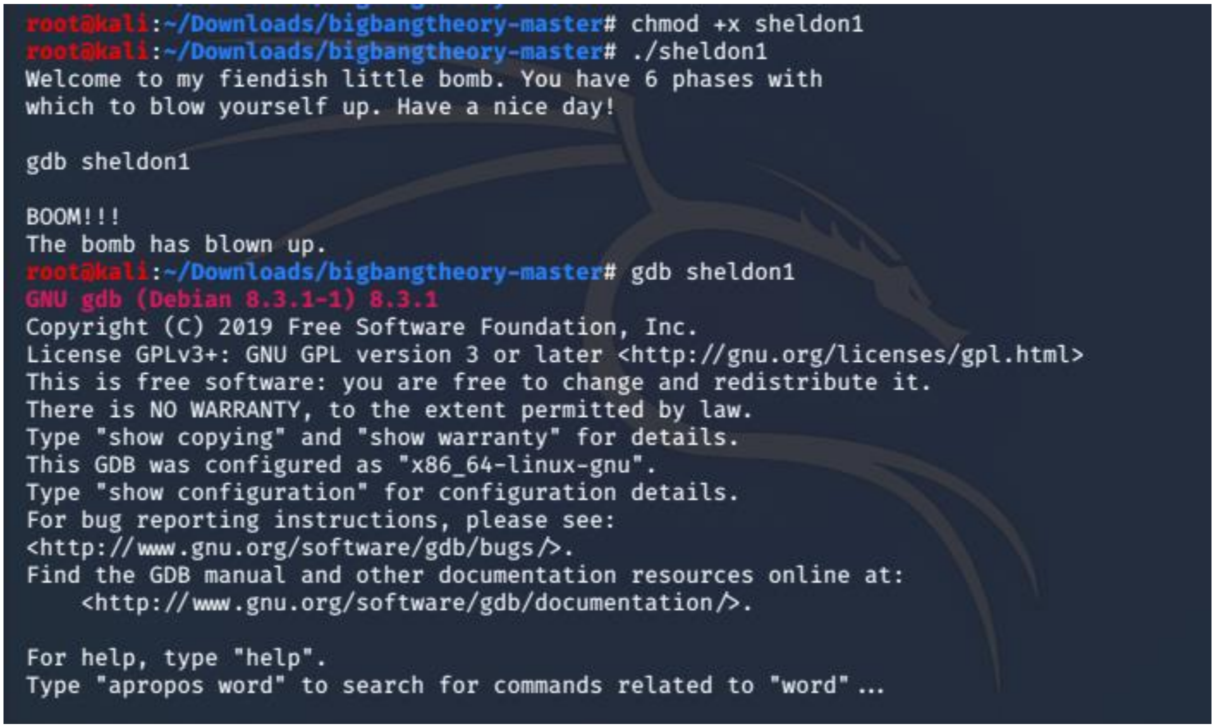
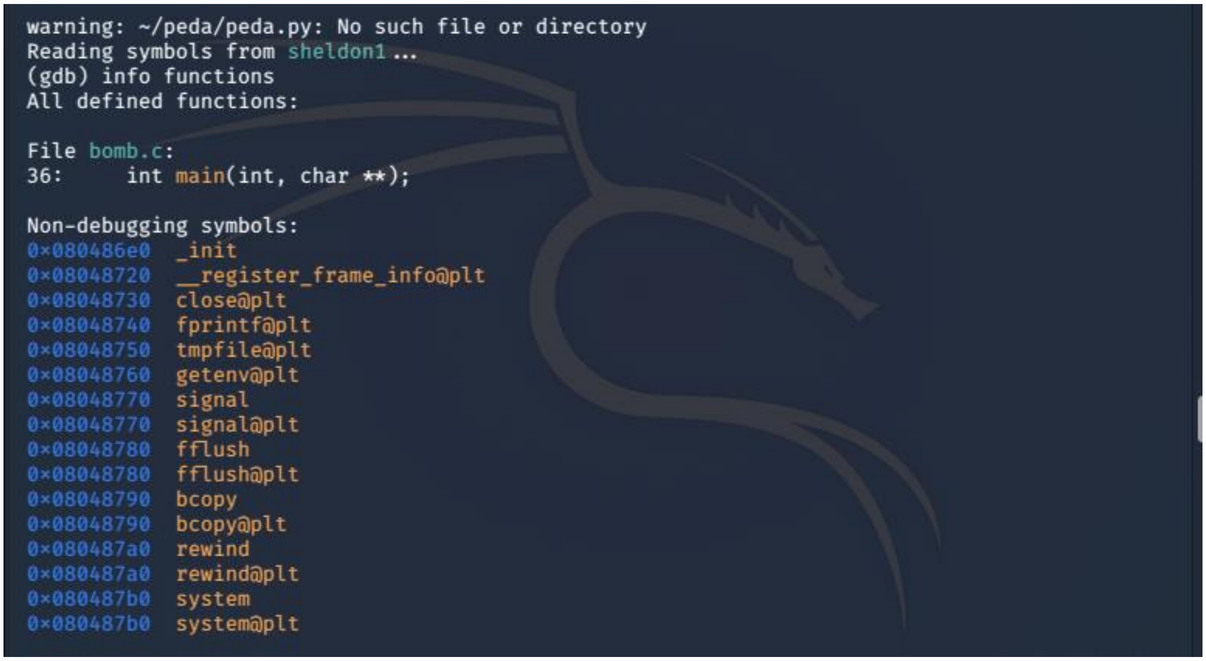
First run the sheldon1 file using “ chmod +x sheldon1” command and “./sheldon1” command.

Then open it using gdb.

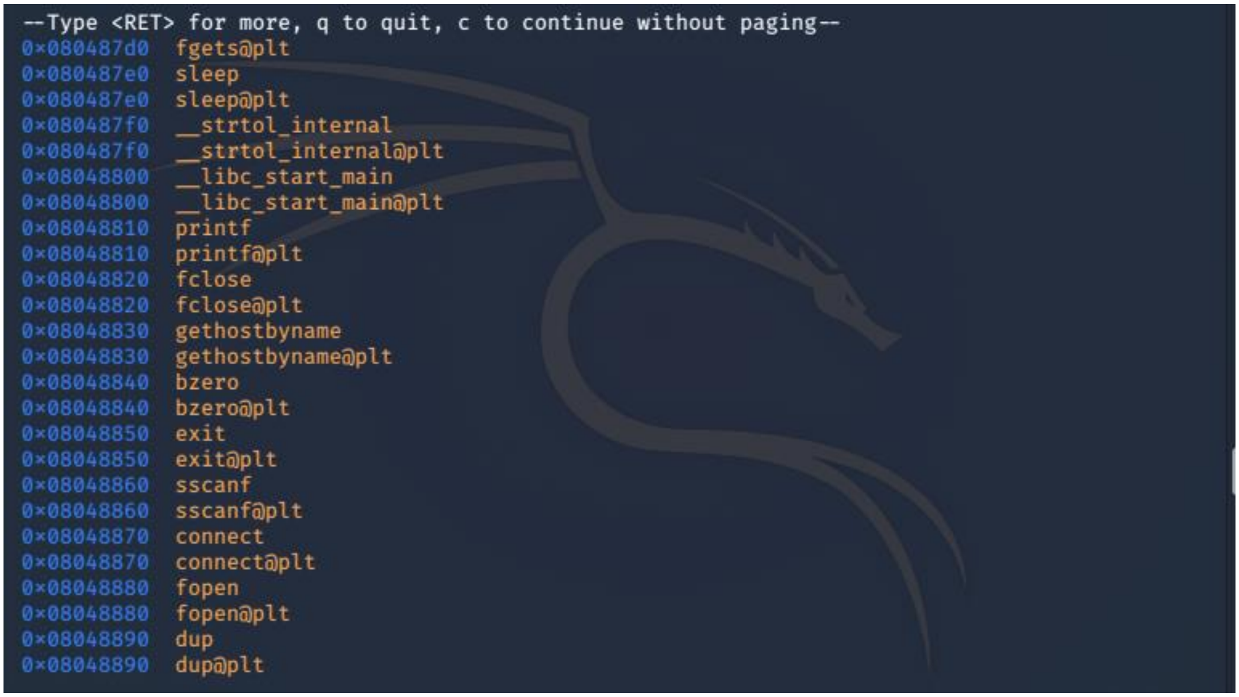


```
root@kali:~/Downloads/bigbangtheory-master# chmod +x sheldon1
root@kali:~/Downloads/bigbangtheory-master# ./sheldon1
Welcome to my fiendish little bomb. You have 6 phases with
which to blow yourself up. Have a nice day!

