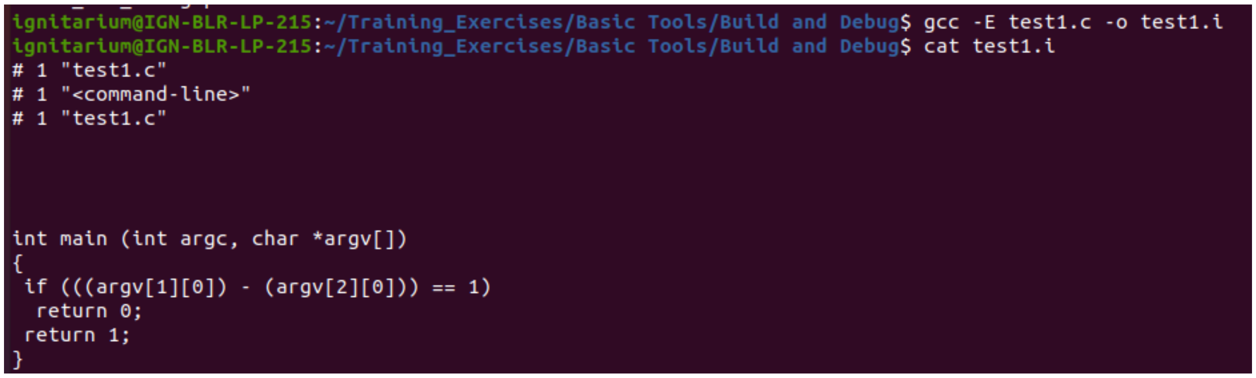
Build and Debug

4.1 GCC Exercises

1. 1st question solution

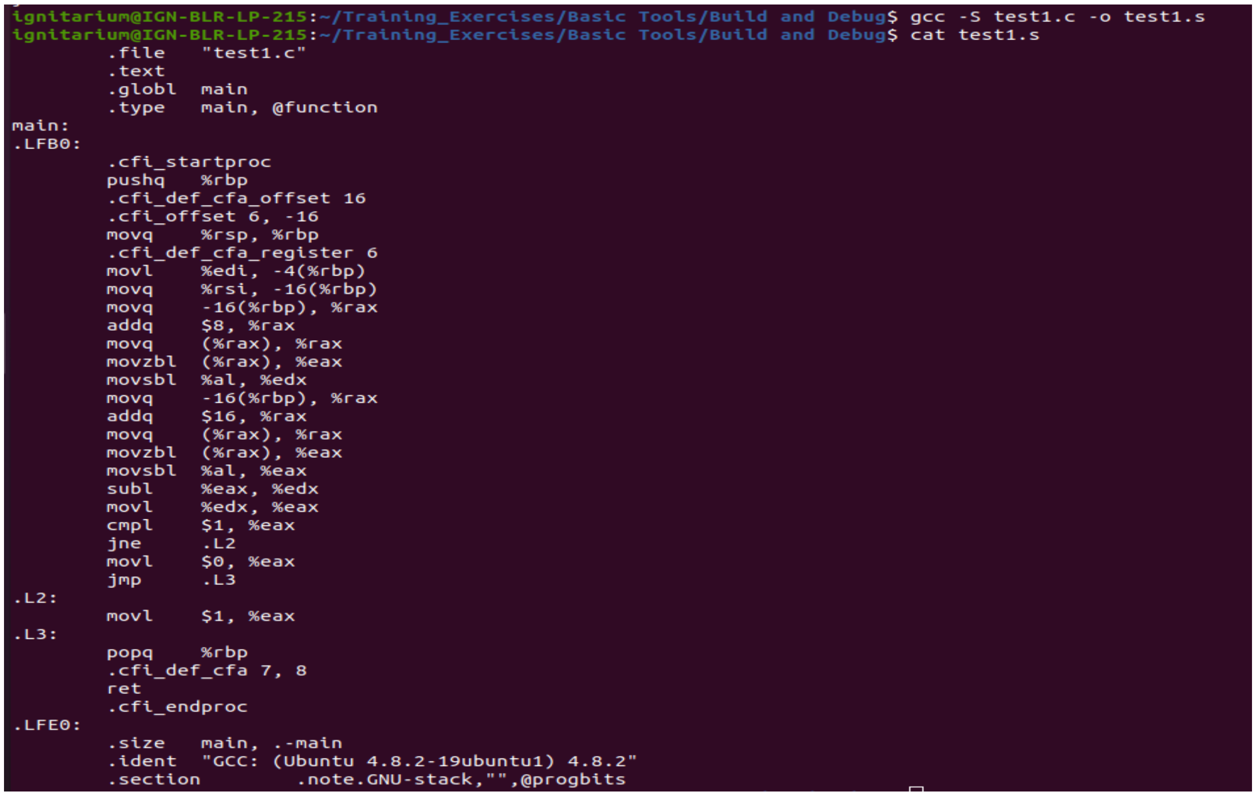
Created test1.c program

Run preprocessor only



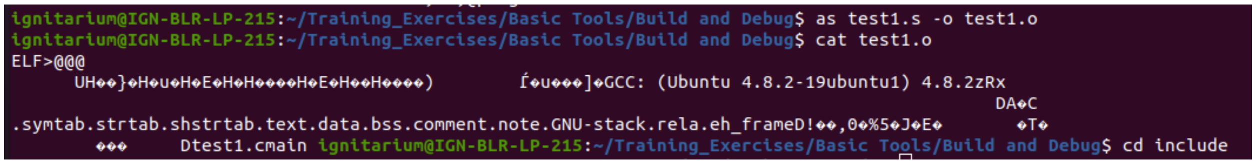
```
ignitarium@IGN-BLR-LP-215:~/Training_Exercises/Basic Tools/Build and Debug$ gcc -E test1.c -o test1.i
ignitarium@IGN-BLR-LP-215:~/Training_Exercises/Basic Tools/Build and Debug$ cat test1.i
# 1 "test1.c"
# 1 "<command-line>"
# 1 "test1.c"




int main (int argc, char *argv[])
{
 if (((argv[1][0]) - (argv[2][0])) == 1)
  return 0;
 return 1;
}
```

Compile only without assembling



```
ignitarium@IGN-BLR-LP-215:~/Training_Exercises/Basic Tools/Build and Debug$ gcc -S test1.c -o test1.s
ignitarium@IGN-BLR-LP-215:~/Training_Exercises/Basic Tools/Build and Debug$ cat test1.s
        .file   "test1.c"
        .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
        movl    %edi, -4(%rbp)
        movq    %rsi, -16(%rbp)
        movq    -16(%rbp), %rax
        addq    $8, %rax
        movq    (%rax), %rax
        movzbl  (%rax), %eax
        movsbl  %al, %edx
        movq    -16(%rbp), %rax
        addq    $16, %rax
        movq    (%rax), %rax
        movzbl  (%rax), %eax
        movsbl  %al, %eax
        subl    %eax, %edx
        movl    %edx, %eax
        cmpl    $1, %eax
        jne     .L2
        movl    $0, %eax
        jmp     .L3
.L2:
        movl    $1, %eax
.L3:
        popq    %rbp
        .cfi_def_cfa 7, 8
        ret
        .cfi_endproc
.LFE0:
        .size   main, .-main
        .ident  "GCC: (Ubuntu 4.8.2-19ubuntu1) 4.8.2"
        .section        .note.GNU-stack,"",@progbits
```

