Linux Programming
Digital Assignment-2

Name:C Lokesh
Reg no:17MIS1088

1. Debugging a recurive function call program.

CODE:

```
#include<stdio.h>
int fibo(int i)
{
    if(i == 0 || i == 1)
    {
        return i;
    }
    return fibo(i-1) + fibo(i-2);
}
int main()
{
    int i;
    for(i=0;i<=10;i++)
    {
        printf("%d\n",fibo(i));
    }

}
```

Output:

```
lokesh@17mis1088:~/Desktop$ gcc -g -o o1 recursive.c
lokesh@17mis1088:~/Desktop$ gdb o1
GNU gdb (Ubuntu 8.3-0ubuntu1) 8.3
Copyright (C) 2019 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gp
l.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".
```

Running gdb

```
(gdb) b 12
Breakpoint 1 at 0x55555555519b: file recursive.c, line 13.
(gdb) run
Starting program: /home/lokesh/Desktop/o1

Breakpoint 1, main () at recursive.c:13
13          for(i=0;i<=10;i++)
(gdb) c
Continuing.
0
1
1
2
3
5
8
```

Setting breakpoint using line number

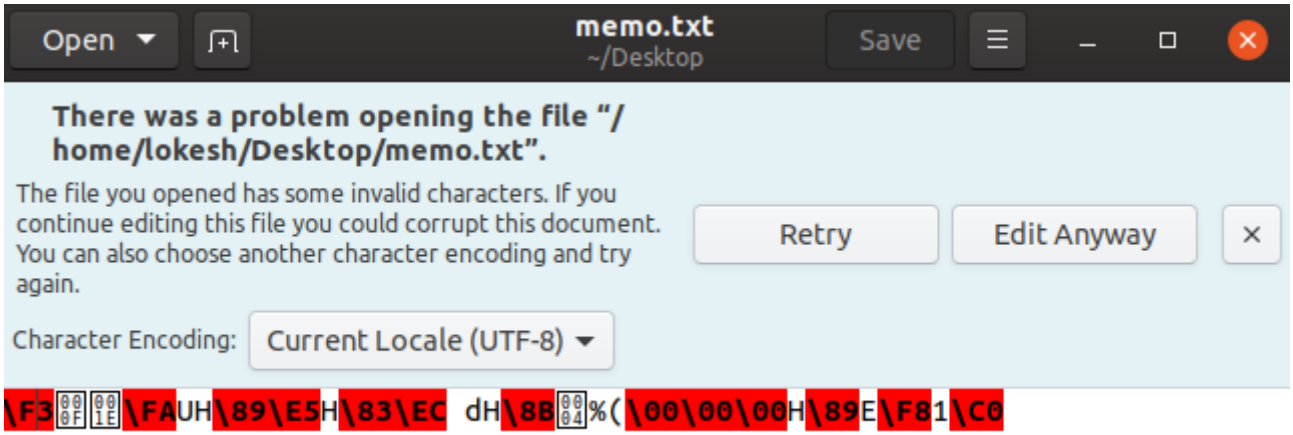
```
(gdb) b 12
Breakpoint 1 at 0x55555555519b: file recursive.c, line 13.
(gdb) run
Starting program: /home/lokesh/Desktop/o1

Breakpoint 1, main () at recursive.c:13
13          for(i=0;i<=10;i++)
(gdb) c
Continuing.
0
1
1
2
3
5
8
```

break point using address

```
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from o1...
(gdb) file o1
Load new symbol table from "o1"? (y or n) y
Reading symbols from o1...
(gdb) print fibo
$1 = {int (int)} 0x1149 <fibo>
(gdb) b *0x1149
Breakpoint 1 at 0x1149: file recursive.c, line 3.
(gdb) run
Starting program: /home/lokesh/Desktop/o1

[3]+  Stopped                 gdb o1
```

