NAME : KSHITIJ GUPTA
Enrolment Number : 21162101007
Sub: CD

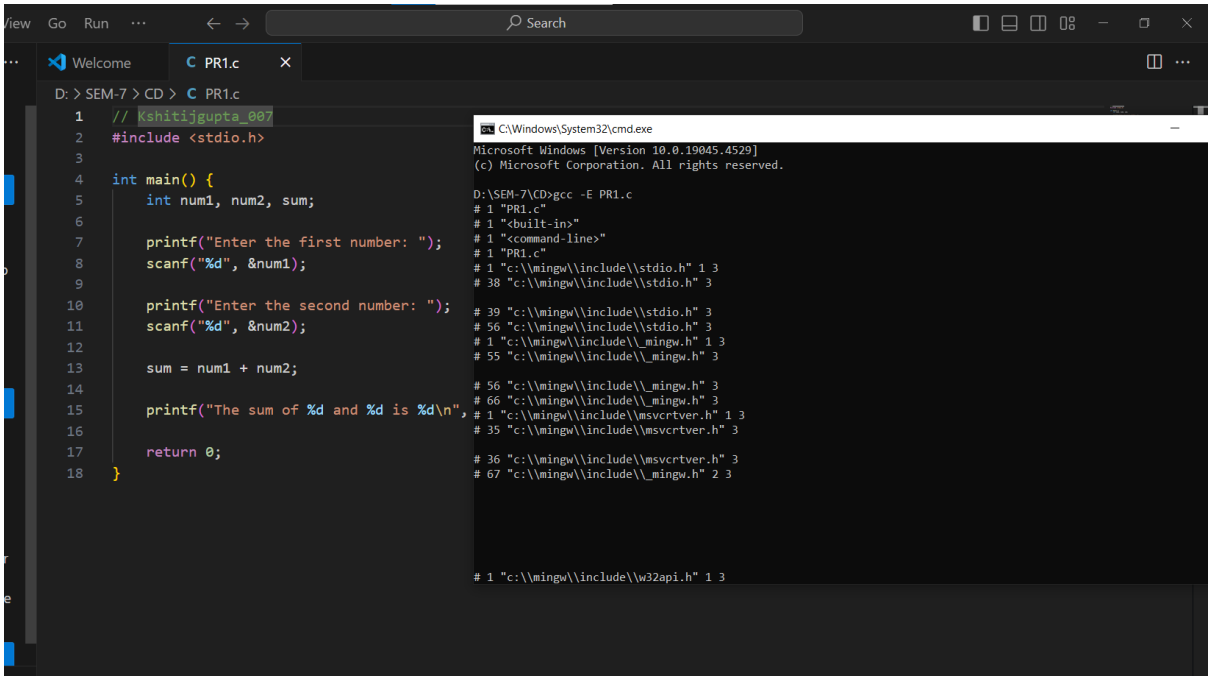
Practical – 1[Batch-71]

Tasks:

1) Understand modules of the compilation process with the help of a program. (Pre-processor, Compiler, Assembler, Linker/Loader)

Considering the following c program of addition of two numbers

```c
#include <stdio.h>

int main() {
    int num1, num2, sum;

    printf("Enter the first number: ");
    scanf("%d", &num1);

    printf("Enter the second number: ");
    scanf("%d", &num2);

    sum = num1 + num2;

    printf("The sum of %d and %d is %d\n", num1, num2, sum);

    return 0;
}
```



Compilation Processes

1. **Preprocessor:**
   It processes the directives (starts with the '#' symbol) and produces the intermediate code.