gdb sheldon1

BOOM!!!
The bomb has blown up.
root@kali:~/Downloads/bigbangtheory-master# gdb sheldon1
GNU gdb (Debian 8.3.1-1) 8.3.1
Copyright (C) 2019 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"...
```

Look for the functions inside sheldon1 using the command “info functions”.

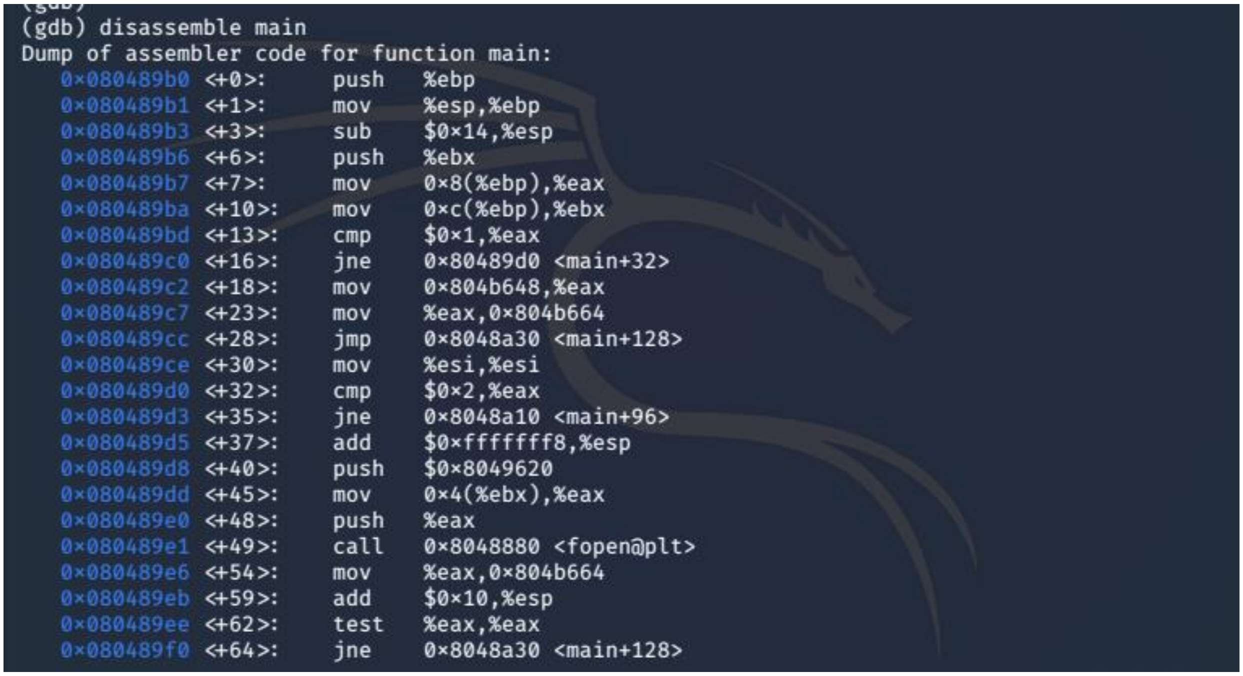


```
warning: ~/peda/peda.py: No such file or directory
Reading symbols from sheldon1...
