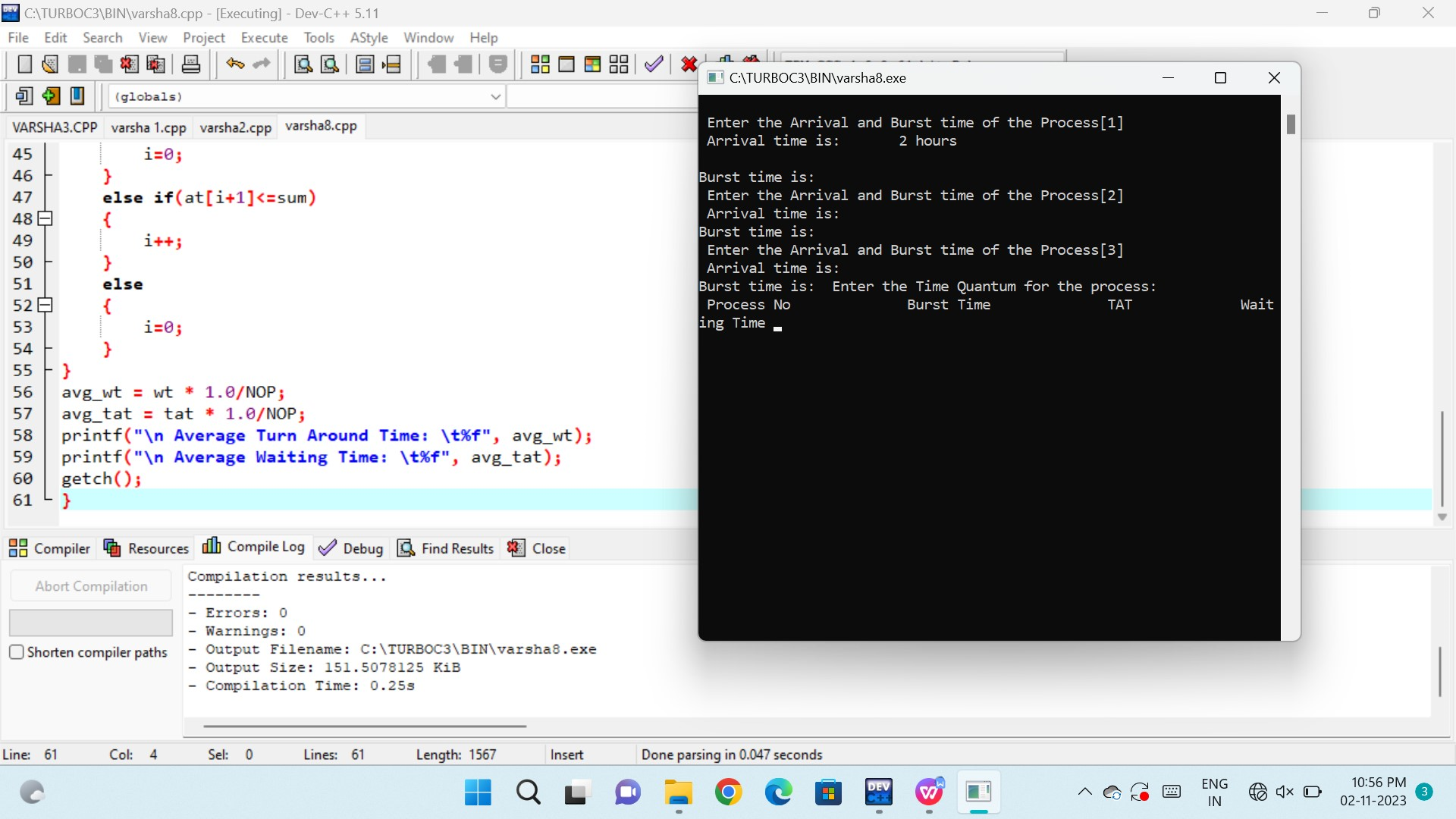


7. Construct a C program to implement non-preemptive SJF algorithm.

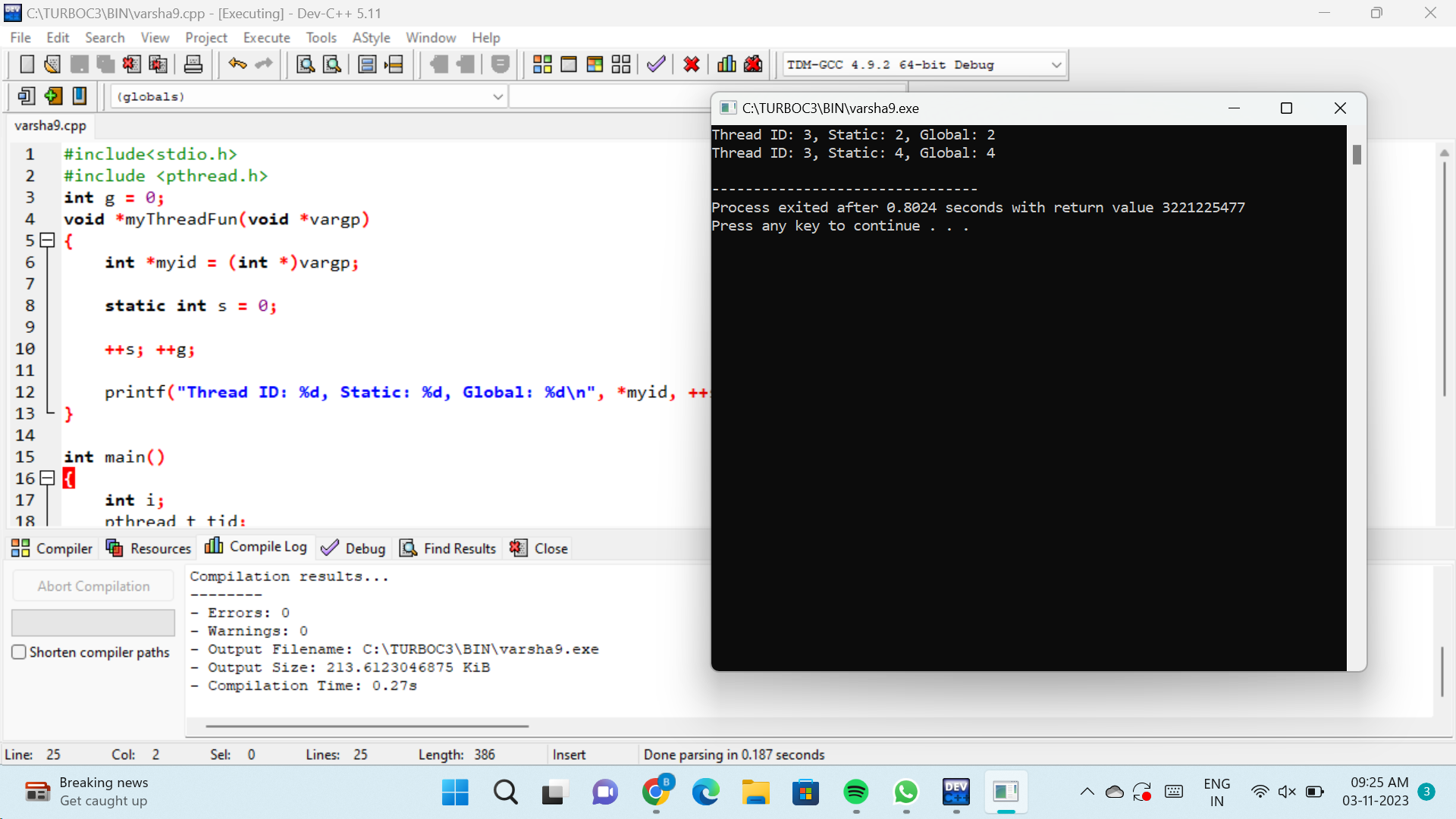




8. Construct a C program to simulate Round Robin scheduling algorithm with C.



9.Illustrate the concept of inter-process communication using shared memory with a C program.



10. Illustrate the concept of inter-process communication using message queue with a C

Program

