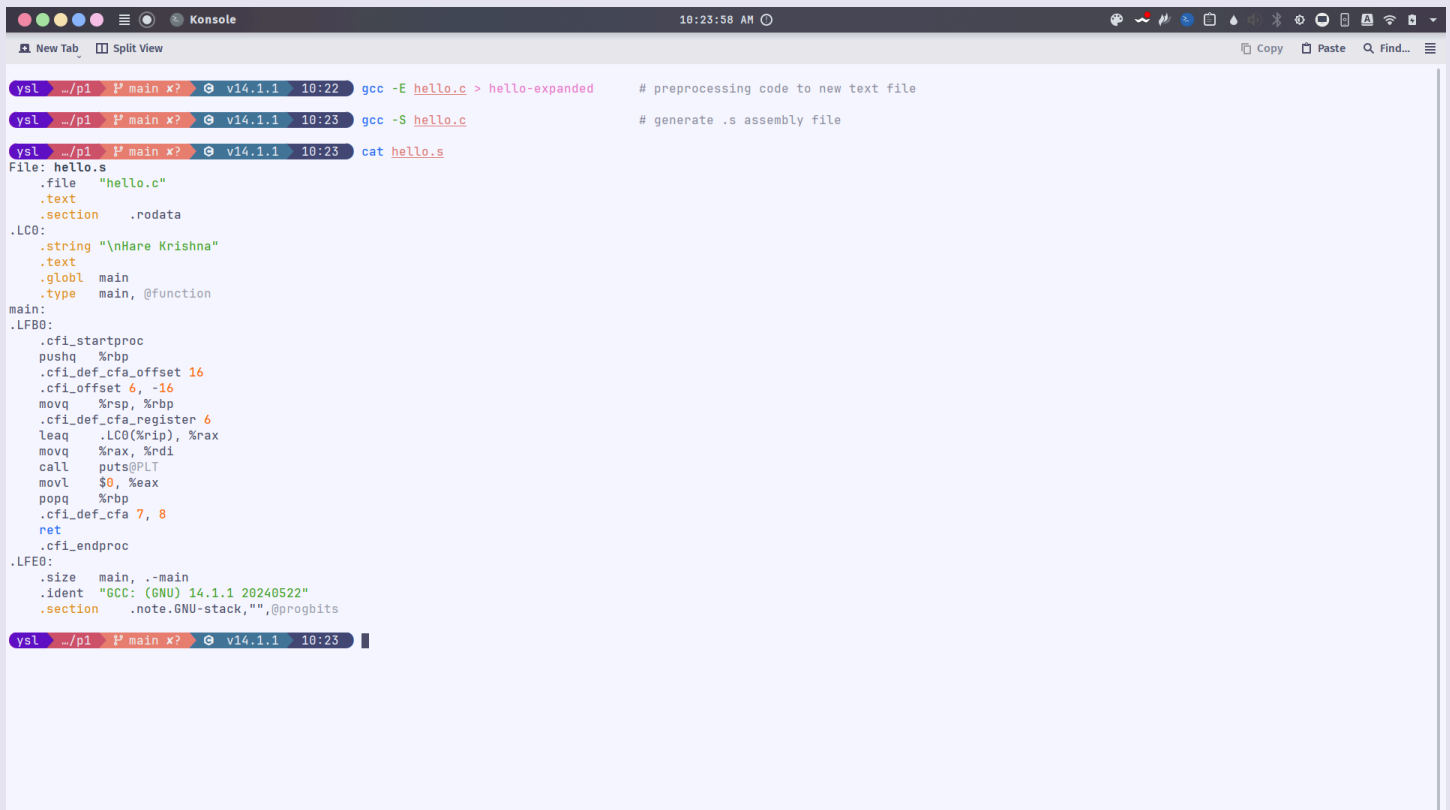


Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 71  
CD Practical 1

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

Stepwise screenshots with commands and comments :



```
ysl ~/p1 ? main x? v14.1.1 10:22 gcc -E hello.c > hello-expanded # preprocessing code to new text file
ysl ~/p1 ? main x? v14.1.1 10:23 gcc -S hello.c # generate .s assembly file
ysl ~/p1 ? main x? v14.1.1 10:23 cat hello.s
File: hello.s
.file "hello.c"
.text
.section .rodata
.LC0:
.string "\nHare Krishna"
.text
.globl main
.type main, @function
main:
.LFB0:
.cfi_startproc
pushq %rbp
.cfi_def_cfa_offset 16
.cfi_offset 4, -16
movq %rsp, %rbp
.cfi_def_cfa_register 6
leaq .LC0(%rip), %rax
movq %rax, %rdi
call puts@PLT
movl $0, %eax
popq %rbp
.cfi_def_cfa 7, 8
ret
.cfi_endproc
.LFE0:
.size main, .-main
.ident "GCC: (GNU) 14.1.1 20240522"
.section .note.GNU-stack,"",@progbits
ysl ~/p1 ? main x? v14.1.1 10:23
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 71  
CD Practical 1