Assemble only without linking

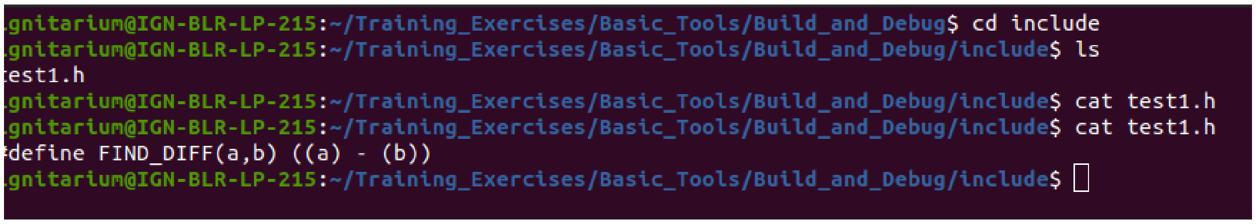


```
ignitarium@IGN-BLR-LP-215:~/Training_Exercises/Basic Tools/Build and Debug$ as test1.s -o test1.o
ignitarium@IGN-BLR-LP-215:~/Training_Exercises/Basic Tools/Build and Debug$ cat test1.o
ELF>@@@
      UH��}�H�u�H�E�H�H����H�E�H��H����)        ſ�u���]�GCC: (Ubuntu 4.8.2-19ubuntu1) 4.8.2zRx
                                                                                             DA�C
.symtab.strtab.shstrtab.text.data.bss.comment.note.GNU-stack.rela.eh_frameD!��,0�%5�J�E�       �T�
        ���     Dtest1.cmain ignitarium@IGN-BLR-LP-215:~/Training_Exercises/Basic Tools/Build and Debug$ cd include
```

—---

2. **2nd question solution**

Created a subdirectory named “include”.
Created a file ‘test1.h’ in that directory and included a macro in this file.
Now include this .h file in the . c file and remove the “DIFFERENCE” from the c file.



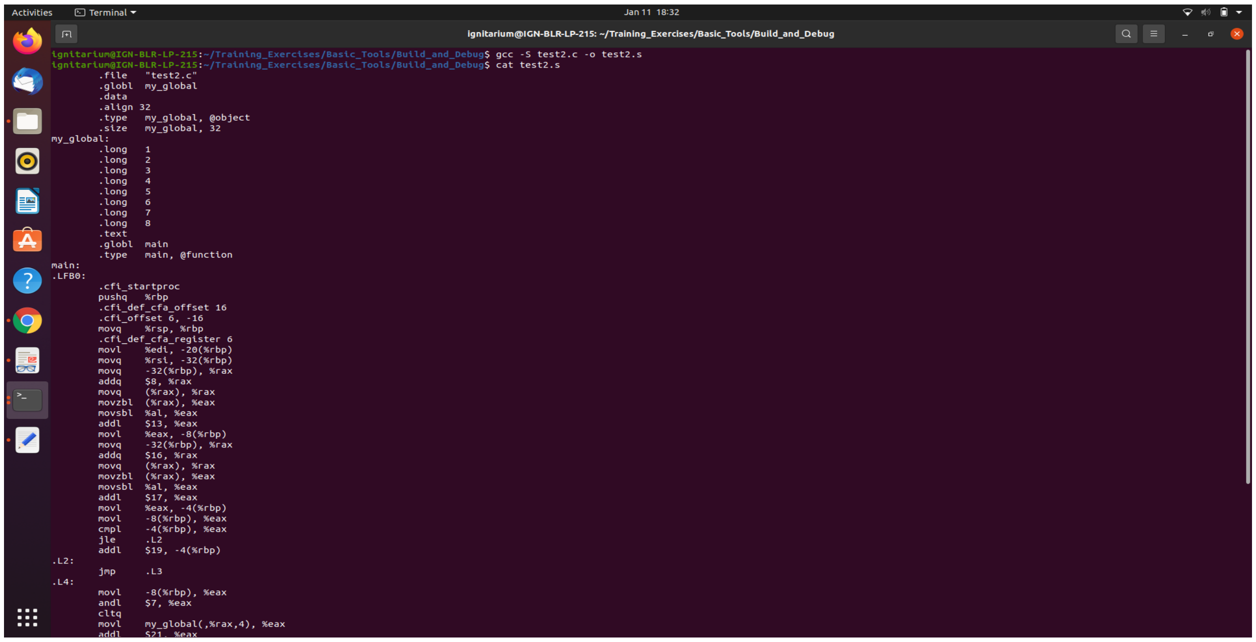
```
gnitarium@IGN-BLR-LP-215:~/Training_Exercises/Basic_Tools/Build_and_Debug$ cd include
gnitarium@IGN-BLR-LP-215:~/Training_Exercises/Basic_Tools/Build_and_Debug/include$ ls
est1.h
gnitarium@IGN-BLR-LP-215:~/Training_Exercises/Basic_Tools/Build_and_Debug/include$ cat test1.h
gnitarium@IGN-BLR-LP-215:~/Training_Exercises/Basic_Tools/Build_and_Debug/include$ cat test1.h
define FIND_DIFF(a,b) ((a) - (b))
gnitarium@IGN-BLR-LP-215:~/Training_Exercises/Basic_Tools/Build_and_Debug/include$ 
```

Command line to pass file1.h file
gcc file1.c -I (path to .h file)

—---

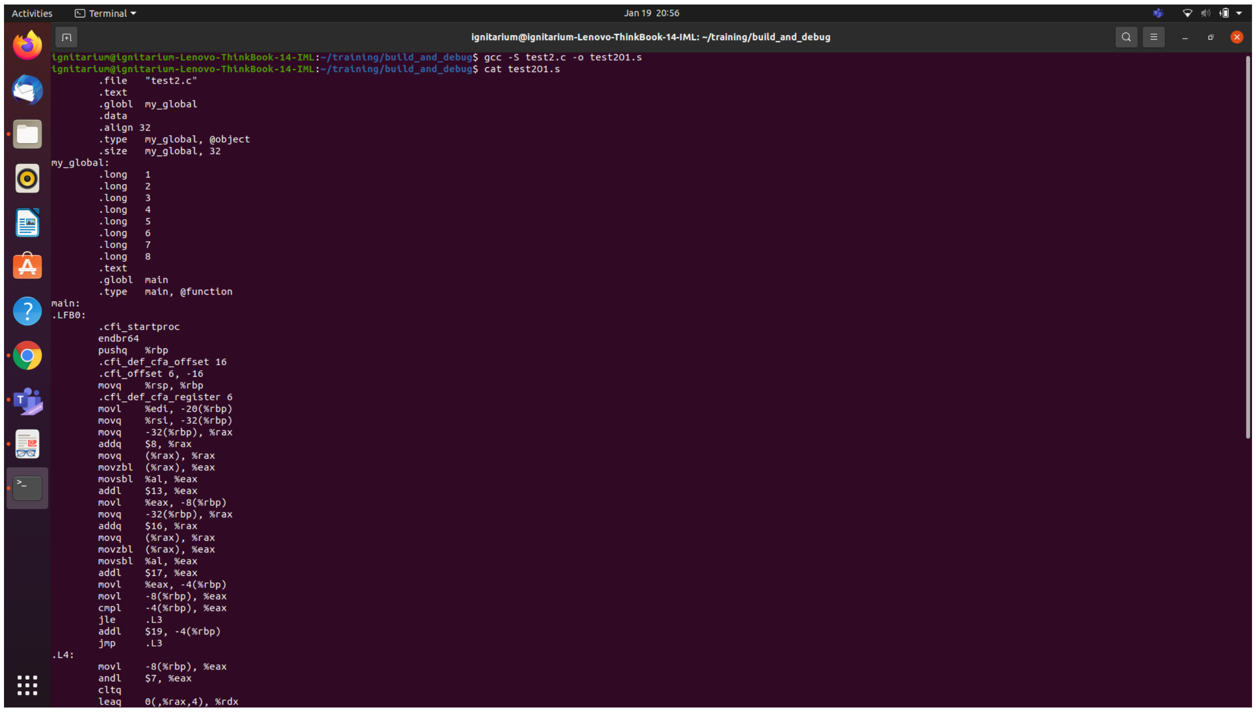
3. **3rd question solution**

<u>Compile without assemble</u>


Activities
Terminal
Jan 11 18:32
ignitarium@IGN-BLR-LP-215: ~/Training_Exercises/Basic_Tools/Build_and_Debug