Here, **gcc -E prac1_1.c** it processes the file and produces the pre-processed output

-E : option to perform the pre-processing stage of compilation

It expands macros, includes the content of header files, and handles conditional compilation, generating an intermediate pre-processed code as output.

```
D:\SEM-7\CD>gcc -E PR1.c
# 1 "PR1.c"
# 1 "<built-in>"
# 1 "<command-line>"
# 1 "PR1.c"
# 1 "c:\\mingw\\include\\stdio.h" 1 3
# 38 "c:\\mingw\\include\\stdio.h" 3

# 39 "c:\\mingw\\include\\stdio.h" 3
# 56 "c:\\mingw\\include\\stdio.h" 3
# 1 "c:\\mingw\\include\\_mingw.h" 1 3
# 55 "c:\\mingw\\include\\_mingw.h" 3

# 56 "c:\\mingw\\include\\_mingw.h" 3
# 66 "c:\\mingw\\include\\_mingw.h" 3
# 1 "c:\\mingw\\include\\msvcrtver.h" 1 3
# 35 "c:\\mingw\\include\\msvcrtver.h" 3

# 36 "c:\\mingw\\include\\msvcrtver.h" 3
# 67 "c:\\mingw\\include\\_mingw.h" 2 3
```

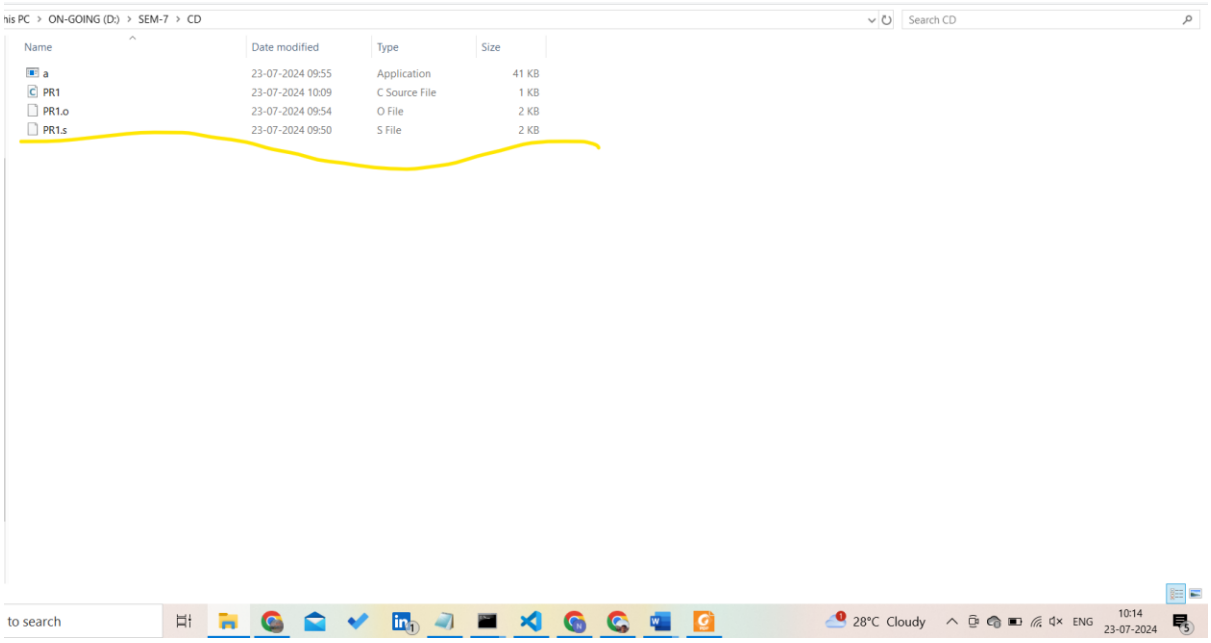
```
# 2 "PR1.c" 2


# 3 "PR1.c"
int main() {
    int num1, num2, sum;

    printf("Enter the first number: ");
    scanf("%d", &num1);

    printf("Enter the second number: ");
    scanf("%d", &num2);

    sum = num1 + num2;

    printf("The sum of %d and %d is %d\n", num1, num2, sum);

    return 0;
}
```

2. **Compiler:**
   The pre-processed code is translated into assembly code specific to the target machine's architecture by the compiler

Here, **gcc -S main.c",** processes the file and generates the assembly code output.