Memory Dumping

```
lokesh@17mis1088:~/Desktop$ gcc -g -o g1 memo.c
lokesh@17mis1088:~/Desktop$ gdb g1
GNU gdb (Ubuntu 8.3-0ubuntu1) 8.3
Copyright (C) 2019 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gp
l.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 g1...
(gdb) b 4
Breakpoint 1 at 0x1189: file memo.c, line 4.
(gdb) b 5
Breakpoint 2 at 0x11a4: file memo.c, line 5.
(gdb) dump binary memory memo.txt 0x1189 0x11a4
(gdb) run
Starting program: /home/lokesh/Desktop/g1
```



Strace Example*To use strace first compile your program as an executable file and then type "strace filename"*

Here: new is the executable so type "strace ./new" and also "strace -e read,write ./new" prints only system calls related to input and output.

```
lokesh@17mis1088:~$ strace tar
execve("/usr/bin/tar", ["tar"], 0x7ffe8b7ff970 /* 51 vars */) = 0
brk(NULL)                               = 0x562ea80d0000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffe9dd65e10) = -1 EINVAL (Invalid argument)
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=64974, ...}) = 0
mmap(NULL, 64974, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f0833524000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libacl.so.1", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220\"\0\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0644, st_size=38936, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f0833522000
mmap(NULL, 41120, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f0833517000
mmap(0x7f0833519000, 20480, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7f0833519000
mmap(0x7f083351e000, 8192, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x7000) = 0x7f083351e000
mmap(0x7f0833520000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x8000) = 0x7f0833520000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libselinux.so.1", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\300p\0\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0644, st_size=163240, ...}) = 0
mmap(NULL, 174640, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f08334ec000
mprotect(0x7f08334f2000, 135168, PROT_NONE) = 0
mmap(0x7f08334f2000, 102400, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x6000) = 0x7f08334f2000
mmap(0x7f083350b000, 28672, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1f000) = 0x7f083350b000
mmap(0x7f0833513000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x26000) = 0x7f0833513000
mmap(0x7f0833515000, 6704, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f0833515000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\360r\2\0\0\0\0\0"..., 832) = 832
lseek(3, 64, SEEK_SET)                  = 64
read(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784) = 784
lseek(3, 848, SEEK_SET)                 = 848
read(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32) = 32
lseek(3, 880, SEEK_SET)                 = 880
read(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0!U\364U\255V\275\207\34\202%\274\312\205\356%"..., 68) = 68
fstat(3, {st_mode=S_IFREG|0755, st_size=2025032, ...}) = 0
openat(AT_FDCWD, "/usr/share/locale/en_IN/LC_MESSAGES/tar.mo", O_RDONLY) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/usr/share/locale/en/LC_MESSAGES/tar.mo", O_RDONLY) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/usr/share/locale-langpack/en_IN/LC_MESSAGES/tar.mo", O_RDONLY) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/usr/share/locale-langpack/en/LC_MESSAGES/tar.mo", O_RDONLY) = -1 ENOENT (No such file or directory)
write(2, "tar: ", 5tar: )                   = 5
write(2, "You must specify one of the '-Ac"..., 76You must specify one of the '-Acdtrux', '--delete' or '--test-label' options) = 76
write(2, "\n", 1
)                       = 1
write(2, "Try 'tar --help' or 'tar --usage"..., 56Try 'tar --help' or 'tar --usage' for more information.
) = 56
close(1)                                = 0
close(2)                                = 0
exit_group(2)                           = ?
+++ exited with 2 +++
```

```
lokesh@17mis1088:~$ strace -e trace=read ./new
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\360r\2\0\0\0\0\0"..., 832) = 832
read(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784) = 784
read(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0", 32) = 32
read(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0!U\364U\255V\275\207\34\202%\274\312\205\356%"..., 68) = 68
read(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784) = 784
read(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0", 32) = 32
read(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0!U\364U\255V\275\207\34\202%\274\312\205\356%"..., 68) = 68
read(-1, 0x7fff1ac689e0, 8192)          = -1 EBADF (Bad file descriptor)
read(-1, 0x7fff1ac689e0, 8192)          = -1 EBADF (Bad file descriptor)
Read: Bad file descriptor
+++ exited with 0 +++
lokesh@17mis1088:~$ strace -e trace=write ./new
write(3, "Read: Bad file descriptor\n", 26Read: Bad file descriptor
) = 26
+++ exited with 0 +++
```

```
lokesh@17mis1088:~$ strace -e trace=read,write ./new
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\360r\2\0\0\0\0\0"..., 832) = 832
read(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784) = 784
read(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0", 32) = 32
read(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0!U\364U\255V\275\207\34\202%\274\312\205\356%"..., 68) = 68
read(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784) = 784
read(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0", 32) = 32
read(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0!U\364U\255V\275\207\34\202%\274\312\205\356%"..., 68) = 68
read(-1, 0x7ffcf567d0e0, 8192)          = -1 EBADF (Bad file descriptor)
read(-1, 0x7ffcf567d0e0, 8192)          = -1 EBADF (Bad file descriptor)
write(3, "Read: Bad file descriptor\n", 26Read: Bad file descriptor
) = 26
+++ exited with 0 +++
```