(gdb) info functions
All defined functions:

File bomb.c:
36:     int main(int, char **);

Non-debugging symbols:
0x080486e0  _init
0x08048720  __register_frame_info@plt
0x08048730  close@plt
0x08048740  fprintf@plt
0x08048750  tmpfile@plt
0x08048760  getenv@plt
0x08048770  signal
0x08048770  signal@plt
0x08048780  fflush
0x08048780  fflush@plt
0x08048790  bcopy
0x08048790  bcopy@plt
0x080487a0  rewind
0x080487a0  rewind@plt
0x080487b0  system
0x080487b0  system@plt
```

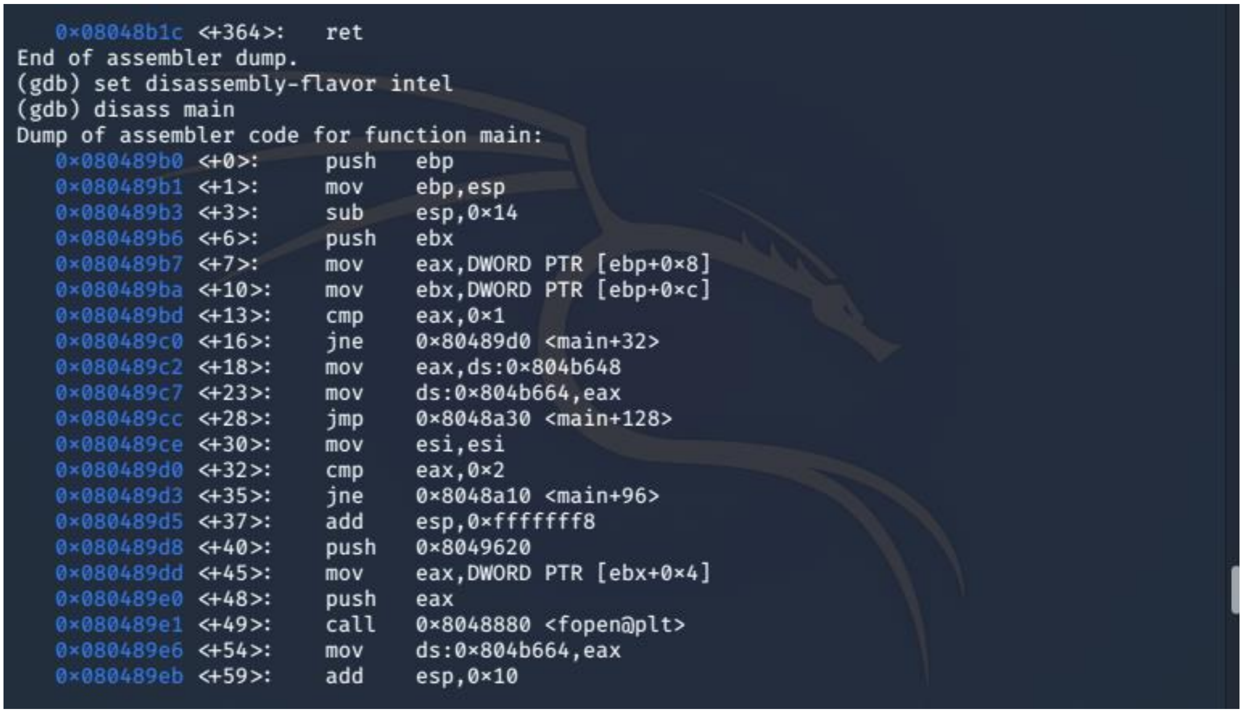


```
--Type <RET> for more, q to quit, c to continue without paging--
0x080487d0  fgets@plt
0x080487e0  sleep
0x080487e0  sleep@plt
0x080487f0  __strtol_internal
0x080487f0  __strtol_internal@plt
0x08048800  __libc_start_main
0x08048800  __libc_start_main@plt
0x08048810  printf
0x08048810  printf@plt
0x08048820  fclose
0x08048820  fclose@plt
0x08048830  gethostbyname
0x08048830  gethostbyname@plt
0x08048840  bzero
0x08048840  bzero@plt
0x08048850  exit
0x08048850  exit@plt
0x08048860  sscanf
0x08048860  sscanf@plt
0x08048870  connect
0x08048870  connect@plt
0x08048880  fopen
0x08048880  fopen@plt
0x08048890  dup
0x08048890  dup@plt
```

Then view the assembly code of the main function using “ disassemble main” command.

