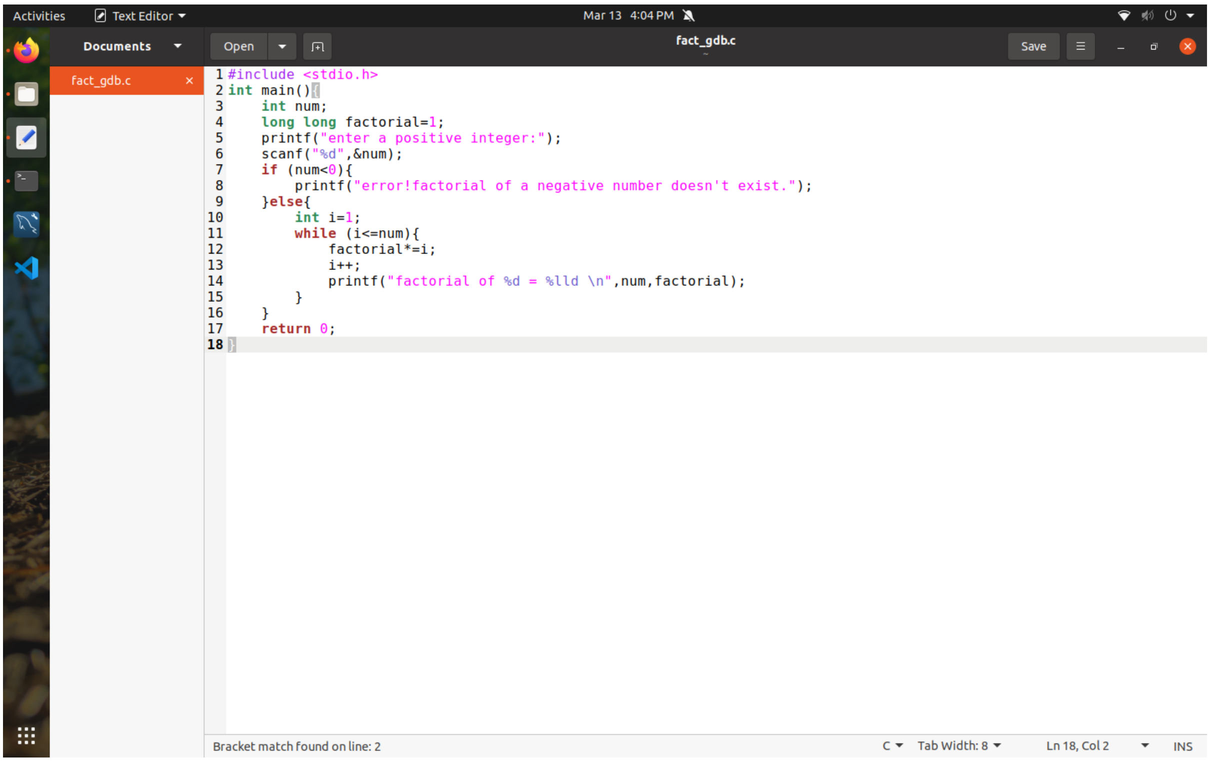
Name: B.Sumanth

Roll No: 422120

Sec: ‘A’

Assignment: Week 4

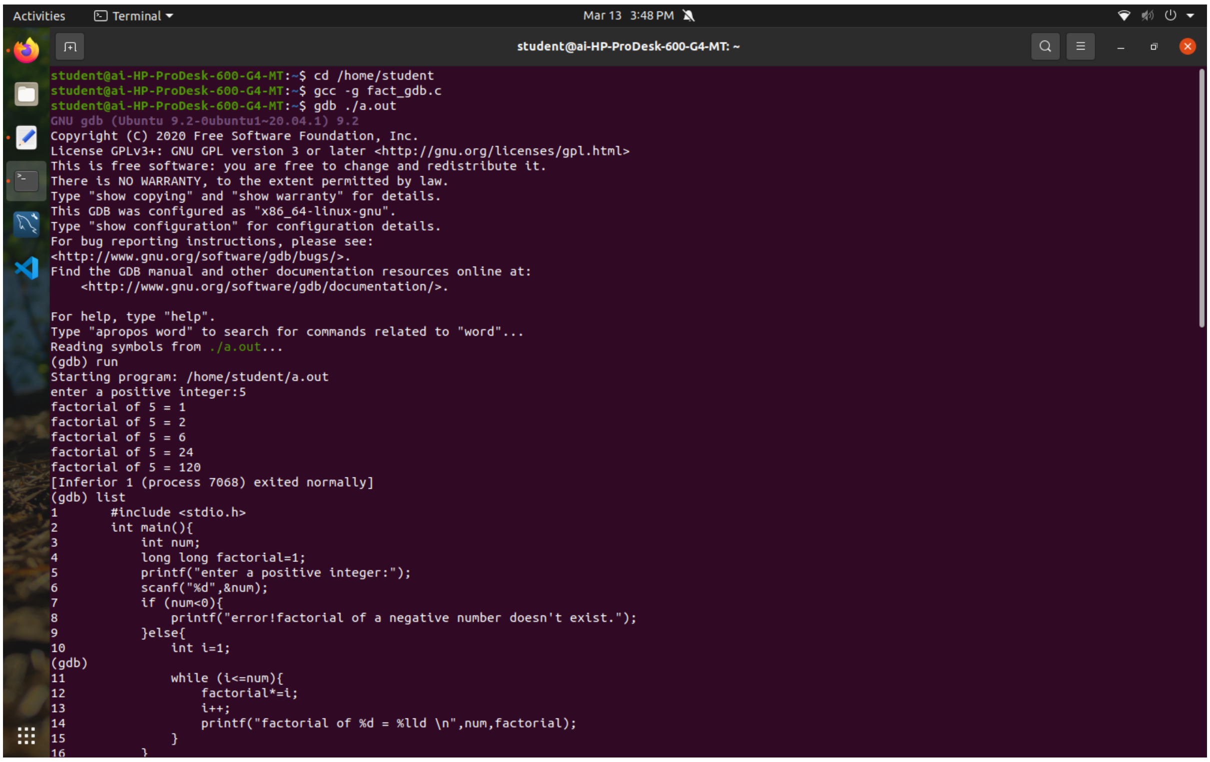
1.



```c
#include <stdio.h>
int main(){
    int num;
    long long factorial=1;
    printf("enter a positive integer:");
    scanf("%d",&num);
    if (num<0){
        printf("error!factorial of a negative number doesn't exist.");
    }else{
        int i=1;
        while (i<=num){
            factorial*=i;
            i++;
            printf("factorial of %d = %lld \n",num,factorial);
        }
    }
    return 0;
}
```

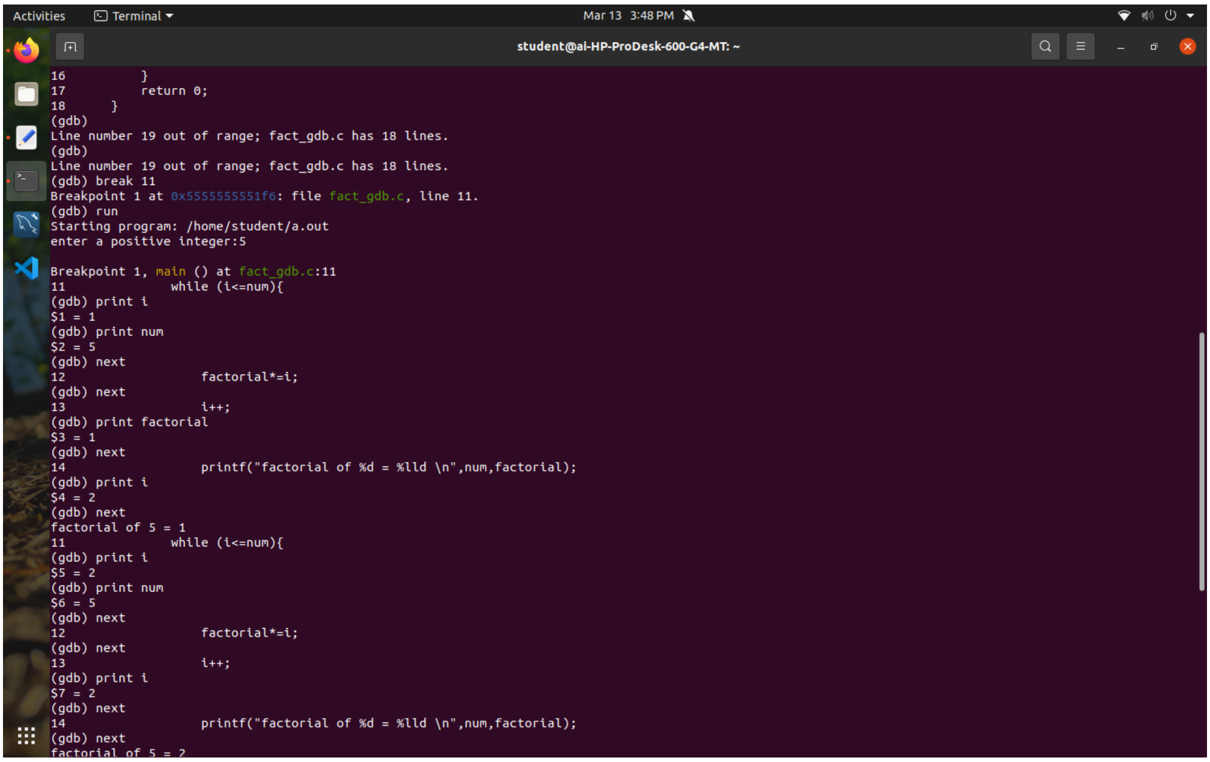