Memory Region:

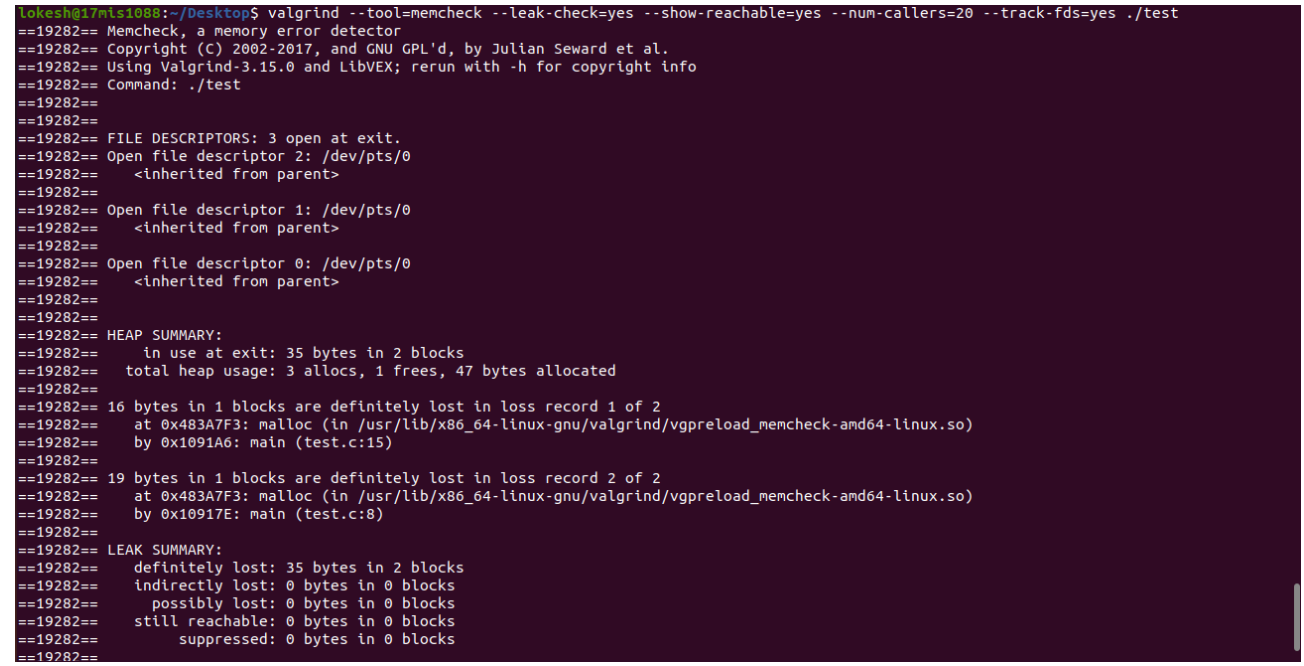
```
(gdb) b 1
Breakpoint 1 at 0x114d: file ex4.c, line 4.
(gdb) mem 0x114d 0x115e
(gdb) info mem
Using user-defined memory regions.
Num Enb Low Addr           High Addr          Attrs
1   y   0x000000000000114d 0x000000000000115e rw nocache
```