```
ysl ~/p1 |? main x? @ v14.1.1 10:24 gcc -c hello.s           # generate object .o from .s assembly code
ysl ~/p1 |? main x? @ v14.1.1 10:25 ls | grep 'hello.o'
.rwxr--r-- 1.5k ysl 23 Jul 10:25 hello.o
ysl ~/p1 |? main x? @ v14.1.1 10:25
ysl ~/p1 |? main x? @ v14.1.1 10:25 gcc -L try hello.o        # generate final output file of code .out in unix based systems, .exe in Windows
ysl ~/p1 |? main x? @ v14.1.1 10:26 ls | grep '.out'
.rwxr-xr-x 15k ysl 23 Jul 10:26 a.out
ysl ~/p1 |? main x? @ v14.1.1 10:26 ./a.out                 # run final output file

Hare Krishna
ysl ~/p1 |? main x? @ v14.1.1 10:26
```

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

Code :

```
kywrds = {
    "auto", "break", "case", "char", "const", "continue", "default", "do",
    "double", "else",
    "enum", "extern", "float", "for", "goto", "if", "inline", "int",
    "long", "register",
    "restrict", "return", "short", "signed", "sizeof", "static", "struct",
    "switch", "typedef",
    "union", "unsigned", "void", "volatile", "while", "_Alignas",
    "_Alignof", "_Atomic",
    "_Bool", "_Complex", "_Decimal128", "_Decimal32", "_Decimal64",
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 71  
CD Practical 1

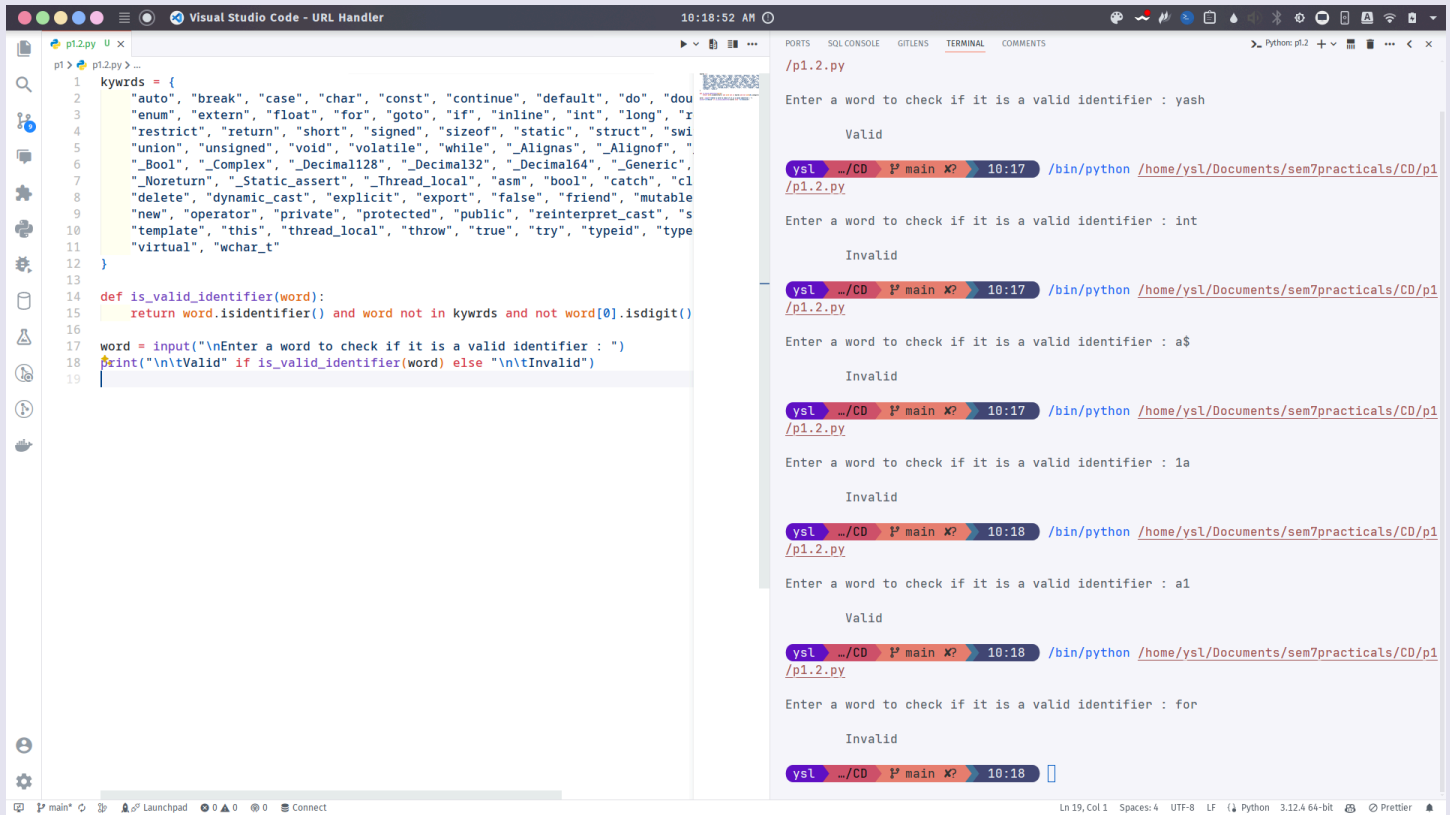
```
"_Generic", "_Imaginary",
    "_Noreturn", "_Static_assert", "_Thread_local", "asm", "bool", "catch",
"class", "const_cast",
    "delete", "dynamic_cast", "explicit", "export", "false", "friend",
"mutable", "namespace",
    "new", "operator", "private", "protected", "public",
"reinterpret_cast", "static_assert",
    "template", "this", "thread_local", "throw", "true", "try", "typeid",
"typename", "using",
    "virtual", "wchar_t"
}

def is_valid_identifier(word):
    return word.isidentifier() and word not in kywrds and not
word[0].isdigit()

word = input("\nEnter a word to check if it is a valid identifier : ")
print("\n\tValid" if is_valid_identifier(word) else "\n\tInvalid")
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA Batch - 71  
CD Practical 1

Output :



The screenshot displays the Visual Studio Code interface with a Python file named `p1.2.py` open in the editor. The script defines a list of keywords and a function to check if a word is a valid identifier. The terminal shows the execution of the script with various inputs and their corresponding outputs.