-S: option to perform the compilation process until the assembly code generation stage

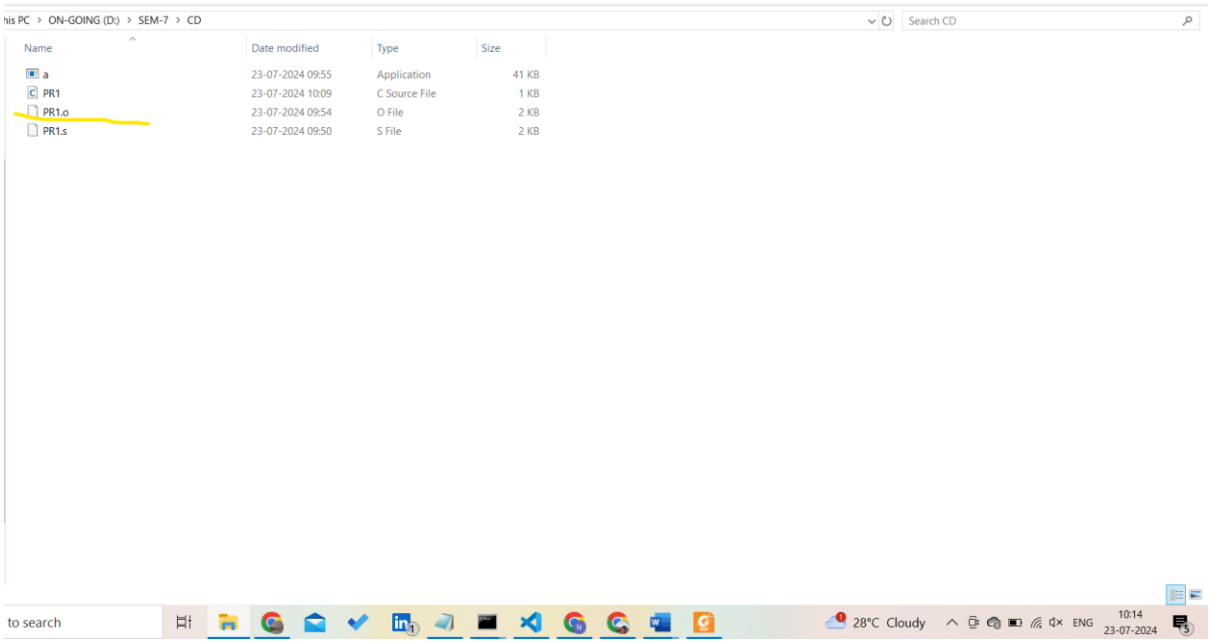
```
D:\SEM-7\CD>gcc -S PR1.c
```



3. **Assembler:**
   It takes the assembly code and translates it into machine code (binary code) that the computer's processor can understand and execute.

-c: option tells GCC to stop after the compilation step, so it will generate the object file but will not perform the final linking stage to create an executable.

```
D:\SEM-7\CD>gcc -c PR1.s
```