Valgrind:

```
#include<stdio.h>

int main()
{
  char *p;

  // Allocation #1 of 19 bytes
  p = (char *) malloc(19);

  // Allocation #2 of 12 bytes
  p = (char *) malloc(12);
  free(p);

  // Allocation #3 of 16 bytes
  p = (char *) malloc(16);

  return 0;
}
```

Output

```
lokesh@17mis1088:~/Desktop$ gcc -o test -g test.c
test.c: In function ‘main’:
test.c:8:16: warning: implicit declaration of function ‘malloc’ [-Wimplicit-function-declaration]
    8 |   p = (char *) malloc(19);
      |                ^~~~~~
test.c:8:16: warning: incompatible implicit declaration of built-in function ‘malloc’
test.c:2:1: note: include ‘<stdlib.h>’ or provide a declaration of ‘malloc’
    1 | #include <stdio.h>
  +++ |+#include <stdlib.h>
    2 |
test.c:12:3: warning: implicit declaration of function ‘free’ [-Wimplicit-function-declaration]
   12 |   free(p);
      |   ^~~~
test.c:12:3: warning: incompatible implicit declaration of built-in function ‘free’
test.c:12:3: note: include ‘<stdlib.h>’ or provide a declaration of ‘free’
```



Exercise:1

first.c :

```
#include <stdio.h>
void func1();
void func2();
int main() {
int i=10;
func1();
printf("In Main(): %d\n",i);
}
void func1() {
int n=20;
printf("In func1(): %d\n",n);
func2();
}
void func2() {
int n = 30;
printf("In func2() : %d\n",n);
}
```

```
lokesh@17mis1088:~/Desktop$ gcc -g -o first first.c
lokesh@17mis1088:~/Desktop$ gdb first
GNU gdb (Ubuntu 8.3-0ubuntu1) 8.3
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"...
Reading symbols from first...
(gdb) b func1
Breakpoint 1 at 0x1183: file first.c, line 9.
(gdb) b func2
Breakpoint 2 at 0x11b9: file first.c, line 14.
(gdb) run
Starting program: /home/lokesh/Desktop/first

Breakpoint 1, func1 () at first.c:9
9       void func1() {
(gdb) c
Continuing.
In func1(): 20

Breakpoint 2, func2 () at first.c:14
14      void func2() {
(gdb) bt
#0  func2 () at first.c:14
#1  0x00005555555551b6 in func1 () at first.c:12
#2  0x0000555555555166 in main () at first.c:6
```

```
(gdb) frame 1
#1  0x00005555555551b6 in func1 () at first.c:12
12      func2();
(gdb) frame 2
#2  0x0000555555555166 in main () at first.c:6
6       func1();
(gdb) p n
No symbol "n" in current context.
(gdb) q
A debugging session is active.

        Inferior 1 [process 21473] will be killed.

Quit anyway? (y or n) y
lokesh@17mis1088:~/Desktop$
```

Exercise:2

```
#include <stdio.h>
int sum(int n);
int main() {
int number, result;
printf("Enter a positive integer: ");
scanf("%d", &number);
result = sum(number);
printf("sum = %d", result);
return 0;
}
int sum(int n) {
if (n != 0)
// sum() function calls itself
return n + sum(n-1);
else
return n;
}
```