Bracket match found on line: 2

OUTPUT:



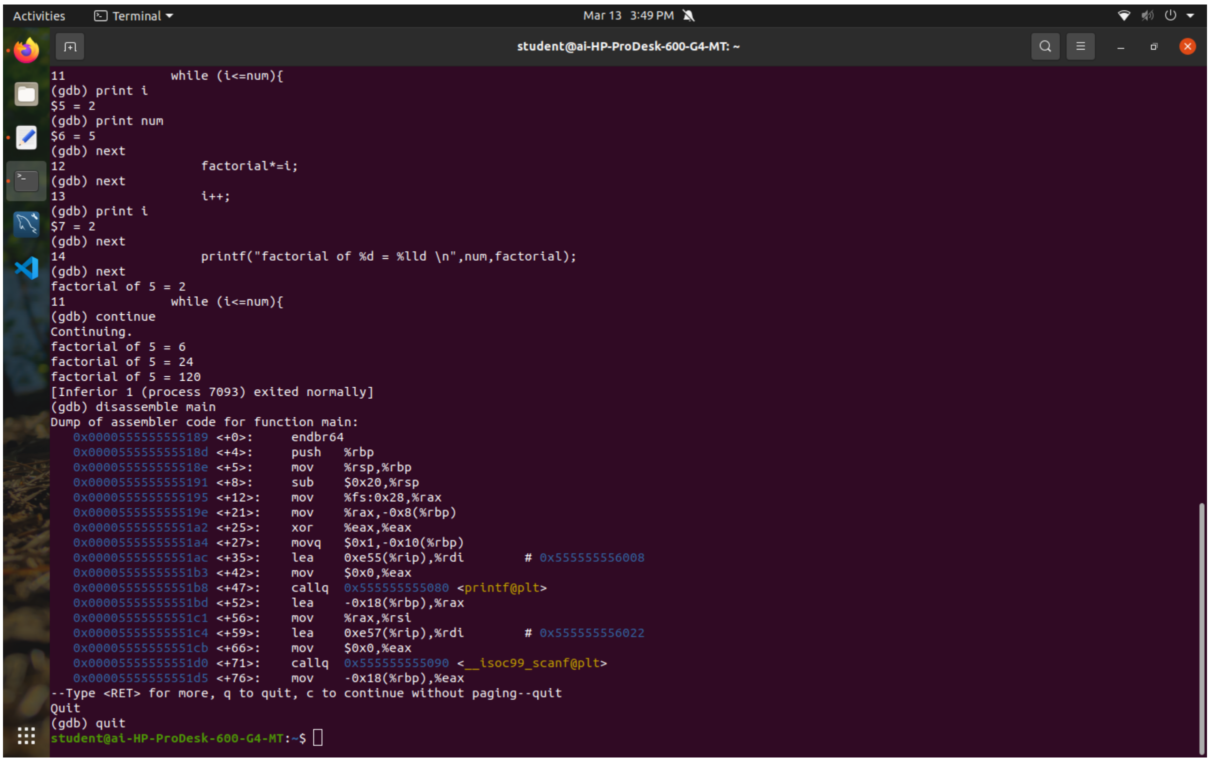
```
student@ai-HP-ProDesk-600-G4-MT:~$ cd /home/student
student@ai-HP-ProDesk-600-G4-MT:~$ gcc -g fact_gdb.c
student@ai-HP-ProDesk-600-G4-MT:~$ gdb ./a.out
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04.1) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
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) run
Starting program: /home/student/a.out
enter a positive integer:5
factorial of 5 = 1
factorial of 5 = 2
factorial of 5 = 6
factorial of 5 = 24
factorial of 5 = 120
[Inferior 1 (process 7068) exited normally]
(gdb) list
1       #include <stdio.h>
