

```
0
1
2
3
4
5
6
7
10
1437812336
akhilesh@akhilesh-VirtualBox:~/Desktop/422104/week7$ gcc -g sum.c -o sum
akhilesh@akhilesh-VirtualBox:~/Desktop/422104/week7$ ./sum
0
1
2
3
4
5
6
7
10
1437812336
akhilesh@akhilesh-VirtualBox:~/Desktop/422104/week7$ gdb sum
GNU gdb (Ubuntu 12.1-0ubuntu1~22.04) 12.1
Copyright (C) 2022 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:
<https://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 sum...
(gdb) run
Starting program: /home/akhilesh/Desktop/422104/week7/sum
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
0
1
2
3
4
5
6
7
10
1431655769
[Inferior 1 (process 4508) exited normally]
(gdb) list
1      #include <stdio.h>
2      #include <stdlib.h>
3      int main() {
4
```

```

3   int main() {
4
5       int n = 8;
6       for(int i=0; i<n; i++){
7           printf("%d\n", i);
8       }
9
10      int *ptr = malloc(sizeof(int));
(gdb)
11
12      if (ptr != NULL) {
13          *ptr = 10;
14          printf("%d\n", *ptr);
15          free(ptr);
16      } else {
17          printf("Failed to allocate memory.\n");
18      }
19
20      printf("%d\n", *ptr);
(gdb)
21
22
23
24      return 0;
25  }
26
27

```

(gdb)

Line number 28 out of range; sum.c has 27 lines.

(gdb) break 5

Breakpoint 1 at 0x5555555551b5: file sum.c, line 5.

(gdb) break 12

Breakpoint 2 at 0x5555555551f8: file sum.c, line 12.

(gdb) run

Starting program: /home/akhilesh/Desktop/422104/week7/sum

[Thread debugging using libthread_db enabled]

Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, main () at sum.c:5

```

5       int n = 8;

```

(gdb) next

```

6       for(int i=0; i<n; i++){

```

(gdb) next

```

7           printf("%d\n", i);

```

(gdb) print n

\$1 = 8

(gdb) next

0

```

6       for(int i=0; i<n; i++){

```

(gdb) next

```

7           printf("%d\n", i);

```

(gdb) continue

Continuing.

1

Continuing.

1
2
3
4
5
6
7

Breakpoint 2, main () at sum.c:12

12 if (ptr != NULL) {

(gdb) next

13 *ptr = 10;

(gdb) continue

Continuing.

10

1431655769

[Inferior 1 (process 4511) exited normally]

(gdb) disassemble

No frame selected.

(gdb) run

Starting program: /home/akhilesh/Desktop/422104/week7/sum

[Thread debugging using libthread_db enabled]

Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, main () at sum.c:5

5 int n = 8;

(gdb) disassemble

Dump of assembler code for function main:

```
0x0000555555551a9 <+0>:      endbr64
0x0000555555551ad <+4>:      push    %rbp
0x0000555555551ae <+5>:      mov     %rsp,%rbp
0x0000555555551b1 <+8>:      sub     $0x10,%rsp
=> 0x0000555555551b5 <+12>:     movl    $0x8,-0xc(%rbp)
0x0000555555551bc <+19>:     movl    $0x0,-0x10(%rbp)
0x0000555555551c3 <+26>:     jmp     0x555555551e2 <main+57>
0x0000555555551c5 <+28>:     mov     -0x10(%rbp),%eax
0x0000555555551c8 <+31>:     mov     %eax,%esi
0x0000555555551ca <+33>:     lea     0xe33(%rip),%rax          # 0x555555556004
0x0000555555551d1 <+40>:     mov     %rax,%rdi
0x0000555555551d4 <+43>:     mov     $0x0,%eax
0x0000555555551d9 <+48>:     call    0x555555550a0 <printf@plt>
0x0000555555551de <+53>:     addl    $0x1,-0x10(%rbp)
0x0000555555551e2 <+57>:     mov     -0x10(%rbp),%eax
0x0000555555551e5 <+60>:     cmp     -0xc(%rbp),%eax
0x0000555555551e8 <+63>:     jl      0x555555551c5 <main+28>
0x0000555555551ea <+65>:     mov     $0x4,%edi
0x0000555555551ef <+70>:     call    0x555555550b0 <malloc@plt>
0x0000555555551f4 <+75>:     mov     %rax,-0x8(%rbp)
0x0000555555551f8 <+79>:     cmpq    $0x0,-0x8(%rbp)
0x0000555555551fd <+84>:     je      0x55555555233 <main+138>
0x0000555555551ff <+86>:     mov     -0x8(%rbp),%rax
0x000055555555203 <+90>:     movl    $0xa,(%rax)
0x000055555555209 <+96>:     mov     -0x8(%rbp),%rax
0x00005555555520d <+100>:    mov     (%rax),%eax
0x00005555555520f <+102>:    mov     %eax,%esi
0x000055555555211 <+104>:    lea     0xdce(%rip),%rax          # 0x555555556004
0x000055555555218 <+111>:    mov     %rax,%rdi
0x00005555555521b <+114>:    mov     $0x0,%eax
0x000055555555220 <+119>:    call    0x555555550a0 <printf@plt>
0x000055555555225 <+124>:    mov     -0x8(%rbp),%rax
0x000055555555229 <+128>:    mov     %rax,%rdi
0x00005555555522c <+131>:    call    0x55555555080 <free@plt>
0x000055555555231 <+136>:    jmp     0x55555555242 <main+153>
0x000055555555233 <+138>:    lea     0xdce(%rip),%rax          # 0x555555556008
0x00005555555523a <+145>:    mov     %rax,%rdi
0x00005555555523d <+148>:    call    0x55555555090 <puts@plt>
0x000055555555242 <+153>:    mov     -0x8(%rbp),%rax
0x000055555555246 <+157>:    mov     (%rax),%eax
0x000055555555248 <+159>:    mov     %eax,%esi
0x00005555555524a <+161>:    lea     0xdb3(%rip),%rax          # 0x555555556004
0x000055555555251 <+168>:    mov     %rax,%rdi
0x000055555555254 <+171>:    mov     $0x0,%eax
0x000055555555259 <+176>:    call    0x555555550a0 <printf@plt>
0x00005555555525e <+181>:    mov     $0x0,%eax
0x000055555555263 <+186>:    leave
0x000055555555264 <+187>:    ret
```

End of assembler dump.