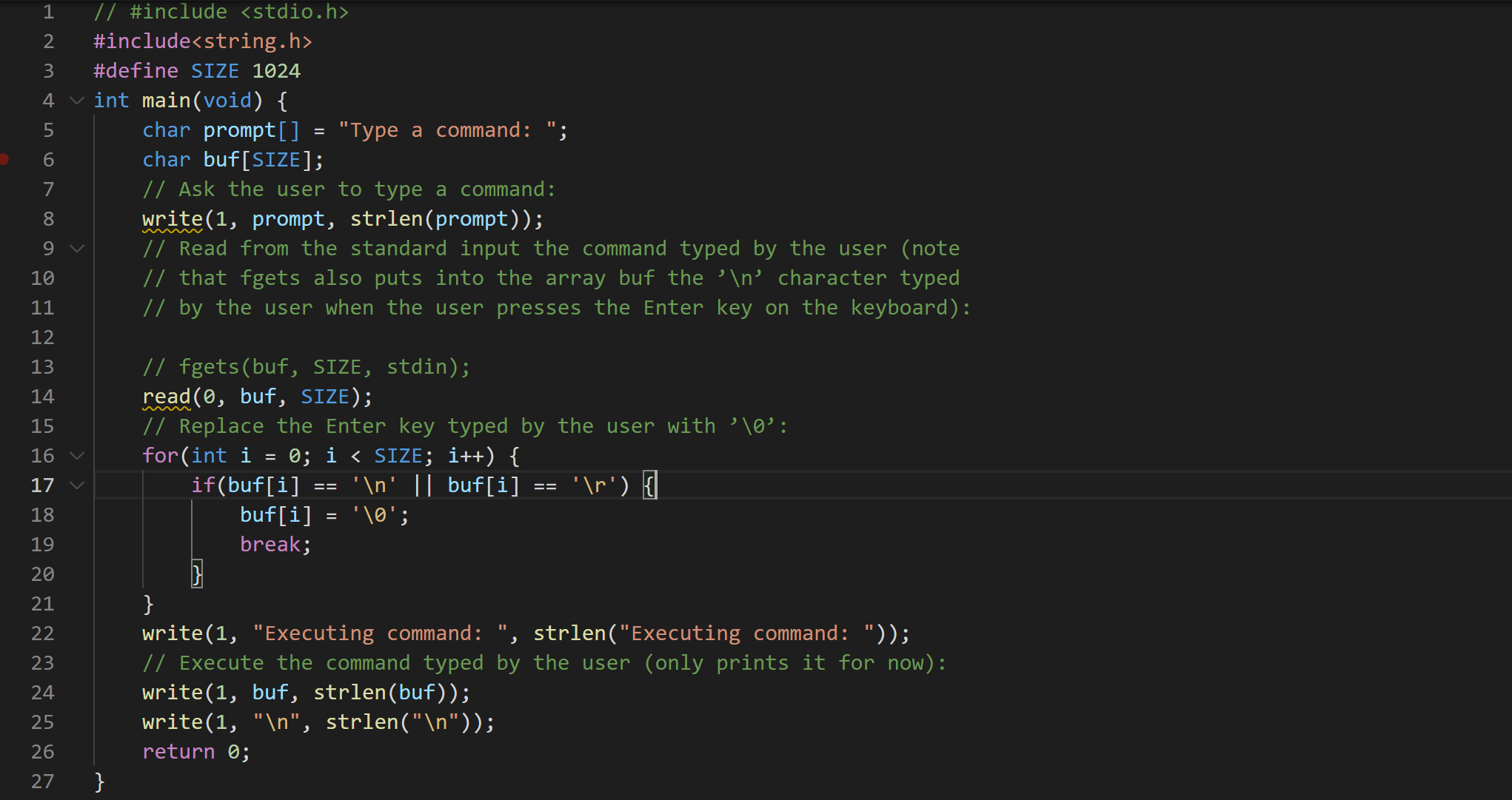
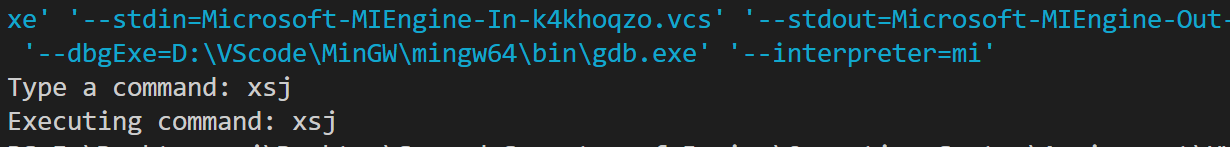
**Question 1**

**How to solve problem:**

1. Replace with "read" function and replace with function where and need to the library , thus we should remove the library line from the beginning of the program.
2. The function : . The is the file handle for which the file pointer is to be obtained and is 1. is the content to be written and the is the length of the file to be written; The function : . The is the file handle for which the file pointer is to be obtained and is 0. is the buffer to hold for what to read and the is the length of the file to be read.
3. All the should be replaced by including the "Type a command: " and "Executing command: " as reminder.
4. Remember to add the in the end of the write message. It also uses the function.