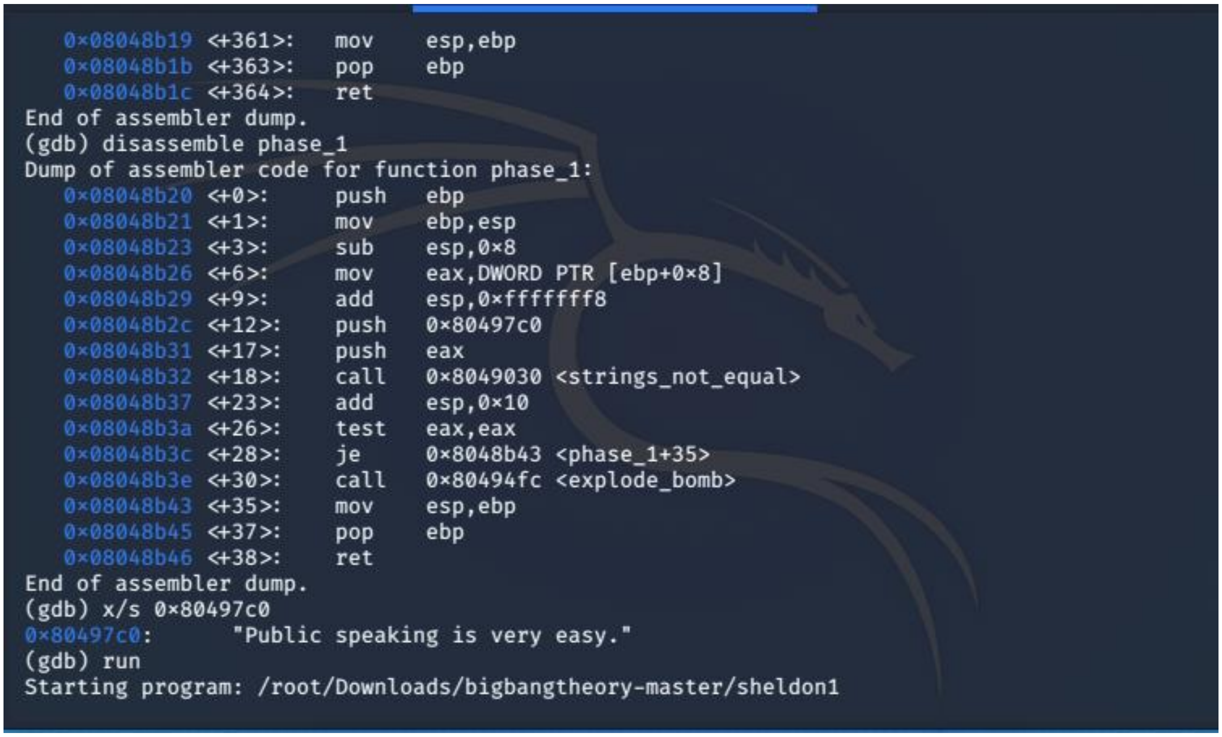


```
(gdb)
(gdb) disassemble main
Dump of assembler code for function main:
   0x080489b0 <+0>:     push   %ebp
   0x080489b1 <+1>:     mov    %esp,%ebp
   0x080489b3 <+3>:     sub    $0x14,%esp
   0x080489b6 <+6>:     push   %ebx
   0x080489b7 <+7>:     mov    0x8(%ebp),%eax
   0x080489ba <+10>:    mov    0xc(%ebp),%ebx
   0x080489bd <+13>:    cmp    $0x1,%eax
   0x080489c0 <+16>:    jne    0x80489d0 <main+32>
   0x080489c2 <+18>:    mov    0x804b648,%eax
   0x080489c7 <+23>:    mov    %eax,0x804b664
   0x080489cc <+28>:    jmp    0x8048a30 <main+128>