```
1 keywords = {
2     "auto", "break", "case", "char", "const", "continue", "default", "do", "dou
3     "enum", "extern", "float", "for", "goto", "if", "inline", "int", "long", "r
4     "restrict", "return", "short", "signed", "sizeof", "static", "struct", "swi
5     "union", "unsigned", "void", "volatile", "while", "_Alignas", "_Alignof", "
6     "_Bool", "_Complex", "_Decimal128", "_Decimal32", "_Decimal64", "_Generic",
7     "_Noreturn", "_Static_assert", "_Thread_local", "asm", "bool", "catch", "cl
8     "delete", "dynamic_cast", "explicit", "export", "false", "friend", "mutable
9     "new", "operator", "private", "protected", "public", "reinterpret_cast", "s
10    "template", "this", "thread_local", "throw", "true", "try", "typeid", "type
11    "virtual", "wchar_t"
12 }
13
14 def is_valid_identifier(word):
15     return word.isidentifier() and word not in keywords and not word[0].isdigit()
16
17 word = input("\nEnter a word to check if it is a valid identifier : ")
18 print("\n\tValid" if is_valid_identifier(word) else "\n\tInvalid")
19
```

The terminal output shows the following sequence of interactions:

```
/p1.2.py
Enter a word to check if it is a valid identifier : yash
Valid
y$ ./CD P main X? 10:17 /bin/python /home/y$l/Documents/sem7practicals/CD/p1
/p1.2.py
Enter a word to check if it is a valid identifier : int
Invalid
y$ ./CD P main X? 10:17 /bin/python /home/y$l/Documents/sem7practicals/CD/p1
/p1.2.py
Enter a word to check if it is a valid identifier : a$
Invalid
y$ ./CD P main X? 10:17 /bin/python /home/y$l/Documents/sem7practicals/CD/p1
/p1.2.py
Enter a word to check if it is a valid identifier : 1a
Invalid
y$ ./CD P main X? 10:18 /bin/python /home/y$l/Documents/sem7practicals/CD/p1
/p1.2.py
Enter a word to check if it is a valid identifier : a1
Valid
y$ ./CD P main X? 10:18 /bin/python /home/y$l/Documents/sem7practicals/CD/p1
/p1.2.py
Enter a word to check if it is a valid identifier : for
Invalid
y$ ./CD P main X? 10:18
```

The status bar at the bottom indicates the file is at line 19, column 1, with 4 spaces, UTF-8 encoding, LF line endings, Python 3.12.4 64-bit, and Prettier formatting.