2       int main(){
3           int num;
4           long long factorial=1;
5           printf("enter a positive integer:");
6           scanf("%d",&num);
7           if (num<0){
8               printf("error!factorial of a negative number doesn't exist.");
9           }else{
10              int i=1;
(gdb)
11              while (i<=num){
12                  factorial*=i;
13                  i++;
14                  printf("factorial of %d = %lld \n",num,factorial);
15              }
16          }
```



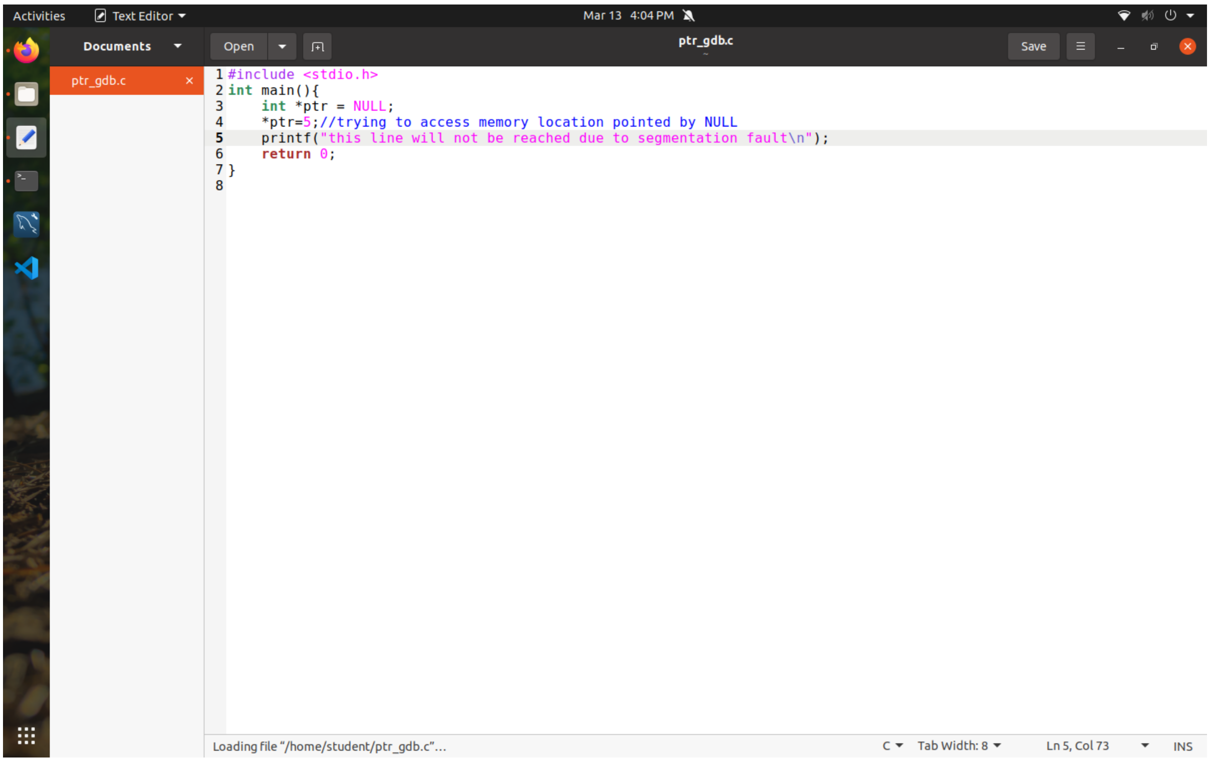
```
16          }
17          return 0;
18      }
(gdb)