   0x080489ce <+30>:    mov    %esi,%esi
   0x080489d0 <+32>:    cmp    $0x2,%eax
   0x080489d3 <+35>:    jne    0x8048a10 <main+96>
   0x080489d5 <+37>:    add    $0xfffffff8,%esp
   0x080489d8 <+40>:    push   $0x8049620
   0x080489dd <+45>:    mov    0x4(%ebx),%eax
   0x080489e0 <+48>:    push   %eax
   0x080489e1 <+49>:    call   0x8048880 <fopen@plt>
   0x080489e6 <+54>:    mov    %eax,0x804b664
   0x080489eb <+59>:    add    $0x10,%esp
   0x080489ee <+62>:    test   %eax,%eax
   0x080489f0 <+64>:    jne    0x8048a30 <main+128>
```



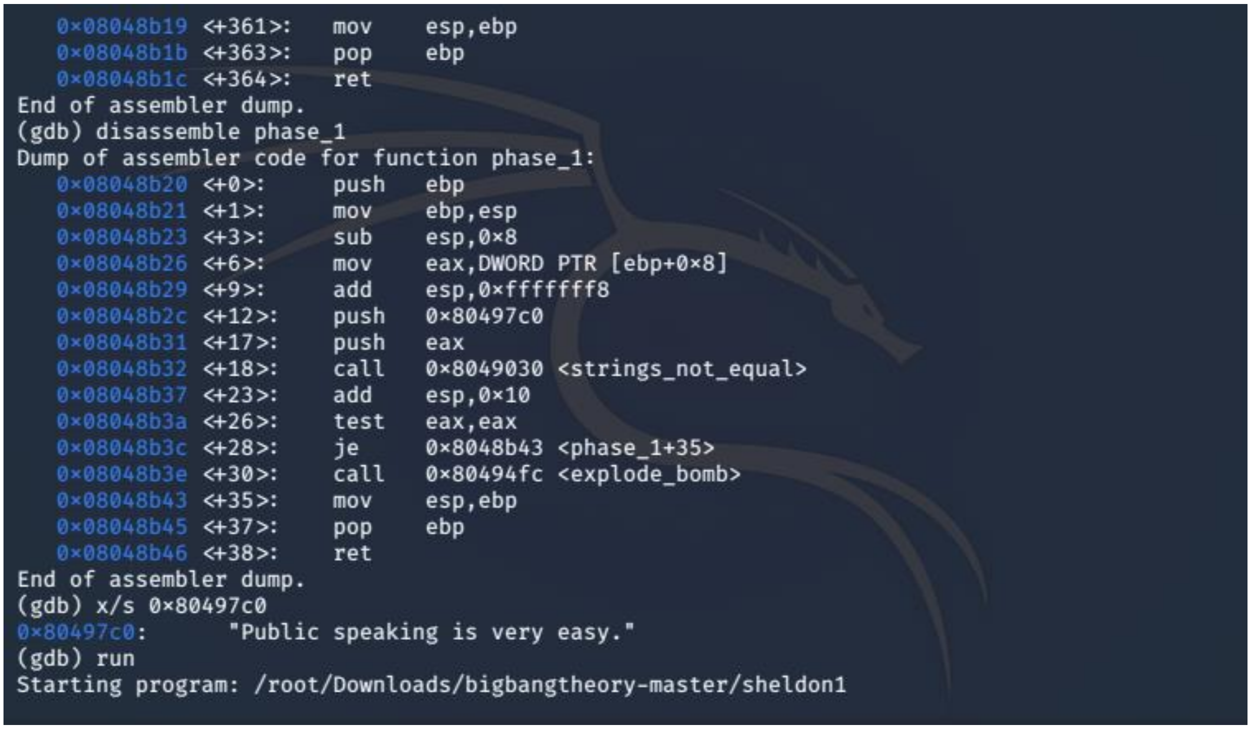
```
   0x08048b1c <+364>:   ret
End of assembler dump.