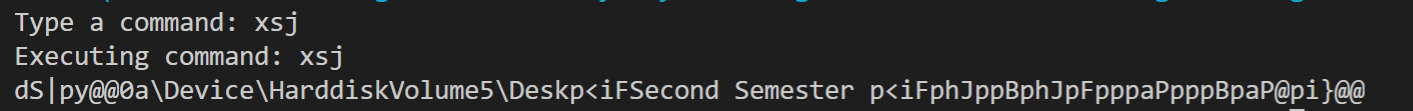


And the run results as follow:



**Problem encountered and solution:**

In the function read, the parameter can be which means up that to 1024 characters can be read. But in the function, the parameter cannot be . That’s since is the buffer address, it assigns the space to you as much as the length you write. So we should use the function to get the size of characters you write. Or you will get the other address information in the buffer, which are usually a lot of garbled code. Show as follow:



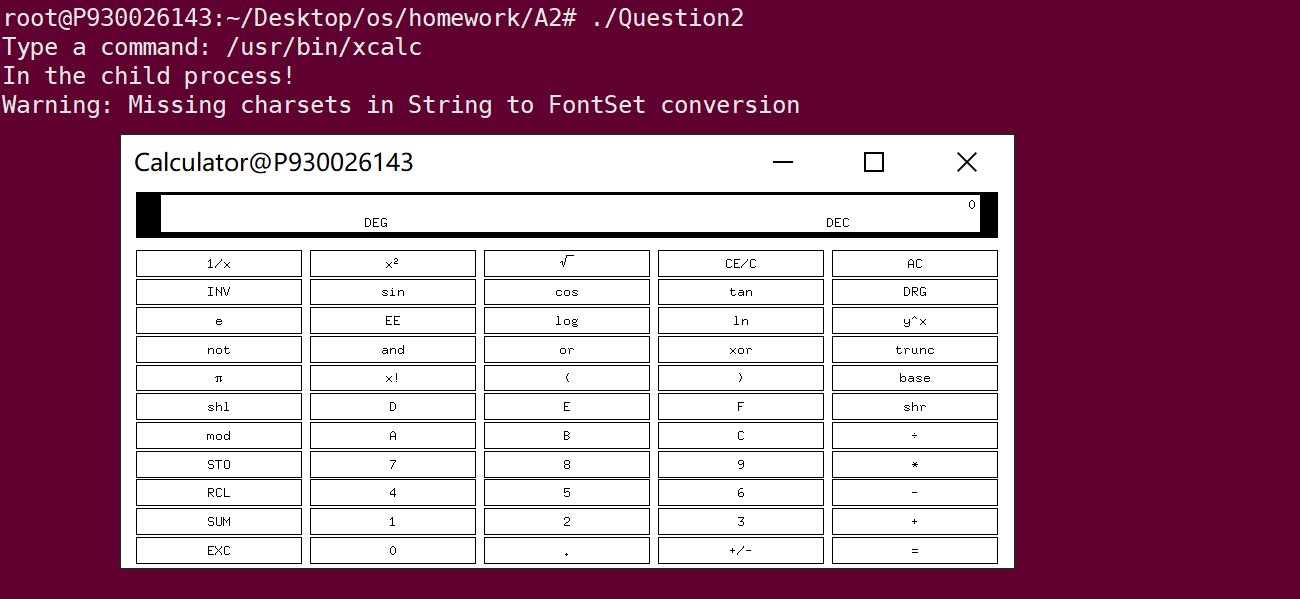
**Question 2**

**How to solve problem:**

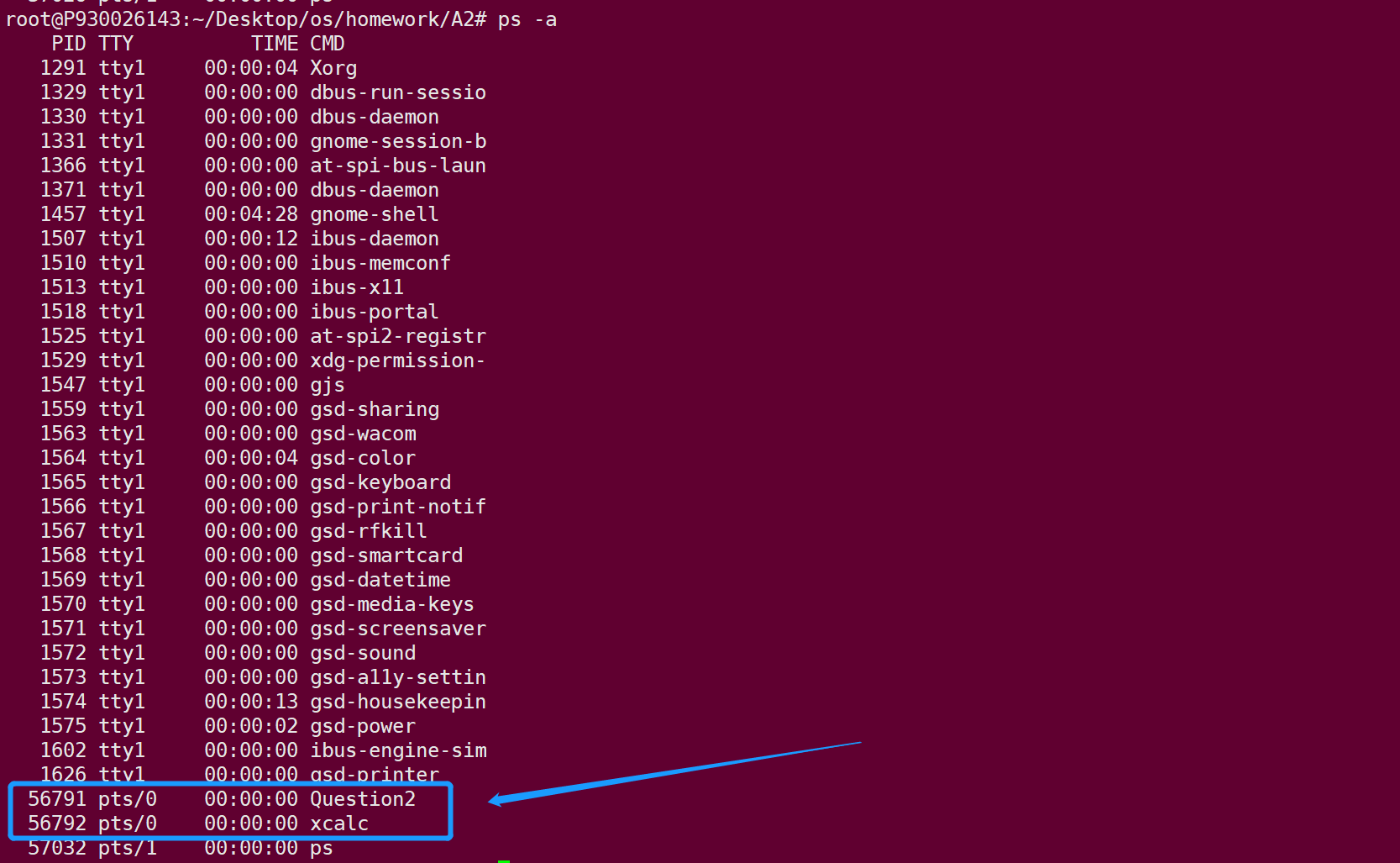
1. Create the child process by function below the read the buffer and Replace the Enter key typed by the user with . And we should add the judgment statement to check whether the child process is created successfully as well as distinguish the child process and the parent process. If , it means that it fails to create the child process. If , it means that it is the child process which the code in this part cannot be execute by the parent process. If , it means that it is the parent process which the code in this part cannot be execute by the child process.
2. In the case, we should be going to use what the user typed as the name of the program to execute in the new process, we can use the function to do that. In the case, we should make the parent process wait for the child process finish, we can use the function to do that. Definition:
3. function is used to temporarily stop the execution of the process until a signal arrives or the child process ends. The end status value of the child process is returned by the status parameter. Thus we should set the variable to represent status.

It should add the header file .

1. function is used to find the file from the PATH environment variable and execute. Definition: The last argument must be terminated by a NULL pointer. The function does not return on success, or -1 on failure. In other words, if the input file does not exist, it will become nothing to execute and nothing to happen. It should add the header file .
2. Then we use to compile the file and open by the , just like follow:



We can find that there is a calculator comes to the screen and we can check the pts by the command , task on the child process can be seen.