Line number 19 out of range; fact_gdb.c has 18 lines.
(gdb)
Line number 19 out of range; fact_gdb.c has 18 lines.
(gdb) break 11
Breakpoint 1 at 0x5555555551f6: file fact_gdb.c, line 11.
(gdb) run
Starting program: /home/student/a.out
enter a positive integer:5

Breakpoint 1, main () at fact_gdb.c:11
11              while (i<=num){
(gdb) print i
$1 = 1
(gdb) print num
$2 = 5
(gdb) next
12                  factorial*=i;
(gdb) next
13                  i++;
(gdb) print factorial
$3 = 1
(gdb) next
14                  printf("factorial of %d = %lld \n",num,factorial);
(gdb) print i
$4 = 2
(gdb) next
factorial of 5 = 1
11              while (i<=num){
(gdb) print i
$5 = 2
(gdb) print num
$6 = 5
(gdb) next
12                  factorial*=i;
(gdb) next
13                  i++;
(gdb) print i
$7 = 2
(gdb) next
14                  printf("factorial of %d = %lld \n",num,factorial);
(gdb) next
factorial of 5 = 2
```



```
11              while (i<=num){
(gdb) print i
$5 = 2
(gdb) print num
$6 = 5
(gdb) next
12                  factorial*=i;
(gdb) next
13                  i++;
(gdb) print i
$7 = 2
(gdb) next
14                  printf("factorial of %d = %lld \n",num,factorial);
(gdb) next
factorial of 5 = 2
11              while (i<=num){
(gdb) continue
Continuing.
factorial of 5 = 6
factorial of 5 = 24
factorial of 5 = 120
[Inferior 1 (process 7093) exited normally]
(gdb) disassemble main
Dump of assembler code for function main:
   0x0000555555555189 <+0>:     endbr64
   0x000055555555518d <+4>:     push   %rbp
   0x000055555555518e <+5>:     mov    %rsp,%rbp
   0x0000555555555191 <+8>:     sub    $0x20,%rsp
   0x0000555555555195 <+12>:    mov    %fs:0x28,%rax
   0x000055555555519e <+21>:    mov    %rax,-0x8(%rbp)
   0x00005555555551a2 <+25>:    xor    %eax,%eax
   0x00005555555551a4 <+27>:    movq   $0x1,-0x10(%rbp)
   0x00005555555551ac <+35>:    lea    0xe55(%rip),%rdi        # 0x555555556008
   0x00005555555551b3 <+42>:    mov    $0x0,%eax
   0x00005555555551b8 <+47>:    callq  0x555555555080 <printf@plt>
   0x00005555555551bd <+52>:    lea    -0x18(%rbp),%rax
   0x00005555555551c1 <+56>:    mov    %rax,%rsi
   0x00005555555551c4 <+59>:    lea    0xe57(%rip),%rdi        # 0x555555556022
   0x00005555555551cb <+66>:    mov    $0x0,%eax
   0x00005555555551d0 <+71>:    callq  0x555555555090 <__isoc99_scanf@plt>
   0x00005555555551d5 <+76>:    mov    -0x18(%rbp),%eax
--Type <RET> for more, q to quit, c to continue without paging--quit
Quit
(gdb) quit
student@ai-HP-ProDesk-600-G4-MT:~$ 
```

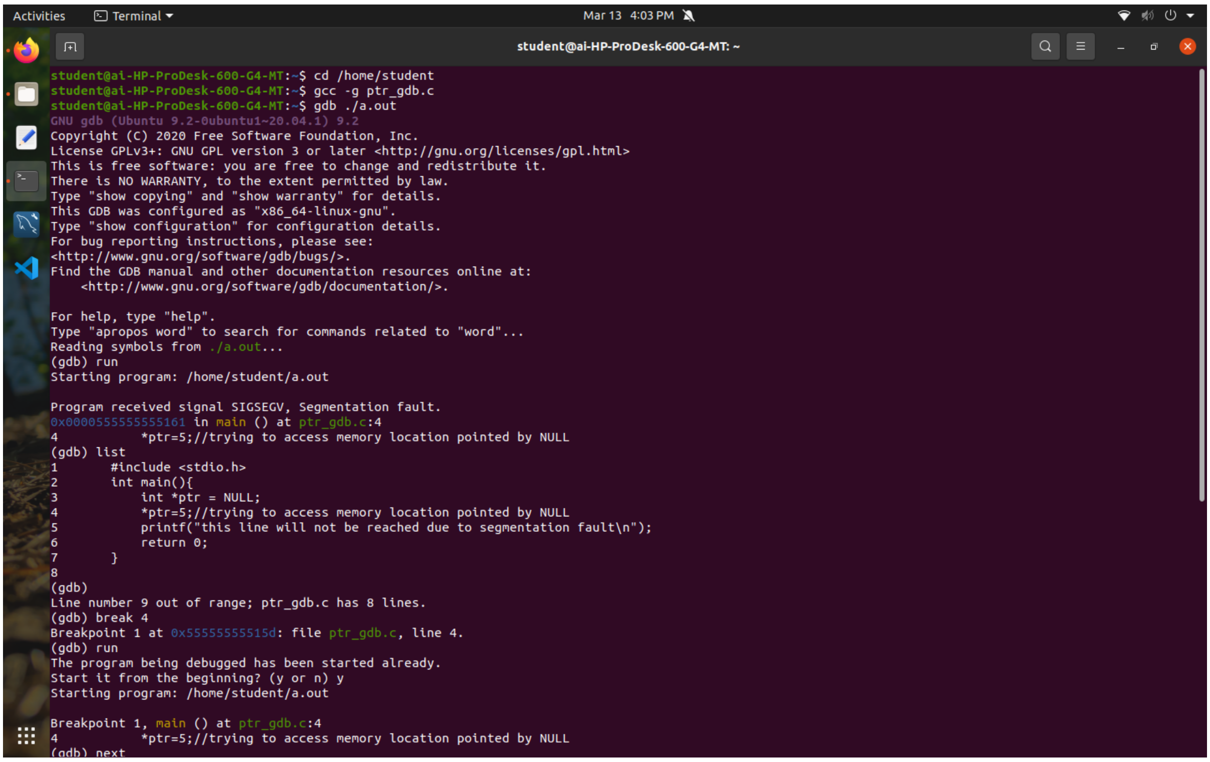
2.



```c
#include <stdio.h>
int main(){
    int *ptr = NULL;
    *ptr=5;//trying to access memory location pointed by NULL
    printf("this line will not be reached due to segmentation fault\n");
    return 0;
}

```

OUTPUT:

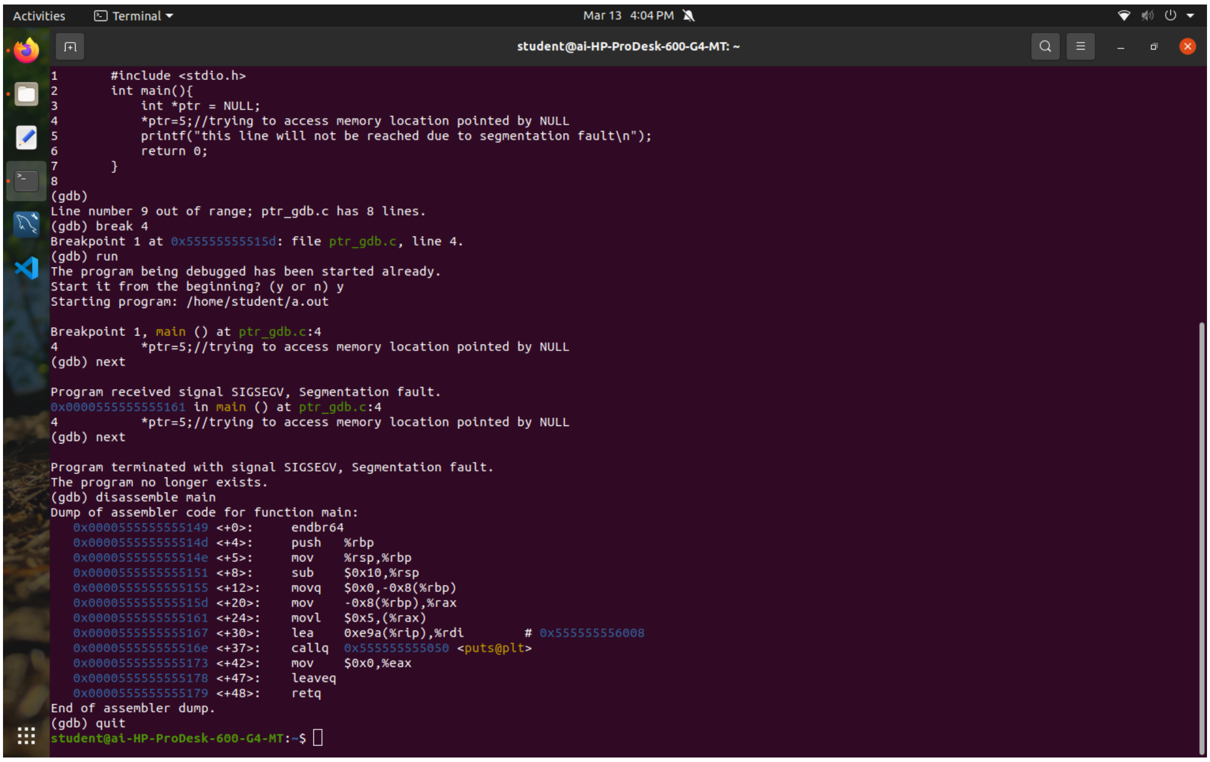


```
student@ai-HP-ProDesk-600-G4-MT:~$ cd /home/student
student@ai-HP-ProDesk-600-G4-MT:~$ gcc -g ptr_gdb.c
student@ai-HP-ProDesk-600-G4-MT:~$ gdb ./a.out
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04.1) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
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) run
Starting program: /home/student/a.out

Program received signal SIGSEGV, Segmentation fault.
0x0000555555555161 in main () at ptr_gdb.c:4
4           *ptr=5;//trying to access memory location pointed by NULL
(gdb) list
1       #include <stdio.h>
2       int main(){
3           int *ptr = NULL;
4           *ptr=5;//trying to access memory location pointed by NULL
5           printf("this line will not be reached due to segmentation fault\n");
6           return 0;
7       }
8
(gdb)
Line number 9 out of range; ptr_gdb.c has 8 lines.