(gdb) set disassembly-flavor intel
(gdb) disass main
Dump of assembler code for function main:
   0x080489b0 <+0>:     push   ebp
   0x080489b1 <+1>:     mov    ebp,esp
   0x080489b3 <+3>:     sub    esp,0x14
   0x080489b6 <+6>:     push   ebx
   0x080489b7 <+7>:     mov    eax,DWORD PTR [ebp+0x8]
   0x080489ba <+10>:    mov    ebx,DWORD PTR [ebp+0xc]
   0x080489bd <+13>:    cmp    eax,0x1
   0x080489c0 <+16>:    jne    0x80489d0 <main+32>
   0x080489c2 <+18>:    mov    eax,ds:0x804b648
   0x080489c7 <+23>:    mov    ds:0x804b664,eax
   0x080489cc <+28>:    jmp    0x8048a30 <main+128>
   0x080489ce <+30>:    mov    esi,esi
   0x080489d0 <+32>:    cmp    eax,0x2
   0x080489d3 <+35>:    jne    0x8048a10 <main+96>
   0x080489d5 <+37>:    add    esp,0xfffffff8
   0x080489d8 <+40>:    push   0x8049620
   0x080489dd <+45>:    mov    eax,DWORD PTR [ebx+0x4]
   0x080489e0 <+48>:    push   eax
   0x080489e1 <+49>:    call   0x8048880 <fopen@plt>
   0x080489e6 <+54>:    mov    ds:0x804b664,eax
   0x080489eb <+59>:    add    esp,0x10
```

View the assembly code of the phase_1 using “ disassemble phase_1” command.

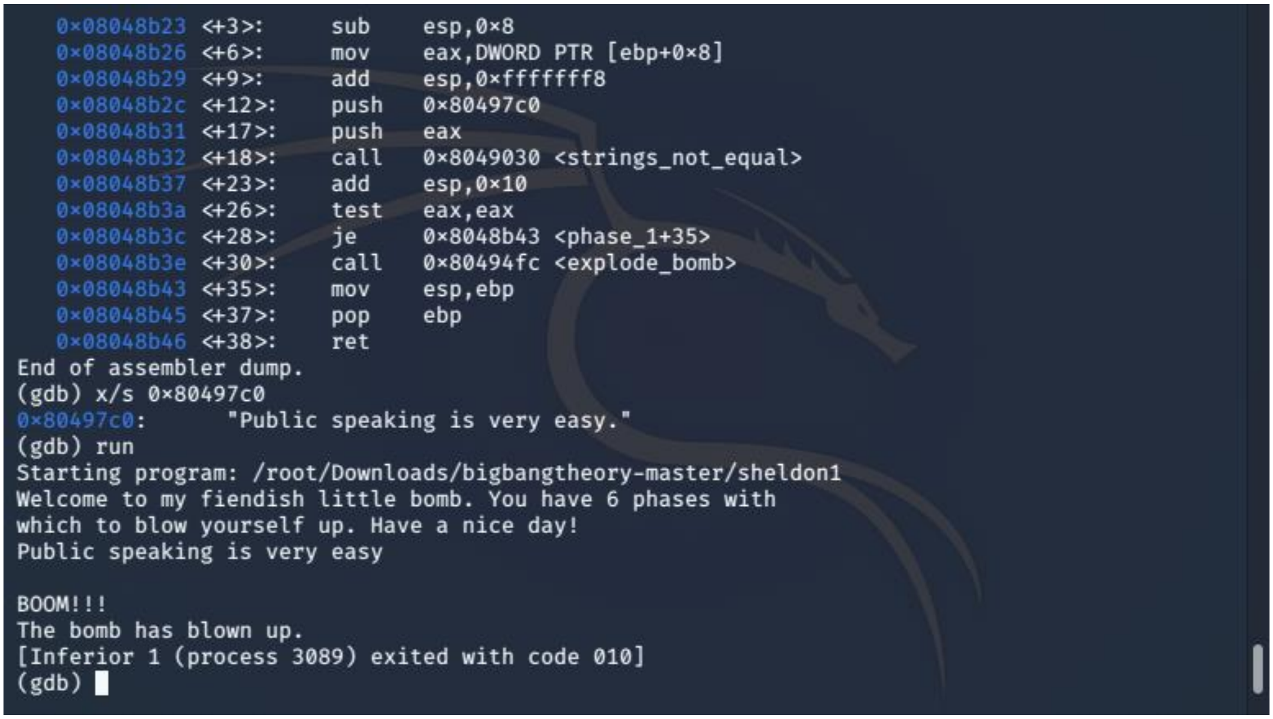


```
   0x08048b19 <+361>:   mov    esp,ebp
   0x08048b1b <+363>:   pop    ebp
   0x08048b1c <+364>:   ret
End of assembler dump.