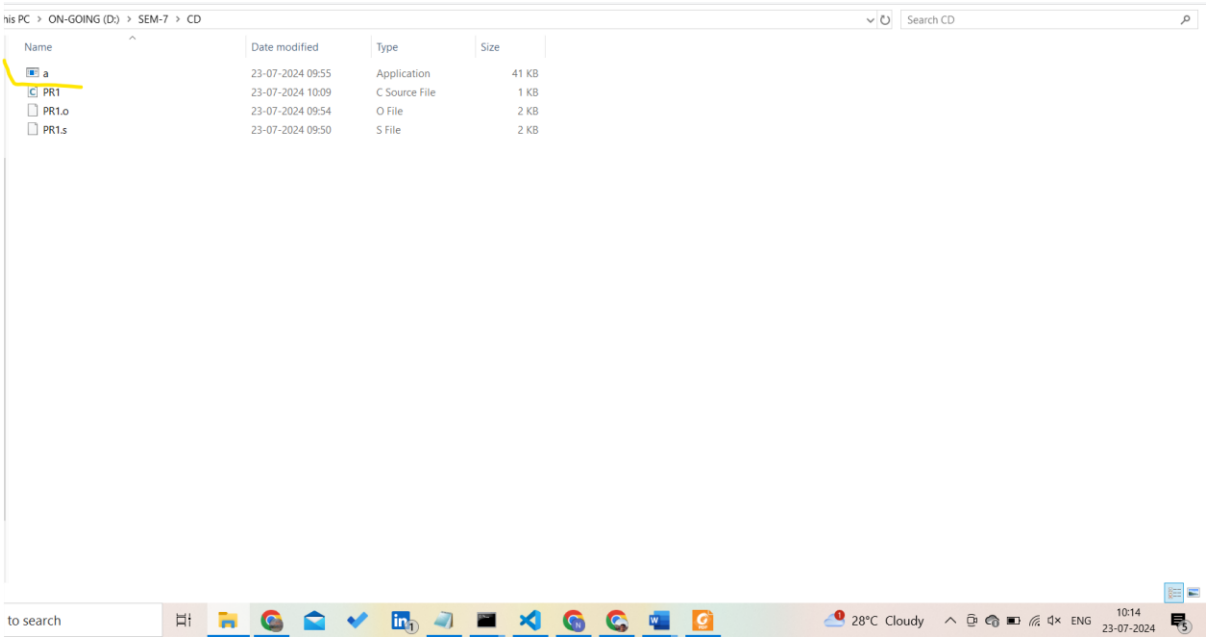
4. Linker/Loader:

Linker takes one or more object files and combines them into a single executable file.

After the executable file is created by the linker, it needs to be loaded into memory before the operating system can execute it which is done by the loader.

Here is the output of the code

```
D:\SEM-7\CD>gcc -L try PR1.o
```



Output result after running .exe file:

```
D:\SEM-7\CD>a
Enter the first number: 20
Enter the second number: 30
The sum of 20 and 30 is 50

D:\SEM-7\CD>
```

2) Write a C program to test whether a given identifier is valid or not.

Source Code:

```c
#include <stdio.h>
#include <string.h>
#include <ctype.h>

int main() {
    char keywords[10][10] = {"auto", "break", "case", "char", "const",
"continue", "default", "do","double", "else", "enum", "extern", "float",
"for", "goto", "if","int", "long", "register", "return", "short", "signed",
"sizeof", "static","struct", "switch", "typedef", "union","unsigned", "void",
"volatile", "while"};

    char identifier[10];
    printf("Enter Identifier: ");
    scanf("%s", identifier);

    // Check if the identifier is a keyword
    for (int i = 0; i < 10; i++) {
        if (strcmp(keywords[i], identifier) == 0) {
            printf("\n%s is a Keyword.\n", identifier);
            return 0;
        }
    }

    // Check if the identifier is a valid identifier
    if ((isalpha(identifier[0]) || identifier[0] == '_')) {
        for (int i = 1; identifier[i] != '\0'; i++) {
            if (!isalnum(identifier[i]) && identifier[i] != '_') {
                printf("\n%s is Not a valid Identifier.\n", identifier);
                return 0;
            }
        }
        printf("\n%s is a valid Identifier.\n", identifier);
    } else {
        printf("\n%s is Not a valid Identifier.\n", identifier);
    }

    return 0;
}
```

```
D:\SEM-7\CD>gcc -c PR1_2.s

D:\SEM-7\CD>gcc -L try PR1_2.o

D:\SEM-7\CD>a
Enter Identifier: kshitij

kshitij is a valid Identifier.

D:\SEM-7\CD>a
Enter Identifier: 07ks

07ks is Not a valid Identifier.

D:\SEM-7\CD>a
Enter Identifier: _kshitij

_kshitij is a valid Identifier.

D:\SEM-7\CD>a
Enter Identifier: kshitij123

kshitij123 is a valid Identifier.

D:\SEM-7\CD>
```