Practical - 1

Aim : Understanding of language processing activities through sample C code which does following operations:

1. Preprocessed code
2. Assembly Code
3. Object Code
4. Executable Code

Program:

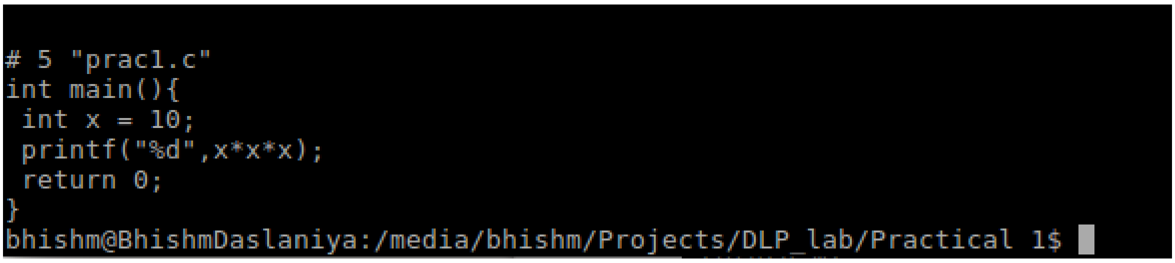
Filename : Prac1.c

```
#include<stdio.h>
#define sq(x) x*x
#define cube(x) sq(x)*x

int main(){
        int x = 10;
        printf("%d",cube(x));
        return 0;
}
```

To Generate preprocessed code: cpp prac1.c > prac1.i

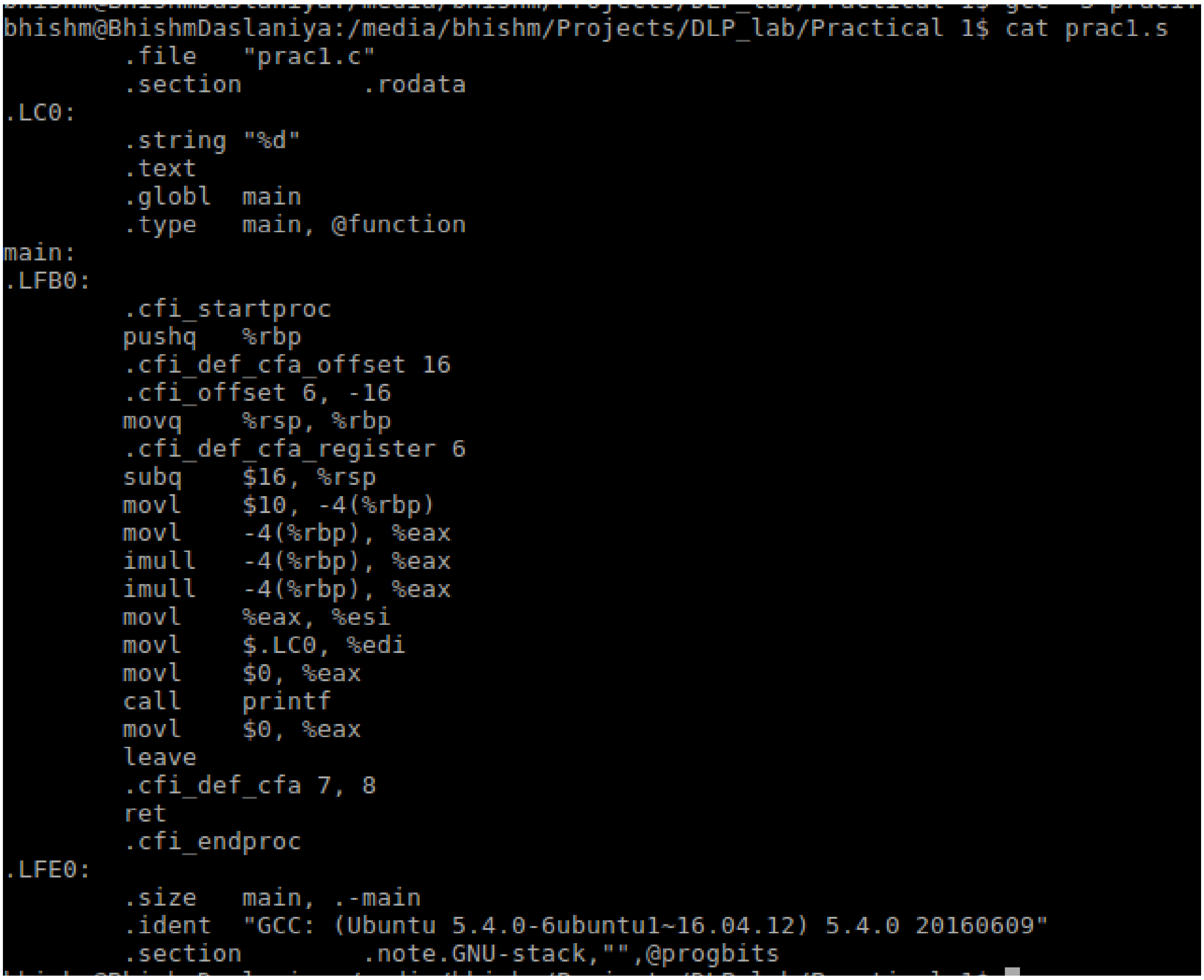
Output of Preprocessor:



```
# 5 "prac1.c"
int main(){
 int x = 10;
 printf("%d",x*x*x);
 return 0;
}
bhishm@BhishmDaslaniya:/media/bhishm/Projects/DLP_lab/Practical 1$
```