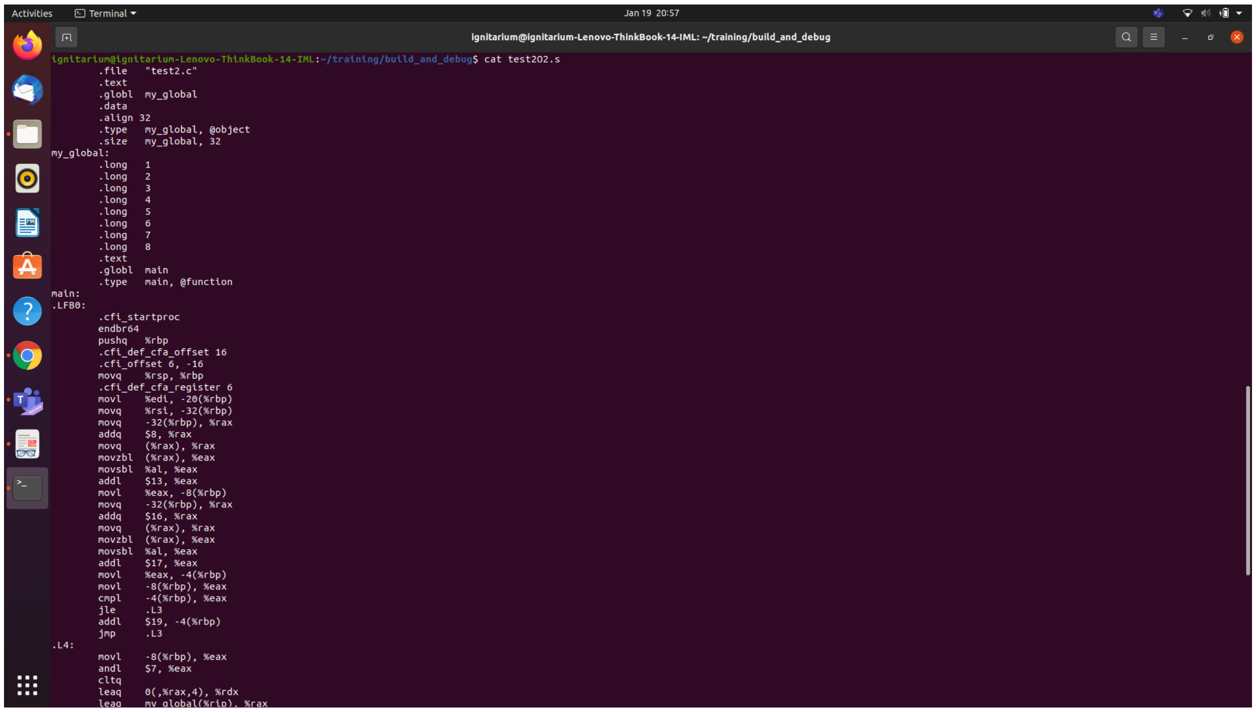
```
ignitarium@IGN-BLR-LP-215:~/Training_Exercises/Basic_Tools/Build_and_Debug$ gcc -S test2.c -o test2.s
ignitarium@IGN-BLR-LP-215:~/Training_Exercises/Basic_Tools/Build_and_Debug$ cat test2.s
        .file   "test2.c"
        .globl  my_global
        .data
        .align 32
        .type   my_global, @object
        .size   my_global, 32
my_global:
        .long   1
        .long   2
        .long   3
        .long   4
        .long   5
        .long   6
        .long   7
        .long   8
        .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
        movl    %edi, -20(%rbp)
        movq    %rsi, -32(%rbp)
        movq    -32(%rbp), %rax
        addq    $8, %rax
        movq    (%rax), %rax
        movzbl  (%rax), %eax
        movsbl  %al, %eax
        addl    $13, %eax
        movl    %eax, -8(%rbp)
        movq    -32(%rbp), %rax
        addq    $16, %rax
        movq    (%rax), %rax
        movzbl  (%rax), %eax
        movsbl  %al, %eax
        addl    $17, %eax
        movl    %eax, -4(%rbp)
        movl    -8(%rbp), %eax
        cmpl    -4(%rbp), %eax
        jle     .L2
        addl    $19, -4(%rbp)
.L2:
        jmp     .L3
.L4:
        movl    -8(%rbp), %eax
        andl    $7, %eax
        cltq
        movl    my_global(,%rax,4), %eax
        addl    $21, %eax
```