The is as follows:

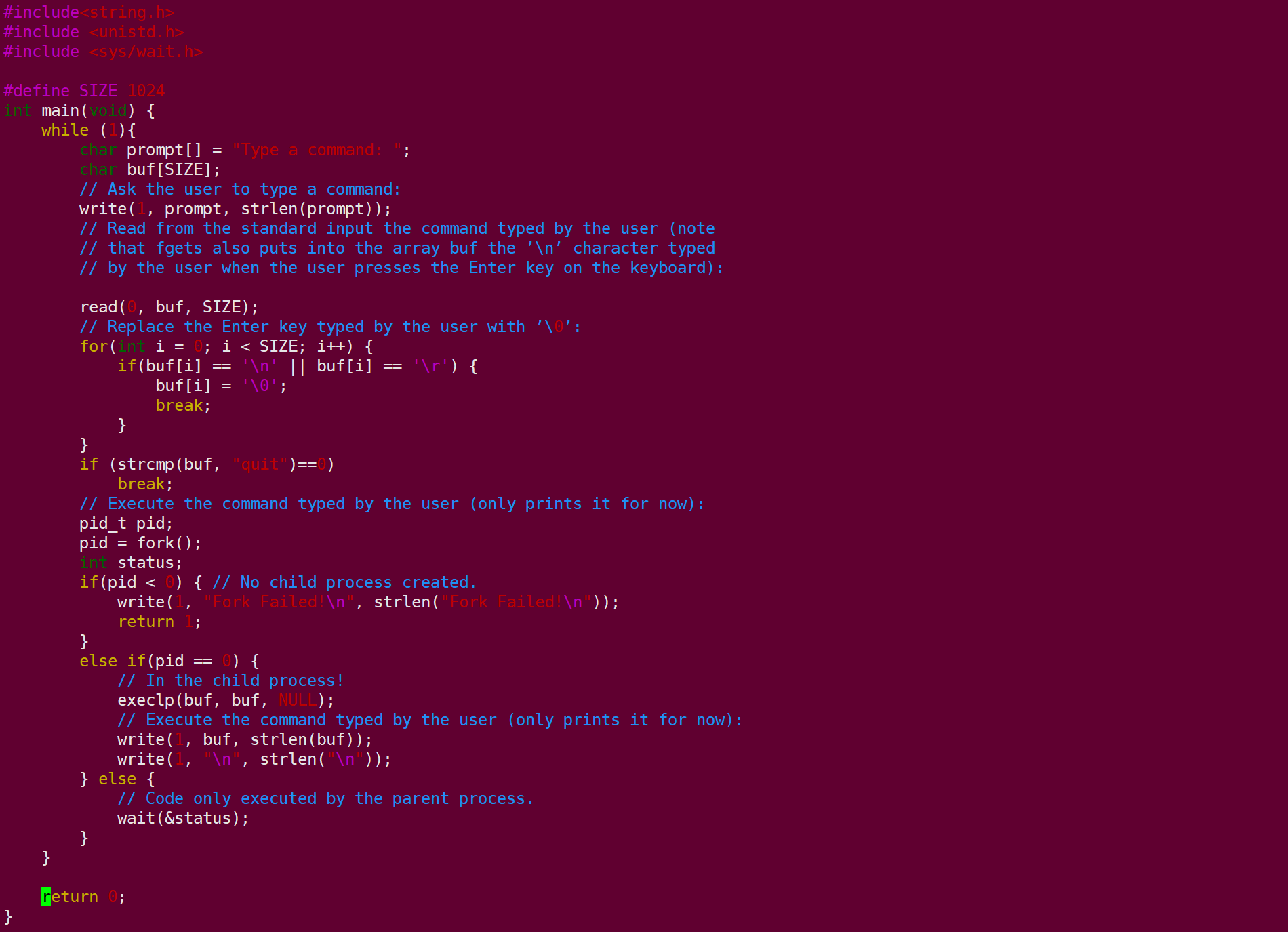


**Question 3**

**How to solve problem:**

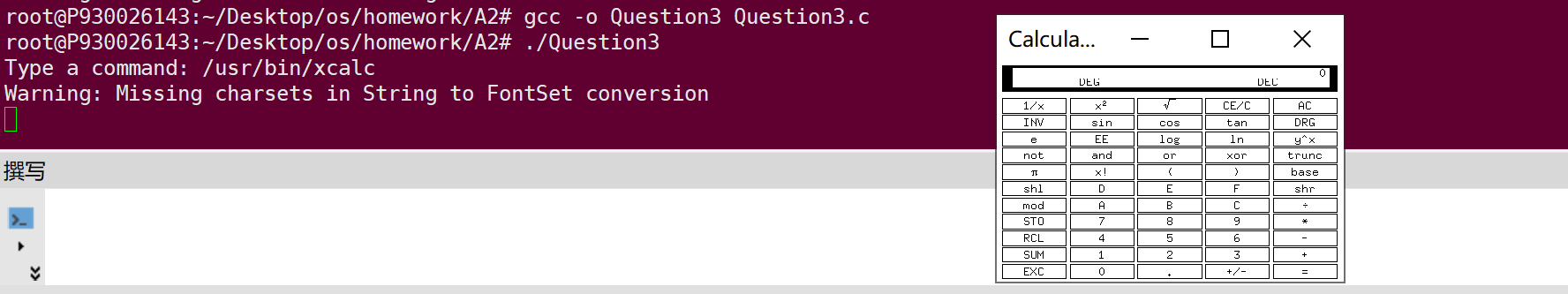
1. Based on the question 2, we should add a loop statement so that so that your program asks the user for input, creates a new child process, waits for the child process to end, then asks the user for input again and again.
2. We can use a while statement but not for statement because there is not a stated time of loop. And then we should add an end condition, when the buffer we read is equal to , it should escape out of the loop before create the child process and finish the parent process.