Output:

```
lokesh@17mis1088:~/Desktop$ gdb second
GNU gdb (Ubuntu 8.3-0ubuntu1) 8.3
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"...
Reading symbols from second...
(gdb) b sum
Breakpoint 1 at 0x120b: file second.c, line 11.
(gdb) run
Starting program: /home/lokesh/Desktop/second
Enter a positive integer: 2

Breakpoint 1, sum (n=0) at second.c:11
11      int sum(int n) {
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) n
Program not restarted.
(gdb) c
Continuing.
#0  sum (n=32767) at second.c:11
#1  0x000055555555522d in sum (n=2) at second.c:14
#2  0x00005555555551d7 in main () at second.c:7
(gdb) frame 1
#1  0x000055555555522d in sum (n=2) at second.c:14
14      return n + sum(n-1);
(gdb) p n
$1 = 2
(gdb) q
A debugging session is active.

        Inferior 1 [process 25258] will be killed.

Quit anyway? (y or n) y
lokesh@17mis1088:~/Desktop$
```

```
#include<stdio.h>
int main()
{
char *s =”Goal”;
char *t =”Home”;
while(*s++ = *t++)
printf(*s);
return 0;
}
```

Exercise:3

```
#include <stdio.h>
```

```
/* Print the sum of the integers from 1 to 1000 */
intmain(int argc, char **argv)
        {
                int i;
                int sum;    sum = 0;
                for(i = 0; i -= 1000; i++)
                        { sum += i; }
                printf("%d\n", sum);
                return 0;
        }
```

```
lokesh@lokesh:~/Desktop$ gcc -g -o one one.c
one.c:3:1: warning: return type defaults to ‘int’ [-Wimplicit-int]
 intmain(int argc, char **argv)
 ^~~~~~~
/usr/bin/ld: /usr/lib/gcc/x86_64-linux-gnu/8/../../../x86_64-linux-gnu/Scr
t1.o: in function `_start':
(.text+0x20): undefined reference to `main'
collect2: error: ld returned 1 exit status
```

There is an error in compilation so we need to correct the code the corrected code will have **int main** function correctly organised.
After organizing we have

```
lokesh@lokesh:~/Desktop$ gcc -g -o one one.c
lokesh@lokesh:~/Desktop$ gdb one
GNU gdb (Ubuntu 8.2.91.20190405-0ubuntu3) 8.2.91.20190405-git
Copyright (C) 2019 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.ht
ml>
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 one...
(gdb) b 6
Breakpoint 1 at 0x1144: file one.c, line 6.
(gdb) b 8
Breakpoint 2 at 0x1154: file one.c, line 8.
(gdb) run
Starting program: /home/lokesh/Desktop/one
```

```
$8 = -2999
(gdb) q
A debugging session is active.

        Inferior 1 [process 5464] will be killed.

Quit anyway? (y or n) y
lokesh@lokesh:~/Desktop$
```

since we are getting the answer there is a logical error in the code that is

for(i = 0; i -= 1000; i++)
{ sum += i; }
Corrected:
for(i = 0; i <= 1000; i++)
{ sum += i; }

```
(gdb) run
Starting program: /home/lokesh/Desktop/one

Breakpoint 1, main (argc=1, argv=0x7fffffffe0c8) at one.c:6
6                       int sum;    sum = 0;
(gdb) p i
$1 = 0
(gdb) p sum
$2 = 0
(gdb) c
Continuing.

Breakpoint 2, main (argc=1, argv=0x7fffffffe0c8) at one.c:8
8                               { sum += i; }
(gdb) p i
$3 = -1000
(gdb) p sum
$4 = 0
(gdb) c
Continuing.

Breakpoint 2, main (argc=1, argv=0x7fffffffe0c8) at one.c:8
8                               { sum += i; }
(gdb) p i
$5 = -1999
(gdb) p sum
$6 = -1000
(gdb) c
Continuing.

Breakpoint 2, main (argc=1, argv=0x7fffffffe0c8) at one.c:8
8                               { sum += i; }
(gdb) p i
$7 = -2998
(gdb) p sum
```

```
lokesh@lokesh:~/Desktop$ gcc -g -o one one.c
lokesh@lokesh:~/Desktop$ gdb one
GNU gdb (Ubuntu 8.2.91.20190405-0ubuntu3) 8.2.91.20190405-git
Copyright (C) 2019 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.ht
ml>
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 one...
(gdb) run
Starting program: /home/lokesh/Desktop/one
500500
[Inferior 1 (process 7105) exited normally]
(gdb) q
lokesh@lokesh:~/Desktop$
```