Optimization level 1 and inspecting assembly code



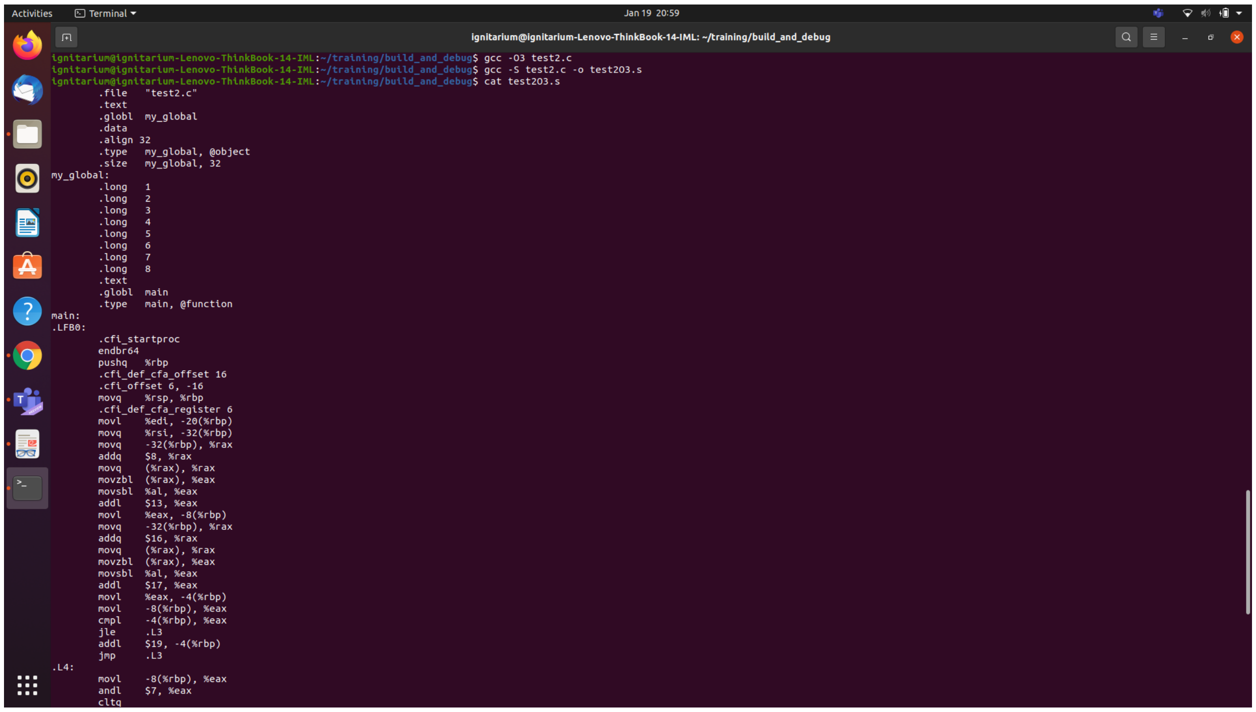
```
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug$ gcc -S test2.c -o test201.s
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug$ cat test201.s
        .file   "test2.c"
        .text
        .globl  my_global
        .data
        .align 32
        .type   my_global, @object
        .size   my_global, 32
my_global:
        .long   1
        .long   2
        .long   3
        .long   4
        .long   5
        .long   6
        .long   7
        .long   8
        .text
        .globl  main
        .type   main, @function
main:
.LFB0:
        .cfi_startproc
        endbr64
        pushq   %rbp
        .cfi_def_cfa_offset 16
        .cfi_offset 6, -16
        movq    %rsp, %rbp
        .cfi_def_cfa_register 6
        movl    %edi, -20(%rbp)
        movq    %rsi, -32(%rbp)
        movq    -32(%rbp), %rax
        addq    $8, %rax
        movq    (%rax), %rax
        movzbl  (%rax), %eax
        movsbl  %al, %eax
        addl    $13, %eax
        movl    %eax, -8(%rbp)
        movq    -32(%rbp), %rax
        addq    $16, %rax
        movq    (%rax), %rax
        movzbl  (%rax), %eax
        movsbl  %al, %eax
        addl    $17, %eax
        movl    %eax, -4(%rbp)
        movl    -8(%rbp), %eax
        cmpl    -4(%rbp), %eax
        jle     .L3
        addl    $19, -4(%rbp)
        jmp     .L3
.L4:
        movl    -8(%rbp), %eax
        andl    $7, %eax
        cltq
        leaq    0(,%rax,4), %rdx
```

Optimization level 2 and inspecting assembly code



```
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug$ cat test202.s
        .file   "test2.c"
        .text
        .globl  my_global
        .data
        .align 32
        .type   my_global, @object
        .size   my_global, 32
my_global:
        .long   1
        .long   2
        .long   3
        .long   4
        .long   5
        .long   6
        .long   7
        .long   8
        .text
        .globl  main
        .type   main, @function
main:
.LFB0:
        .cfi_startproc
        endbr64
        pushq   %rbp
        .cfi_def_cfa_offset 16
        .cfi_offset 6, -16
        movq    %rsp, %rbp
        .cfi_def_cfa_register 6
        movl    %edi, -20(%rbp)
        movq    %rsi, -32(%rbp)
        movq    -32(%rbp), %rax
        addq    $8, %rax
        movq    (%rax), %rax
        movzbl  (%rax), %eax
        movsbl  %al, %eax
        addl    $13, %eax
        movl    %eax, -8(%rbp)
        movq    -32(%rbp), %rax
        addq    $16, %rax
        movq    (%rax), %rax
        movzbl  (%rax), %eax
        movsbl  %al, %eax
        addl    $17, %eax
        movl    %eax, -4(%rbp)
        movl    -8(%rbp), %eax
        cmpl    -4(%rbp), %eax
        jle     .L3
        addl    $19, -4(%rbp)
        jmp     .L3
.L4:
        movl    -8(%rbp), %eax
        andl    $7, %eax
        cltq
        leaq    0(,%rax,4), %rdx
        leaq    my_global(%rip), %rax
```

Optimization level 3 and inspecting assembly code

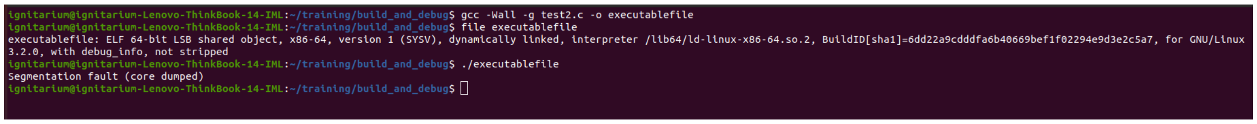


```
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug$ gcc -O3 test2.c
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug$ gcc -S test2.c -o test2O3.s
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug$ cat test2O3.s
        .file   "test2.c"
        .text
        .globl  my_global
        .data
        .align 32
        .type   my_global, @object
        .size   my_global, 32
my_global:
        .long   1
        .long   2
        .long   3
        .long   4
        .long   5
        .long   6
        .long   7
        .long   8
        .text
        .globl  main
        .type   main, @function
main:
.LFB0:
        .cfi_startproc
        endbr64
        pushq   %rbp
        .cfi_def_cfa_offset 16
        .cfi_offset 6, -16
        movq    %rsp, %rbp
        .cfi_def_cfa_register 6
        movl    %edi, -20(%rbp)
        movq    %rsi, -32(%rbp)
        movq    -32(%rbp), %rax
        addq    $8, %rax
        movq    (%rax), %rax
        movzbl  (%rax), %eax
        movsbl  %al, %eax
        addl    $13, %eax
        movl    %eax, -8(%rbp)
        movq    -32(%rbp), %rax
        addq    $16, %rax
        movq    (%rax), %rax
        movzbl  (%rax), %eax
        movsbl  %al, %eax
        addl    $17, %eax
        movl    %eax, -4(%rbp)
        movl    -8(%rbp), %eax
        cmpl    -4(%rbp), %eax
        jle     .L3
        addl    $19, -4(%rbp)
        jmp     .L3
.L4:
        movl    -8(%rbp), %eax
        andl    $7, %eax
        cltq
```

—--

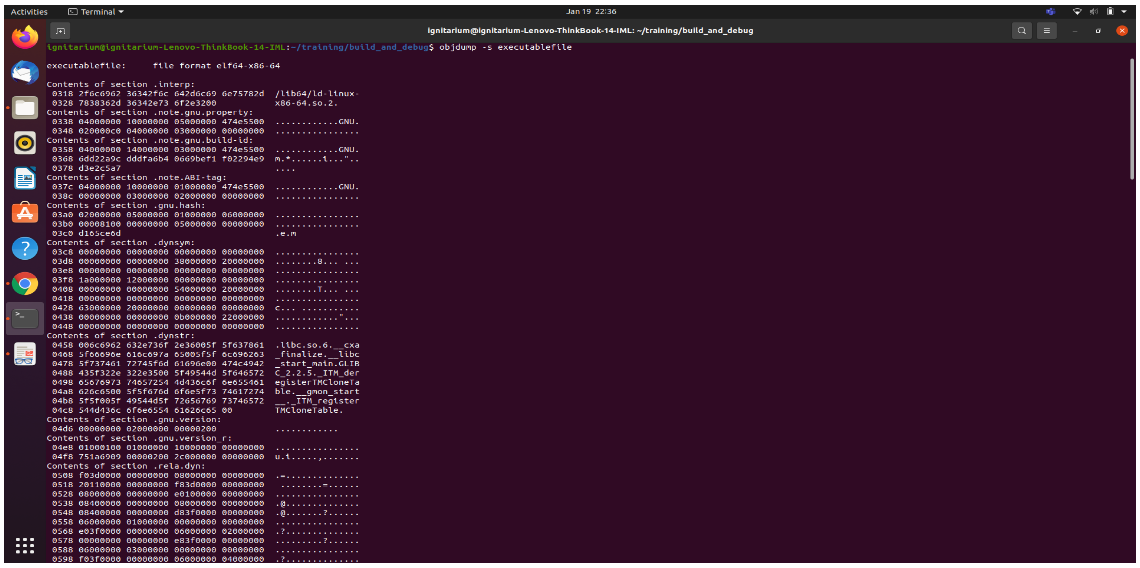
4. 4th question solution

Generated the ELF executable file for test2.c and ran the executable file.



```
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug$ gcc -Wall -g test2.c -o executablefile
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug$ file executablefile
executablefile: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=6dd22a9cdddfa6b40669bef1f02294e9d3e2c5a7, for GNU/Linux
3.2.0, with debug_info, not stripped
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug$ ./executablefile
Segmentation fault (core dumped)
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug$ 
```