The file code is as follow:

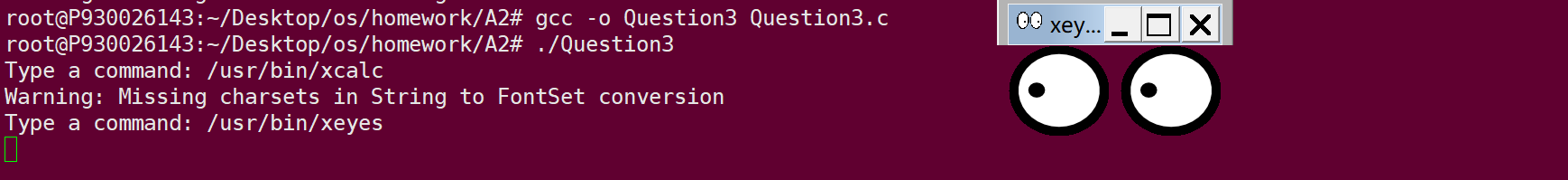


Then we use to compile the file and open by the , just like follow:

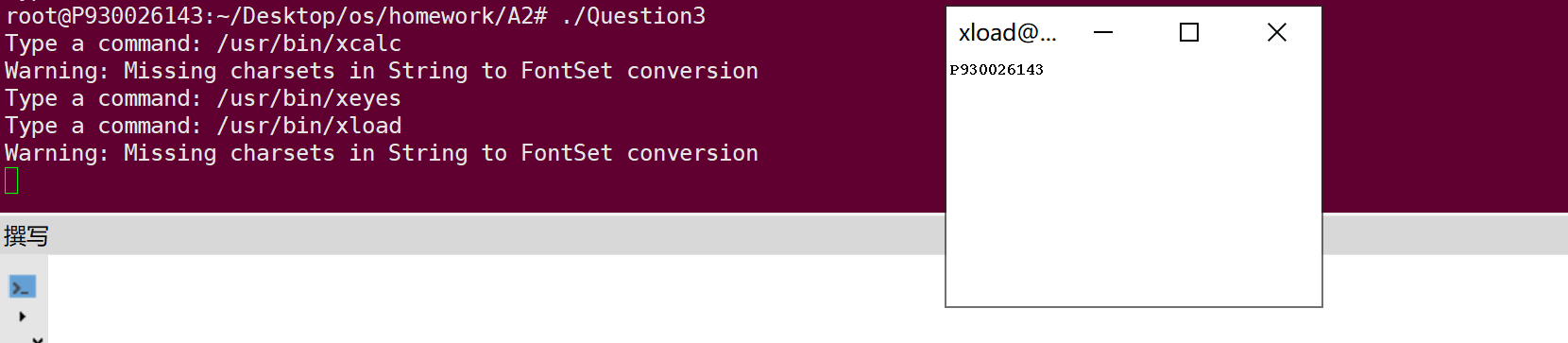
First, typed :



Second, close the calculator and continue to type , we will find a pair of eyes looking at the mouse arrow:



Third, close the eyes and continue to type , we will see a textbox which contain the hostname pop out to the screen:



In the end, close the textbox and continue to type , we will find that the program exits successfully:

