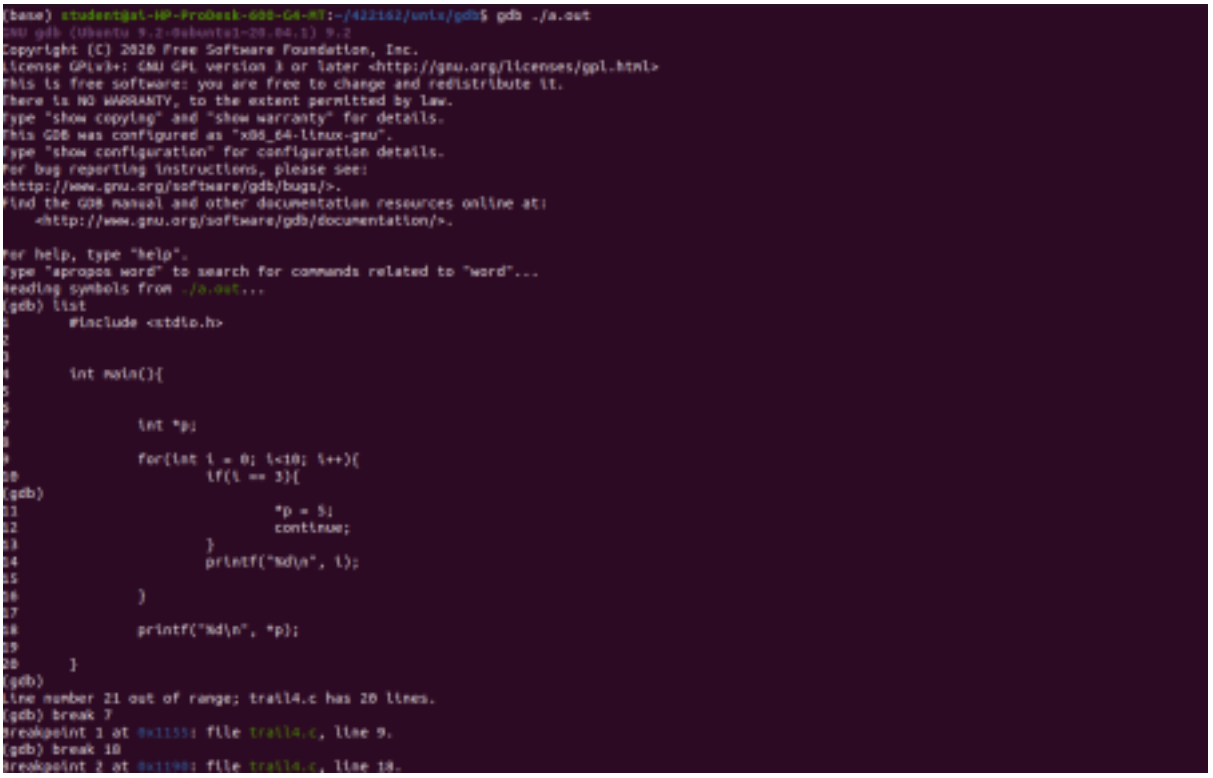
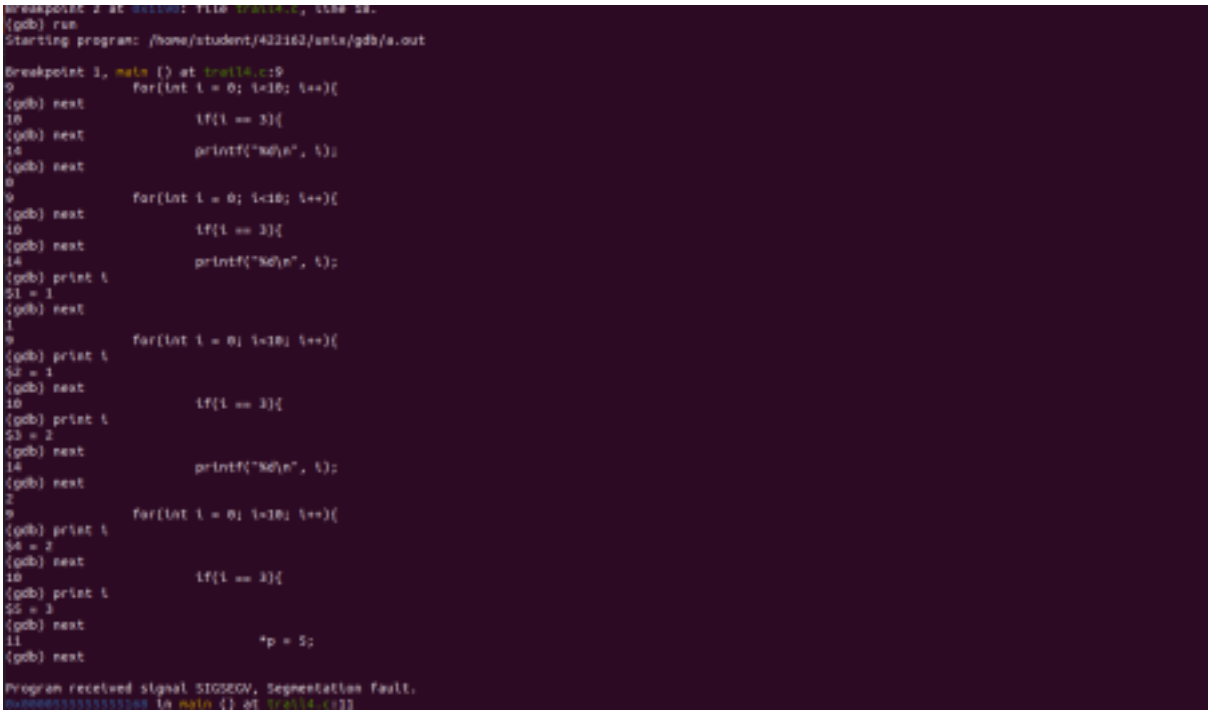
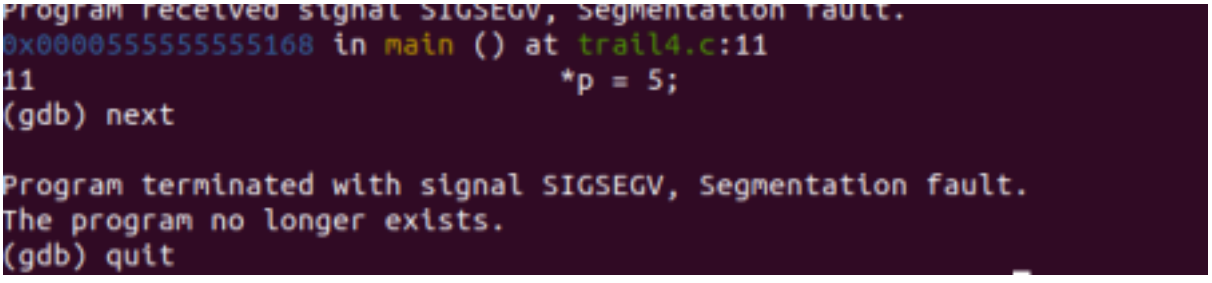
seg fault in for loop:







Seg fault with illegal pointer:

```
(base) student@pt-HP-ProDesk-600-G4-MT:~/422162/unix/gdb$ gdb ./a.out
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04.1) 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) list
1       #include<stdio.h>
2
3
4
5       double* getptr(){
6               double x = 10;
7               return &x;
8       }
9
10      int main(){
(gdb)
11
12              int i =10;
13              int j = 16;
14
15              double* k  = getptr();
16
17              for(int x = 0; x<5;x++){
18                      i += x;
19              }
20
(gdb)
```

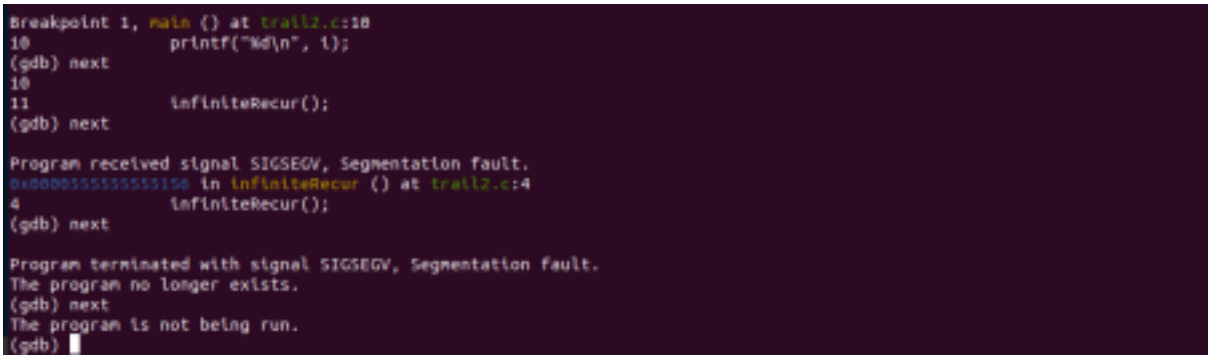
```
21              double z = (i+j)/2;
22              printf("%f\n", z);
23              z = *k;
24
25
26      }
(gdb)
Line number 27 out of range; trall3.c has 26 lines.
(gdb) break 15
Breakpoint 1 at 0x11c0: file trall3.c, line 15.
(gdb) break 24
Breakpoint 2 at 0x1233: file trall3.c, line 26.
(gdb) run
Starting program: /home/student/422162/unix/gdb/a.out

Breakpoint 1, main () at trall3.c:15
15              double* k  = getptr();
(gdb) next
17              for(int x = 0; x<5;x++){
(gdb) next
18                      i += x;
(gdb) next
17              for(int x = 0; x<5;x++){
(gdb) next
18                      i += x;
(gdb) next
17              for(int x = 0; x<5;x++){
(gdb) next
18                      i += x;
(gdb) next
17              for(int x = 0; x<5;x++){
(gdb) print i
$1 = 13
(gdb) next
18                      i += x;
(gdb) print j
$2 = 16
(gdb) next
17              for(int x = 0; x<5;x++){
(gdb) next
```

```
(gdb) next
21              double z = (i+j)/2;
(gdb) print i
$3 = 20
(gdb) print z
$4 = 0
(gdb) print j
$5 = 16
(gdb) next
22              printf("%f\n", z);
(gdb) next
18.000000
23              z = *k;
(gdb) next

Program received signal SIGSEGV, Segmentation fault.
0x0000555555555225 in main () at trall3.c:23
23              z = *k;
(gdb) next
```

```
Program terminated with signal SIGSEGV, Segmentation fault.
The program no longer exists.
(gdb) disassemble main
Dump of assembler code for function main:
   0x00005555555551ac <+0>:     endbr64
   0x00005555555551b0 <+4>:     push   %rbp
   0x00005555555551b1 <+5>:     mov    %rsp,%rbp
   0x00005555555551b4 <+8>:     sub    $0x20,%rsp
   0x00005555555551b8 <+12>:    movl   $0xa,-0x1c(%rbp)
   0x00005555555551bf <+19>:    movl   $0x10,-0x14(%rbp)
   0x00005555555551c6 <+26>:    mov    $0x0,%eax
   0x00005555555551cb <+31>:    callq  0x5555555555169 <getptr>
   0x00005555555551d0 <+36>:    mov    %rax,-0x10(%rbp)
   0x00005555555551d4 <+40>:    movl   $0x8,-0x18(%rbp)
   0x00005555555551db <+47>:    jmp    0x55555555551e7 <main+59>
   0x00005555555551dd <+49>:    mov    -0x18(%rbp),%eax
   0x00005555555551e0 <+52>:    add    %eax,-0x1c(%rbp)
   0x00005555555551e3 <+55>:    addl   $0x1,-0x18(%rbp)
   0x00005555555551e7 <+59>:    cmpl   $0x4,-0x18(%rbp)
   0x00005555555551eb <+63>:    jle    0x55555555551dd <main+49>
   0x00005555555551ed <+65>:    mov    -0x1c(%rbp),%edx
   0x00005555555551f0 <+68>:    mov    -0x14(%rbp),%eax
   0x00005555555551f3 <+71>:    add    %edx,%eax
   0x00005555555551f5 <+73>:    mov    %eax,%edx
   0x00005555555551f7 <+75>:    shr    $0x1f,%edx
   0x00005555555551fa <+78>:    add    %edx,%eax
   0x00005555555551fc <+80>:    sar    %eax
   0x00005555555551fe <+82>:    cvtsi2sd %eax,%xmm0
   0x0000555555555202 <+86>:    movsd  %xmm0,-0x8(%rbp)
   0x0000555555555207 <+91>:    mov    -0x8(%rbp),%rax
   0x000055555555520b <+95>:    movq   %rax,%xmm0
   0x0000555555555210 <+100>:   lea    0xdf1(%rip),%rdi        # 0x5555555560008
   0x0000555555555217 <+107>:   mov    $0x1,%eax
   0x000055555555521c <+112>:   callq  0x5555555555070 <printf@plt>
   0x0000555555555221 <+117>:   mov    -0x10(%rbp),%rax
   0x0000555555555225 <+121>:   movsd  (%rax),%xmm0
   0x0000555555555229 <+125>:   movsd  %xmm0,-0x8(%rbp)
   0x000055555555522e <+130>:   mov    $0x0,%eax
   0x0000555555555233 <+135>:   leaveq
   0x0000555555555234 <+136>:   retq
End of assembler dump.
(gdb)
```

Seg fault with infinite recursion:

```
(base) student@pi-HP-ProDesk-600-G4-MT:~/422162/unix/gdb$ gcc -g trail2.c
(base) student@pi-HP-ProDesk-600-G4-MT:~/422162/unix/gdb$ gdb ./a.out
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04.1) 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) list
1       #include<stdio.h>
2
3       void infiniteRecur(){
4               infiniteRecur();
5       }
6
7       int main(){
8
9               int i = 10;
10              printf("%d\n", i);
(gdb)
11              infiniteRecur();
12
13              return 0;
14
15
16      }
(gdb)
Line number 17 out of range; trail2.c has 16 lines.
(gdb) break 10
Breakpoint 1 at 0x1171: file trail2.c, line 10.
(gdb) break 13
Breakpoint 2 at 0x1191: file trail2.c, line 13.
(gdb) run
Starting program: /home/student/422162/unix/gdb/a.out
```


Breakpoint 1, main () at trail2.c:10
10 printf("%d\n", i);
(gdb) next
10
11 infiniteRecur();
(gdb) next
Program received signal SIGSEGV, Segmentation fault.
0x0000555555555156 in infiniteRecur () at trail2.c:4
4 infiniteRecur();
(gdb) next
Program terminated with signal SIGSEGV, Segmentation fault.
The program no longer exists.
(gdb) next
The program is not being run.
(gdb)