a) Objdump utility



```
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug$ objdump -s executablefile

executablefile:     file format elf64-x86-64

Contents of section .interp:
 0318 2f6c6962 36342f6c 642d6c69 6e75782d  /lib64/ld-linux-
 0328 7838362d 36342e73 6f2e3200           x86-64.so.2.
Contents of section .note.gnu.property:
 0338 04000000 10000000 05000000 474e5500  ............GNU.
 0348 020000c0 04000000 03000000 00000000  ................
Contents of section .note.gnu.build-id:
 0358 04000000 14000000 03000000 474e5500  ............GNU.
 0368 6dd22a9c dddfa6b4 0669bef1 f02294e9  m.*......i..."..
 0378 d3e2c5a7                             ....
Contents of section .note.ABI-tag:
 037c 04000000 10000000 01000000 474e5500  ............GNU.
 038c 00000000 03000000 02000000 00000000  ................
Contents of section .gnu.hash:
 03a0 02000000 05000000 01000000 06000000  ................
 03b0 00008100 00000000 05000000 00000000  ................
 03c0 d165ce6d                             .e.m
Contents of section .dynsym:
 03c8 00000000 00000000 00000000 00000000  ................
 03d8 00000000 00000000 38000000 20000000  ........8... ...
 03e8 00000000 00000000 00000000 00000000  ................
 03f8 1a000000 12000000 00000000 00000000  ................
 0408 00000000 00000000 54000000 20000000  ........T... ...
 0418 00000000 00000000 00000000 00000000  ................
 0428 63000000 20000000 00000000 00000000  c... ...........
 0438 00000000 00000000 0b000000 22000000  ............"...
 0448 00000000 00000000 00000000 00000000  ................
Contents of section .dynstr:
 0458 006c6962 632e736f 2e36005f 5f637861  .libc.so.6.__cxa
 0468 5f66696e 616c697a 65005f5f 6c696263  _finalize.__libc
 0478 5f737461 72745f6d 61696e00 474c4942  _start_main.GLIB
 0488 435f322e 322e3500 5f49544d 5f646572  C_2.2.5._ITM_der
 0498 65676973 74657254 4d436c6f 6e655461  egisterTMCloneTa
 04a8 626c6500 5f5f676d 6f6e5f73 74617274  ble.__gmon_start
 04b8 5f5f005f 49544d5f 72656769 73746572  __._ITM_register
 04c8 544d436c 6f6e6554 61626c65 00        TMCloneTable.
Contents of section .gnu.version:
 04d6 00000000 02000000 00000200           ............
Contents of section .gnu.version_r:
 04e8 01000100 01000000 10000000 00000000  ................
 04f8 751a6909 00000200 2c000000 00000000  u.i.....,.......
Contents of section .rela.dyn:
 0508 f03d0000 00000000 08000000 00000000  .=..............
 0518 20110000 00000000 f83d0000 00000000   ........=......
 0528 08000000 00000000 e0100000 00000000  ................
 0538 08400000 00000000 08000000 00000000  .@..............
 0548 08400000 00000000 d83f0000 00000000  .@.......?......
 0558 06000000 01000000 00000000 00000000  ................
 0568 e03f0000 00000000 06000000 02000000  .?..............
 0578 00000000 00000000 e83f0000 00000000  .........?......
 0588 06000000 03000000 00000000 00000000  ................
 0598 f03f0000 00000000 06000000 04000000  .?..............
```

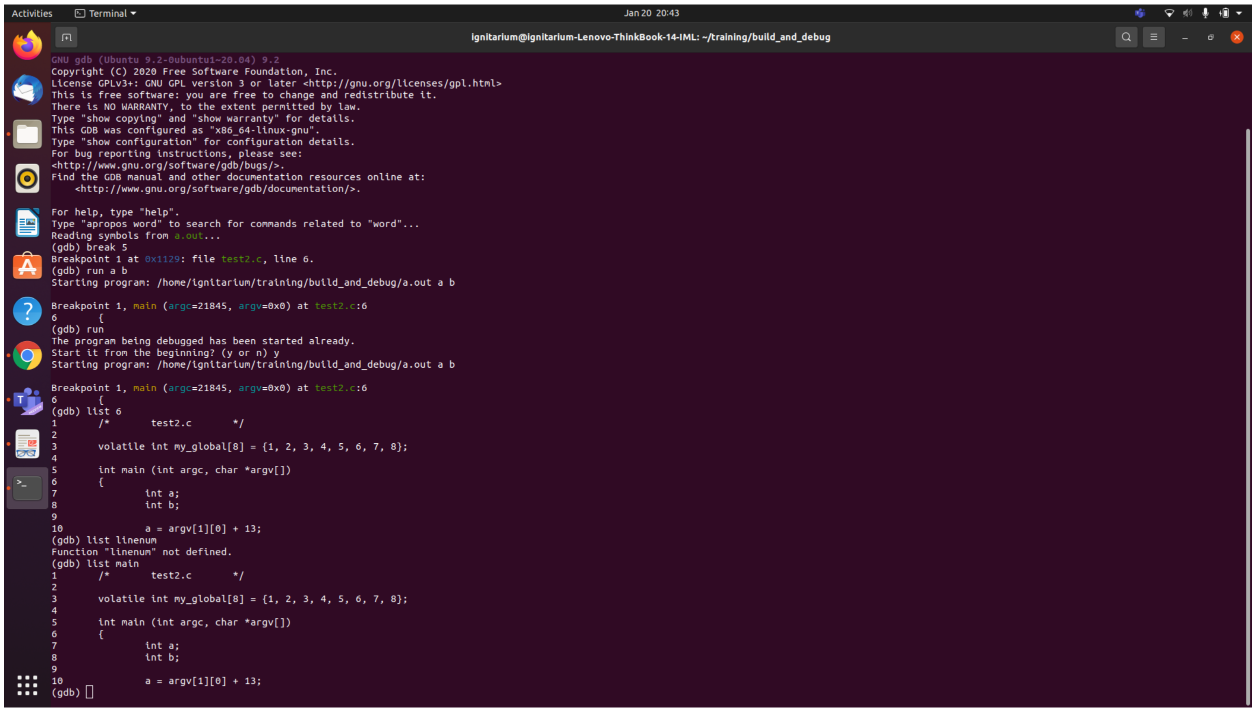
b) Generate hex file

gcc -c example.c objcopy --change-address 0xE0000 -O ihex example.o example.hex
it will generate example.hex file
use Bless Hex Editor to see the .hex file clearly