To generate assembly code: gcc -S prac1.i

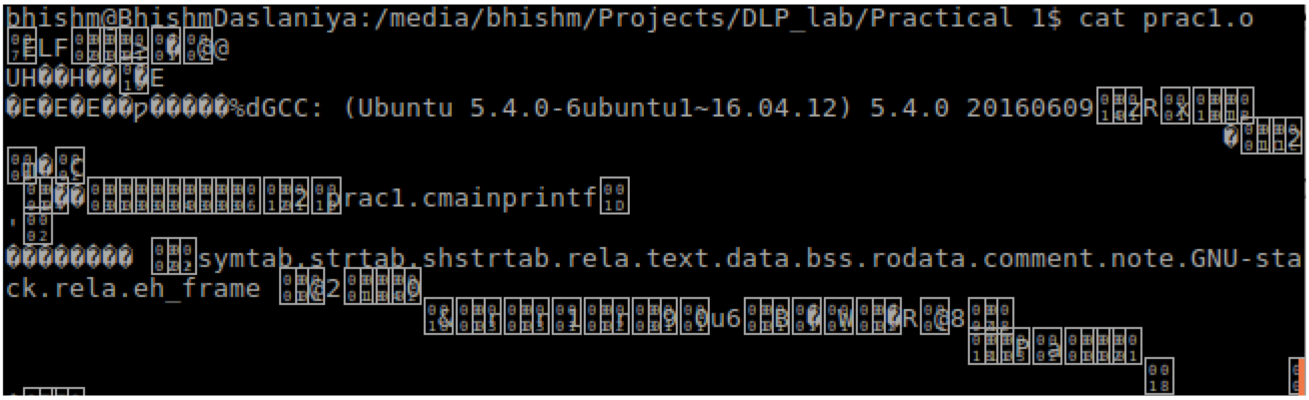
Output of compiler:



```
bhishm@BhishmDaslaniya:/media/bhishm/Projects/DLP_lab/Practical 1$ cat prac1.s
        .file   "prac1.c"
        .section        .rodata
.LC0:
        .string "%d"
        .text
        .globl  main
        .type   main, @function
main:
.LFB0:
        .cfi_startproc
        pushq   %rbp
        .cfi_def_cfa_offset 16
        .cfi_offset 6, -16
        movq    %rsp, %rbp
        .cfi_def_cfa_register 6
        subq    $16, %rsp
        movl    $10, -4(%rbp)
        movl    -4(%rbp), %eax
        imull   -4(%rbp), %eax
        imull   -4(%rbp), %eax
        movl    %eax, %esi
        movl    $.LC0, %edi
        movl    $0, %eax
        call    printf
        movl    $0, %eax
        leave
        .cfi_def_cfa 7, 8
        ret
        .cfi_endproc
.LFE0:
        .size   main, .-main
        .ident  "GCC: (Ubuntu 5.4.0-6ubuntu1~16.04.12) 5.4.0 20160609"
        .section        .note.GNU-stack,"",@progbits
```

To generate object code: as -o prac1.o prac1.s

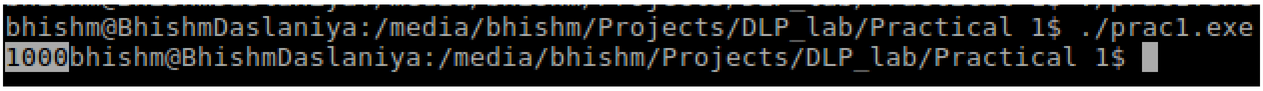
Output of assembler:



```
bhishm@BhishmDaslaniya:/media/bhishm/Projects/DLP_lab/Practical 1$ cat prac1.o
�ELF�����@�@
UH��H���E
�E�E�E��p�����%dGCC: (Ubuntu 5.4.0-6ubuntu1~16.04.12) 5.4.0 20160609��zR�x���
                                                                          ����2
��C
 ����������������2�prac1.cmainprintf�
,�
�������� ��symtab.strtab.shstrtab.rela.text.data.bss.rodata.comment.note.GNU-sta
ck.rela.eh_frame ��@2����@
                              �&��������1�����9�0u6��B���W���R�@8��
                                                                   ���P�a�����
```

To generate executable code: gcc prac1.o -o prac1.exe

Output of executable file:



```
bhishm@BhishmDaslaniya:/media/bhishm/Projects/DLP_lab/Practical 1$ ./prac1.exe
1000bhishm@BhishmDaslaniya:/media/bhishm/Projects/DLP_lab/Practical 1$
```

Conclusion: From this practical I have learnt about internal processing of language processor and also learnt about different stages of generating the output of code.