***Observation***

Most Common Error I faced when I was writing the Code was:

1. Syntax Errors: Forgetting semicolons at the end of statements, mismatched parentheses, missing braces, etc.(This was the most Common error I faced as while writing code I usually Forget to add ; and }).
2. Logical Errors: When the program compiles and runs but does not produce the expected output due to errors in the logic of the code. (I faced this error When I was making the Trying to Deque the element from queue as it was not performing as expected. I also got this error while doing infix to postfix and prefix as when I completed the programthe output would not be the desired one (It was producing only Letter not Symbols.)
3. Memory Leaks: Forgetting to deallocate memory allocated dynamically using malloc, calloc, or realloc, leading to memory leaks. (This mainly occurred when I was implementing linked list using stack and queue.)
4. Buffer Overflows: Writing data beyond the bounds of an array or buffer, potentially overwriting important data or causing segmentation faults. ( This mainly occurred when I was the size of stack was Static like I already defined the size but I gave the input more than the size.)
5. Uninitialized Variables: Using variables without initializing them, which can lead to unpredictable behavior. (I feel like this Error is faced by every programmer as We mainly don’t focus on writing the main logic and there are certain parts where we need new variable but forget to initialize it. This happed a lot to me while I was writing the logic as sometimes, I require a new variable and I forget to initialize it.)
6. Type Mismatch Errors: Passing arguments of the wrong type to functions, or assigning values of incompatible types to variables. ( I faced this error When Int was the datatype but when I was calling them I was Calling char.
7. Infinite Loops: Writing loops that never terminate due to incorrect loop conditions or missing loop control statements.
8. Off-by-One Errors: Incorrectly accessing array elements or iterating through loops with incorrect start or end conditions, leading to unexpected behavior.
9. Ignoring Return Values: Neglecting to check return values of functions like malloc, scanf, or fopen, which may lead to undetected errors.
10. Failure to Include Header Files: Forgetting to include necessary header files, resulting in compilation errors due to missing function prototypes or type definitions.(This could be one of the most stupid error I have done. See the thing is in some Text editor Even if you start from int main the program will run and when I used to write code in copy I didn’t include header file which led me to forget writing the header file although while making all the program I forgot to Write #include<stdlib.h> which is necessary for Dynamic memory allocation)
11. Ignoring Warnings: Ignoring compiler warnings, which may indicate potential issues in the code.(I got int to Pointer warning which also comes int his category).

***Some Major Things That I learned for this Lab file:***

* 1. **Memory Efficiency**: Pointers enable efficient memory usage by allowing data structures to allocate memory dynamically only when needed, rather than pre-allocating fixed-size memory blocks. This results in better memory utilization, especially for large or variable-sized data sets.
  2. **Reusability**: Functions allow code to be reused across different parts of the program. For example, the same stack implementation can be used in multiple places without rewriting the code. This promotes code efficiency and reduces redundancy.

One more thing:

I have Attached two Screenshot Which shows the compilation time While using function and by not using function. It basically Concludes that Compilation time Using the function gets reduced.