—--

4.2 GDB Exercises

1. 1st question solution



```
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from a.out...
(gdb) break 5
Breakpoint 1 at 0x1129: file test2.c, line 6.
(gdb) run a b
Starting program: /home/ignitarium/training/build_and_debug/a.out a b

Breakpoint 1, main (argc=21845, argv=0x0) at test2.c:6
6       {
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/ignitarium/training/build_and_debug/a.out a b

Breakpoint 1, main (argc=21845, argv=0x0) at test2.c:6
6       {
(gdb) list 6
1       /*      test2.c      */
2
3       volatile int my_global[8] = {1, 2, 3, 4, 5, 6, 7, 8};
4
5       int main (int argc, char *argv[])
6       {
7               int a;
8               int b;
9
10              a = argv[1][0] + 13;
(gdb) list linenum
Function "linenum" not defined.
(gdb) list main
1       /*      test2.c      */
2
3       volatile int my_global[8] = {1, 2, 3, 4, 5, 6, 7, 8};
4
5       int main (int argc, char *argv[])
6       {
7               int a;
8               int b;
9
10              a = argv[1][0] + 13;
(gdb) 
```

—--

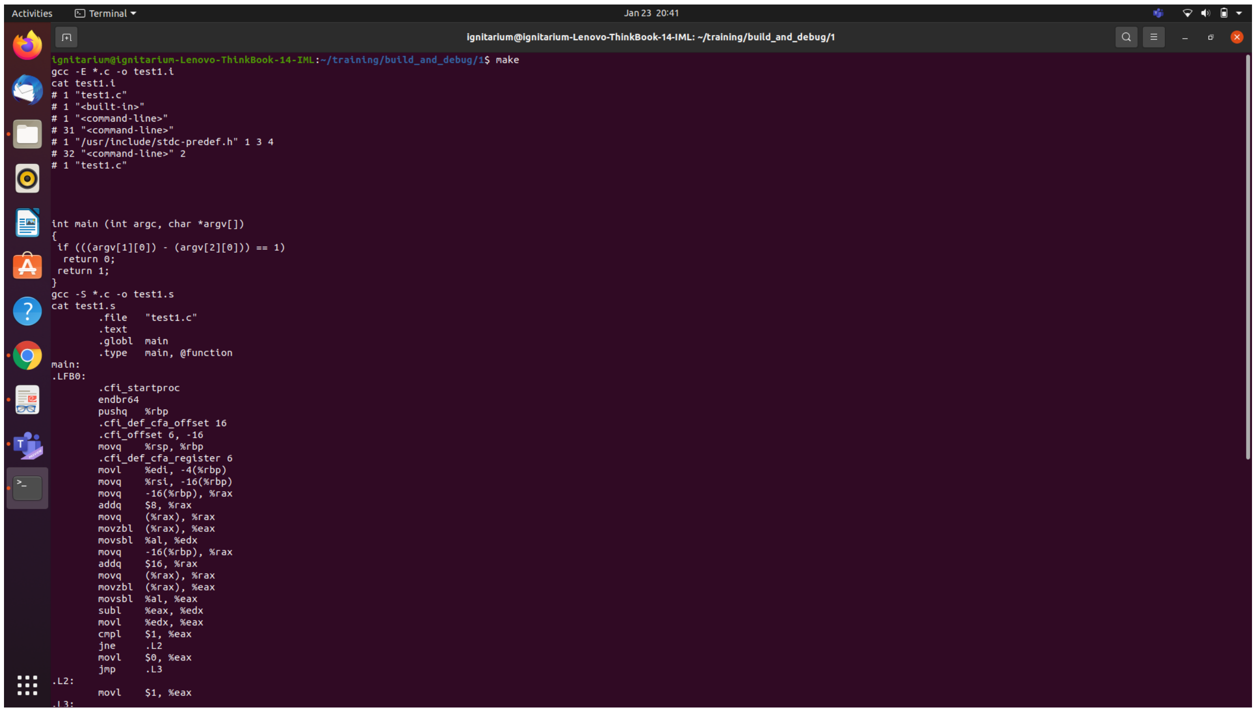
2. 2nd question solution

Enabling core dump
ulimit -S -c unlimited

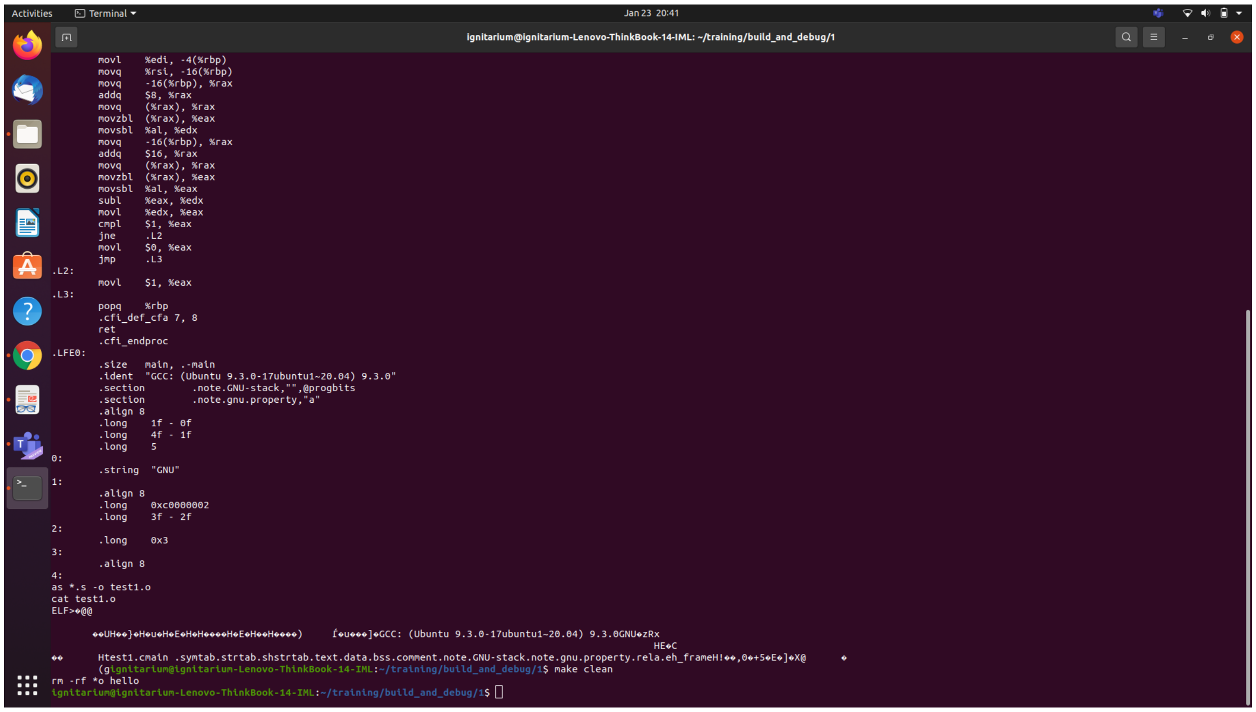
—--

4.3 GNU Make exercises

1. 1st question solution

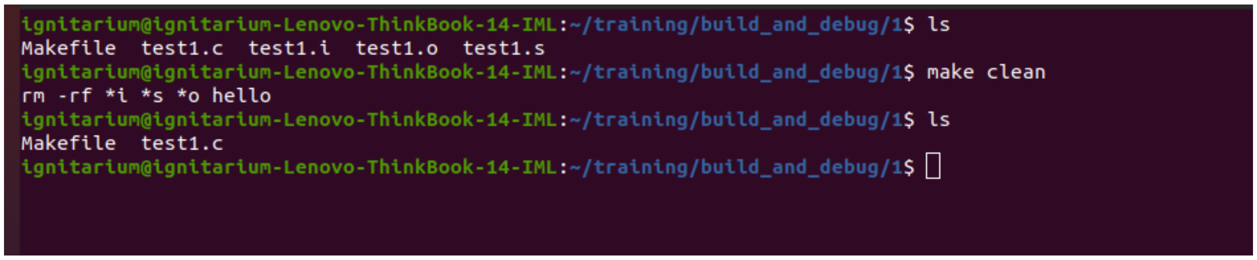

Activities
Terminal
Jan 23 20:41
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML: ~/training/build_and_debug/1

```
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug/1$ make
gcc -E *.c -o test1.i
cat test1.i
# 1 "test1.c"
# 1 "<built-in>"
# 1 "<command-line>"
# 31 "<command-line>"
# 1 "/usr/include/stdc-predef.h" 1 3 4
# 32 "<command-line>" 2
# 1 "test1.c"




int main (int argc, char *argv[])
{
 if (((argv[1][0]) - (argv[2][0])) == 1)
  return 0;
 return 1;
}
gcc -S *.c -o test1.s
cat test1.s
        .file   "test1.c"
        .text
        .globl  main
        .type   main, @function
main:
.LFB0:
        .cfi_startproc
        endbr64
        pushq   %rbp
        .cfi_def_cfa_offset 16
        .cfi_offset 6, -16
        movq    %rsp, %rbp
        .cfi_def_cfa_register 6
        movl    %edi, -4(%rbp)
        movq    %rsi, -16(%rbp)
        movq    -16(%rbp), %rax
        addq    $8, %rax
        movq    (%rax), %rax
        movzbl  (%rax), %eax
        movsbl  %al, %edx
        movq    -16(%rbp), %rax
        addq    $16, %rax
        movq    (%rax), %rax
        movzbl  (%rax), %eax
        movsbl  %al, %eax
        subl    %eax, %edx
        movl    %edx, %eax
        cmpl    $1, %eax
        jne     .L2
