Task 1J: Using #ifdef Directive to Prevent Multiple Inclusion of Header Files

Objective:
To understand and demonstrate how the #ifdef preprocessor directive can be used in a header file to ensure that it is included only once in a C program. This technique is known as an include guard and prevents multiple definition errors during compilation.

1. Concept Overview

When a header file is included multiple times, it can lead to redefinition errors. To avoid this, include guards are used with #ifdef, #define, and #endif directives. They ensure that the contents of a header file are processed only once by the compiler.

2. Header File (myheader.h)

```
#ifndef MYHEADER_H
 #define MYHEADER_H

 void displayMessage();

 #endif
```

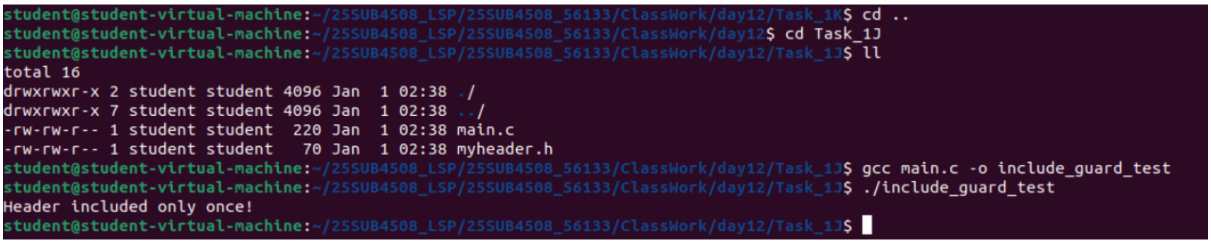
3. Source File (main.c)

```
#include <stdio.h>
 #include "myheader.h"
 #include "myheader.h"   // Included twice intentionally

 void displayMessage() {
         printf("Header included only once!\n");
 }

 int main() {
         displayMessage();
         return 0;
 }
```

4. Compilation Instructions

Compile the program using GCC:

```
gcc main.c -o include_guard_test
```



```
student@student-virtual-machine:~/25SUB4508_LSP/25SUB4508_56133/ClassWork/day12/Task_1K$ cd ..
student@student-virtual-machine:~/25SUB4508_LSP/25SUB4508_56133/ClassWork/day12$ cd Task_1J
student@student-virtual-machine:~/25SUB4508_LSP/25SUB4508_56133/ClassWork/day12/Task_1J$ ll
total 16
drwxrwxr-x 2 student student 4096 Jan  1 02:38 ./
drwxrwxr-x 7 student student 4096 Jan  1 02:38 ../
-rw-rw-r-- 1 student student  220 Jan  1 02:38 main.c
-rw-rw-r-- 1 student student   70 Jan  1 02:38 myheader.h
student@student-virtual-machine:~/25SUB4508_LSP/25SUB4508_56133/ClassWork/day12/Task_1J$ gcc main.c -o include_guard_test
student@student-virtual-machine:~/25SUB4508_LSP/25SUB4508_56133/ClassWork/day12/Task_1J$ ./include_guard_test
Header included only once!
student@student-virtual-machine:~/25SUB4508_LSP/25SUB4508_56133/ClassWork/day12/Task_1J$ 
```

5. Program Output

```
Header included only once!
```

6. Observations & Explanation

1. The header file is included twice in the source file.
2. The #ifndef directive checks if the macro MYHEADER_H is not defined.
3. On first inclusion, MYHEADER_H is not defined, so the header content is processed.
4. MYHEADER_H is then defined using #define.
5. On second inclusion, MYHEADER_H is already defined, so the header content is skipped.
6. This prevents multiple declarations and compilation errors.

7. Conclusion

This task demonstrates the importance of include guards in C programming. Using #ifdef (or #ifndef) ensures that header files are included only once, making the code safer, modular, and free from redefinition issues.