Exercise:4

```
#include<stdio.h>


int main()
        {
                char *s = “Goal”;
                char *t = “Home”;
                while(*s++ = *t++)
                        printf(*s);
                return 0;
        }
```

```
lokesh@lokesh:~/Desktop$ gcc -g -o two two.c
two.c: In function ‘main’:
two.c:5:13: error: stray ‘\342’ in program
   char *s = ◆◆◆Goal”;
             ^
two.c:5:14: error: stray ‘\200’ in program
   char *s = ◆◆◆Goal”;
              ^
two.c:5:15: error: stray ‘\234’ in program
   char *s = ◆◆Goal”;
               ^
two.c:5:20: error: stray ‘\342’ in program
   char *s = “Goal◆◆◆;
                    ^
two.c:5:21: error: stray ‘\200’ in program
   char *s = “Goal◆◆◆;
                     ^
two.c:5:22: error: stray ‘\235’ in program
   char *s = “Goal◆◆;
                      ^
two.c:5:16: error: ‘Goal’ undeclared (first use in this function)
   char *s = “Goal”;
                ^~~~
```

We are getting the errors because of the **“”** quatation marks are used to compile the program we need to use **""**using that

```
#include<stdio.h>

int main()
	{
		char *s = "Goal";
		char *t = "Home";
		while(*s++ = *t++)
			printf(*s);
		return 0;
	}
```

now we compile the code

```
lokesh@lokesh:~/Desktop$ gcc -g -o two two.c
two.c: In function ‘main’:
two.c:8:11: warning: passing argument 1 of ‘printf’ makes pointer from int
eger without a cast [-Wint-conversion]
    printf(*s);
           ^~
In file included from two.c:1:
/usr/include/stdio.h:332:43: note: expected ‘const char * restrict’ but ar
gument is of type ‘char’
 extern int printf (const char *__restrict __format, ...);
                    ~~~~~~~~~~~~~~~~~~~~~~~^~~~~~~~
two.c:8:4: warning: format not a string literal and no format arguments [-
Wformat-security]
    printf(*s);
    ^~~~~~
lokesh@lokesh:~/Desktop$ gdb two
GNU gdb (Ubuntu 8.2.91.20190405-0ubuntu3) 8.2.91.20190405-git
Copyright (C) 2019 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.ht
ml>
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 two...
(gdb) run
Starting program: /home/lokesh/Desktop/two

Program received signal SIGSEGV, Segmentation fault.
main () at two.c:7
7                       while(*s++ = *t++)
```

Gdb shows the error in 7th line i.e, in the while loop. So let's understabd the while loop first. The while loop is matching the two string literals(*s* and *t*) and changing *s* to *t*. This modification can't be done without causing a segmentation fault as they both are declared as string literals. So we need to change first string to char array. This way, the elements of the first string can be modified without causing a segmentation fault.

char arr[] = "Goal";

char *s = arr;

Now, string *s* can be morphed into string *t* via the while loop without faults. The loop doesn't need a body, and just to check if the transformation has been done properly, we print the array. So, our modified code should look like :

```
#include<stdio.h>

int main()
	{
		char arr[] = "Goal";
		char *s = arr;
		char *t = "Home";
		while(*s++ = *t++);
		printf("%s", arr);
		return 0;
	}
```

```
lokesh@lokesh:~/Desktop$ gcc -g -o two two.c
lokesh@lokesh:~/Desktop$ gdb two
GNU gdb (Ubuntu 8.2.91.20190405-0ubuntu3) 8.2.91.20190405-git
Copyright (C) 2019 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.ht
ml>
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 two...
(gdb) run
Starting program: /home/lokesh/Desktop/two
Hoal
Hoal
Homl
Home
[Inferior 1 (process 4073) exited normally]
```