        movl    $0, %eax
        jmp     .L3
.L2:
        movl    $1, %eax
```


Activities
Terminal
Jan 23 20:41
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML: ~/training/build_and_debug/1

```
        movl    %edi, -4(%rbp)
        movq    %rsi, -16(%rbp)
        movq    -16(%rbp), %rax
        addq    $8, %rax
        movq    (%rax), %rax
        movzbl  (%rax), %eax
        movsbl  %al, %edx
        movq    -16(%rbp), %rax
        addq    $16, %rax
        movq    (%rax), %rax
        movzbl  (%rax), %eax
        movsbl  %al, %eax
        subl    %eax, %edx
        movl    %edx, %eax
        cmpl    $1, %eax
        jne     .L2
        movl    $0, %eax
        jmp     .L3
.L2:
        movl    $1, %eax
.L3:
        popq    %rbp
        .cfi_def_cfa 7, 8
        ret
        .cfi_endproc
.LFE0:
        .size   main, .-main
        .ident  "GCC: (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0"
        .section        .note.GNU-stack,"",@progbits
        .section        .note.gnu.property,"a"
        .align 8
        .long    1f - 0f
        .long    4f - 1f
        .long    5
0:
        .string  "GNU"
1:
        .align 8
        .long    0xc0000002
        .long    3f - 2f
2:
        .long    0x3
3:
        .align 8
4:
as *.s -o test1.o
cat test1.o
ELF>�@@

       ��UH��}�H�u�H�E�H�H����H�E�H��H����)     Í�u���]�GCC: (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0GNU�zRx
                                                                                                              HE�C
��      Htest1.cmain .symtab.strtab.shstrtab.text.data.bss.comment.note.GNU-stack.note.gnu.property.rela.eh_frameH!��,0�+5�E�]�X@      �
        (gignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug/1$ make clean
rm -rf *o hello
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug/1$ 
```

Cleaning target files



```
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug/1$ ls
Makefile  test1.c  test1.i  test1.o  test1.s
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug/1$ make clean
rm -rf *i *s *o hello
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug/1$ ls
Makefile  test1.c
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug/1$ 
```

—---

2. **2nd question solution**

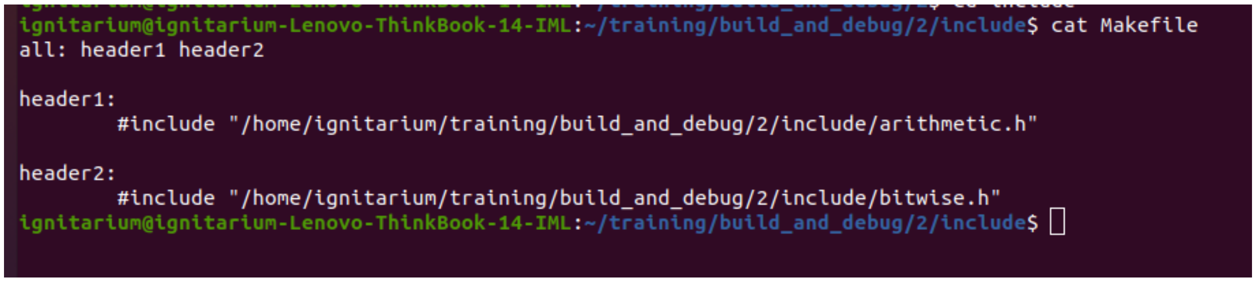
a) Created arithmetic.c and implemented add function, and exported that function into arithmetic.h

b) Created bitwise.c and implemented two functions shift_left and shift_right

c) Implemented app1.c

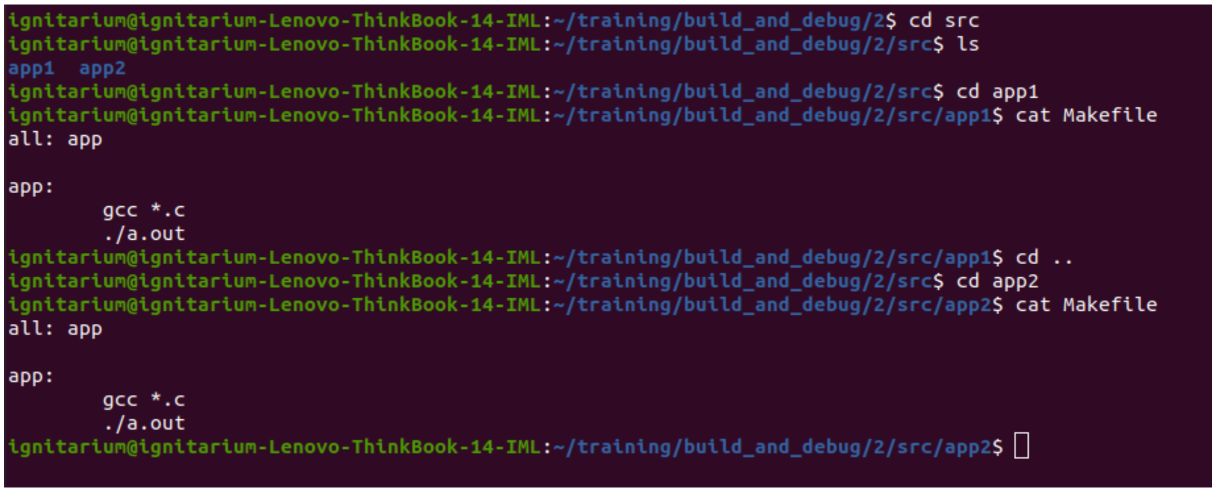
d) Implemented app2.c

e)