(gdb) disassemble phase_1
Dump of assembler code for function phase_1:
   0x08048b20 <+0>:     push   ebp
   0x08048b21 <+1>:     mov    ebp,esp
   0x08048b23 <+3>:     sub    esp,0x8
   0x08048b26 <+6>:     mov    eax,DWORD PTR [ebp+0x8]
   0x08048b29 <+9>:     add    esp,0xfffffff8
   0x08048b2c <+12>:    push   0x80497c0
   0x08048b31 <+17>:    push   eax
   0x08048b32 <+18>:    call   0x8049030 <strings_not_equal>
   0x08048b37 <+23>:    add    esp,0x10
   0x08048b3a <+26>:    test   eax,eax
   0x08048b3c <+28>:    je     0x8048b43 <phase_1+35>
   0x08048b3e <+30>:    call   0x80494fc <explode_bomb>
   0x08048b43 <+35>:    mov    esp,ebp
   0x08048b45 <+37>:    pop    ebp
   0x08048b46 <+38>:    ret
End of assembler dump.
(gdb) x/s 0x80497c0
0x80497c0:      "Public speaking is very easy."
(gdb) run
Starting program: /root/Downloads/bigbangtheory-master/sheldon1
```

By looking into the 0x80497c0 memory location we could found the “Public speaking is very easy.” String.



```
   0×08048b19 <+361>:   mov    esp,ebp
   0×08048b1b <+363>:   pop    ebp
   0×08048b1c <+364>:   ret
End of assembler dump.
(gdb) disassemble phase_1
Dump of assembler code for function phase_1:
   0×08048b20 <+0>:     push   ebp
   0×08048b21 <+1>:     mov    ebp,esp
   0×08048b23 <+3>:     sub    esp,0×8
   0×08048b26 <+6>:     mov    eax,DWORD PTR [ebp+0×8]
   0×08048b29 <+9>:     add    esp,0×fffffff8
   0×08048b2c <+12>:    push   0×80497c0
   0×08048b31 <+17>:    push   eax
   0×08048b32 <+18>:    call   0×8049030 <strings_not_equal>
   0×08048b37 <+23>:    add    esp,0×10
   0×08048b3a <+26>:    test   eax,eax
   0×08048b3c <+28>:    je     0×8048b43 <phase_1+35>
   0×08048b3e <+30>:    call   0×80494fc <explode_bomb>
   0×08048b43 <+35>:    mov    esp,ebp
   0×08048b45 <+37>:    pop    ebp
   0×08048b46 <+38>:    ret
End of assembler dump.
(gdb) x/s 0×80497c0
0×80497c0:      "Public speaking is very easy."
(gdb) run
Starting program: /root/Downloads/bigbangtheory-master/sheldon1
```



```
   0×08048b23 <+3>:     sub    esp,0×8
   0×08048b26 <+6>:     mov    eax,DWORD PTR [ebp+0×8]
   0×08048b29 <+9>:     add    esp,0×fffffff8
   0×08048b2c <+12>:    push   0×80497c0
   0×08048b31 <+17>:    push   eax
   0×08048b32 <+18>:    call   0×8049030 <strings_not_equal>
   0×08048b37 <+23>:    add    esp,0×10
   0×08048b3a <+26>:    test   eax,eax
   0×08048b3c <+28>:    je     0×8048b43 <phase_1+35>
   0×08048b3e <+30>:    call   0×80494fc <explode_bomb>
   0×08048b43 <+35>:    mov    esp,ebp
   0×08048b45 <+37>:    pop    ebp
   0×08048b46 <+38>:    ret
End of assembler dump.
(gdb) x/s 0×80497c0
0×80497c0:      "Public speaking is very easy."
(gdb) run
Starting program: /root/Downloads/bigbangtheory-master/sheldon1
Welcome to my fiendish little bomb. You have 6 phases with
which to blow yourself up. Have a nice day!
Public speaking is very easy

BOOM!!!
The bomb has blown up.
[Inferior 1 (process 3089) exited with code 010]
(gdb) █
```