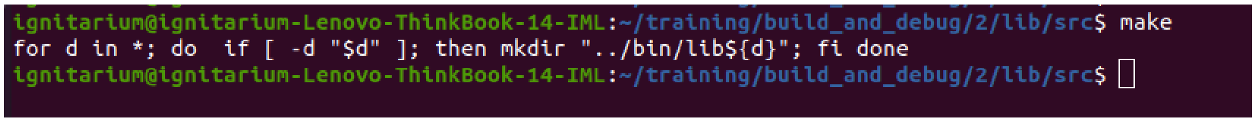
```
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug/2/include$ cat Makefile
all: header1 header2

header1:
        #include "/home/ignitarium/training/build_and_debug/2/include/arithmetic.h"

header2:
        #include "/home/ignitarium/training/build_and_debug/2/include/bitwise.h"
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug/2/include$ 
```

f)



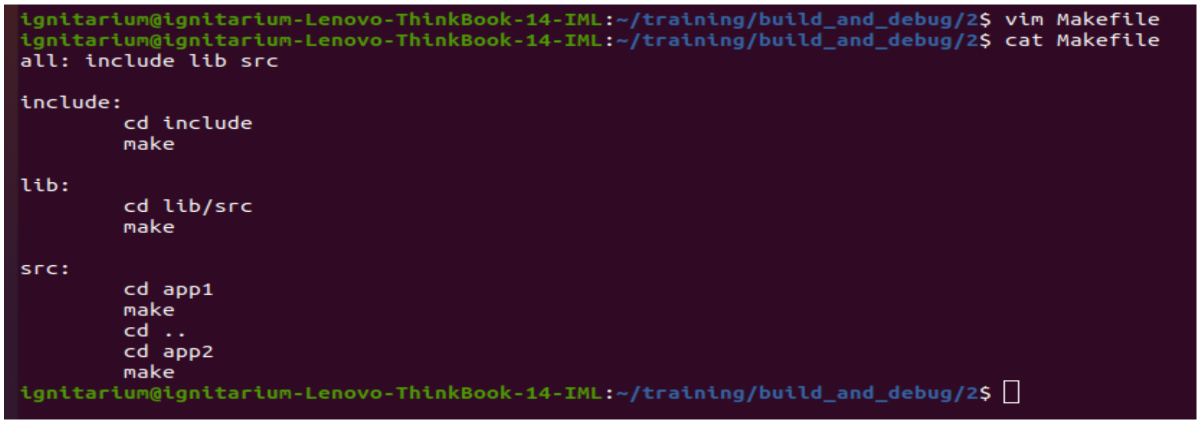
```
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug/2$ cd src
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug/2/src$ ls
app1  app2
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug/2/src$ cd app1
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug/2/src/app1$ cat Makefile
all: app

app:
        gcc *.c
        ./a.out
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug/2/src/app1$ cd ..
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug/2/src$ cd app2
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug/2/src/app2$ cat Makefile
all: app

app:
        gcc *.c
        ./a.out
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug/2/src/app2$ 
```

g)



```
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug/2/lib/src$ make
for d in *; do  if [ -d "$d" ]; then mkdir "../bin/lib${d}"; fi done
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug/2/lib/src$ 
```

h)



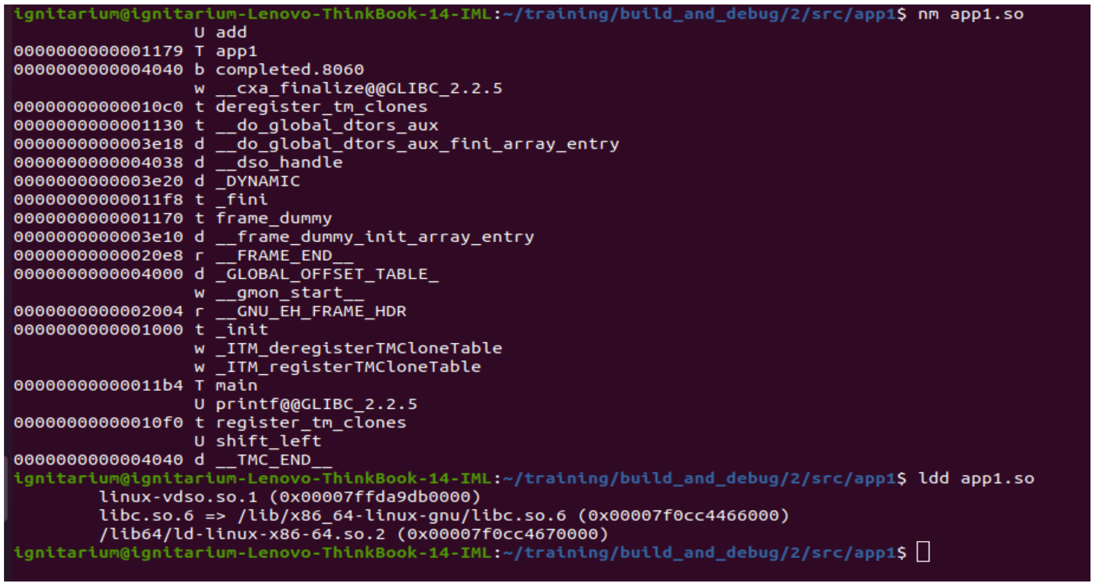
```
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug/2$ vim Makefile
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug/2$ cat Makefile
all: include lib src

include:
        cd include
        make

lib:
        cd lib/src
        make

src:
        cd app1
        make
        cd ..
        cd app2
        make
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug/2$ 
```

i) done

j) done

---

3. 3rd question solution



```
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug/2/src/app1$ nm app1.so
                 U add
0000000000001179 T app1
0000000000004040 b completed.8060
                 w __cxa_finalize@@GLIBC_2.2.5
00000000000010c0 t deregister_tm_clones
0000000000001130 t __do_global_dtors_aux
0000000000003e18 d __do_global_dtors_aux_fini_array_entry
0000000000004038 d __dso_handle
0000000000003e20 d _DYNAMIC
00000000000011f8 t _fini
0000000000001170 t frame_dummy
0000000000003e10 d __frame_dummy_init_array_entry
00000000000020e8 r __FRAME_END__
0000000000004000 d _GLOBAL_OFFSET_TABLE_
                 w __gmon_start__
0000000000002004 r __GNU_EH_FRAME_HDR
0000000000001000 t _init
                 w _ITM_deregisterTMCloneTable
                 w _ITM_registerTMCloneTable
00000000000011b4 T main
                 U printf@@GLIBC_2.2.5
00000000000010f0 t register_tm_clones
                 U shift_left
0000000000004040 d __TMC_END__
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug/2/src/app1$ ldd app1.so
        linux-vdso.so.1 (0x00007ffda9db0000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f0cc4466000)
        /lib64/ld-linux-x86-64.so.2 (0x00007f0cc4670000)
ignitarium@ignitarium-Lenovo-ThinkBook-14-IML:~/training/build_and_debug/2/src/app1$ 
```

THANK YOU