(gdb) break 4
Breakpoint 1 at 0x55555555515d: file ptr_gdb.c, line 4.
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/student/a.out

Breakpoint 1, main () at ptr_gdb.c:4
4           *ptr=5;//trying to access memory location pointed by NULL
(gdb) next
```



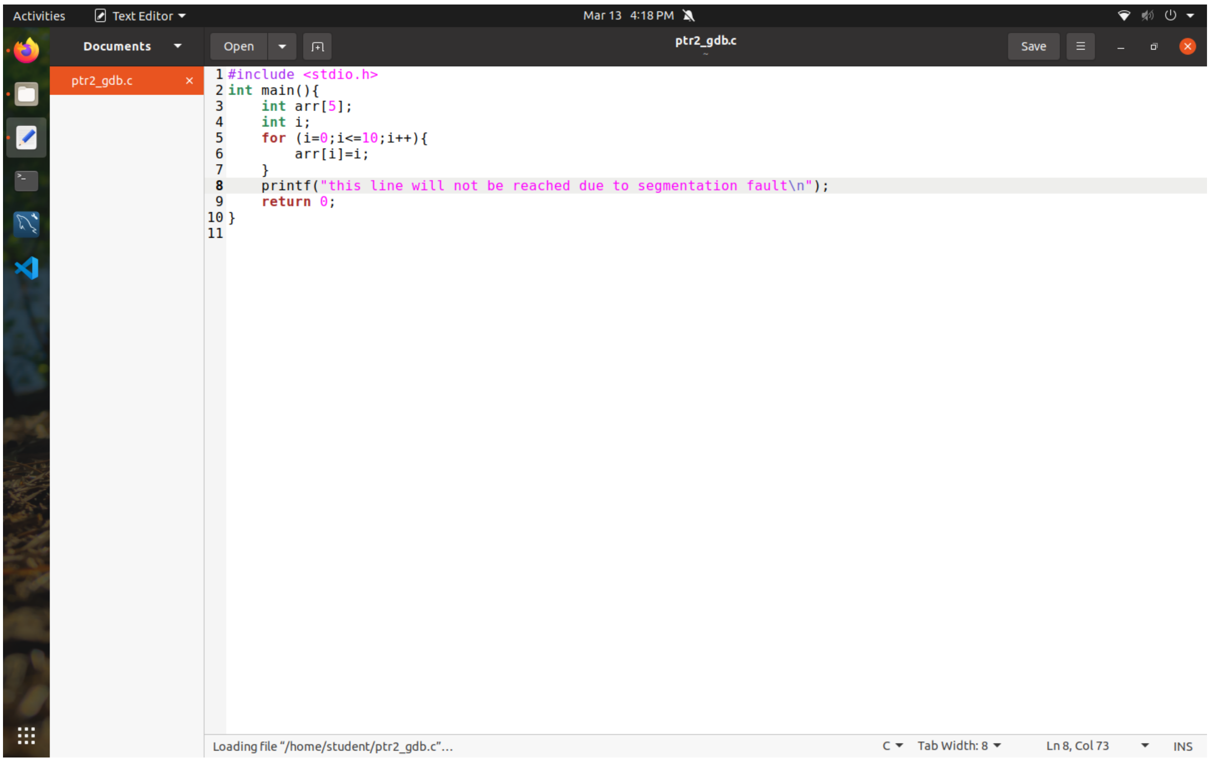
```
1       #include <stdio.h>
2       int main(){
3           int *ptr = NULL;
4           *ptr=5;//trying to access memory location pointed by NULL
5           printf("this line will not be reached due to segmentation fault\n");
6           return 0;
7       }
8
(gdb)
Line number 9 out of range; ptr_gdb.c has 8 lines.
(gdb) break 4
Breakpoint 1 at 0x55555555515d: file ptr_gdb.c, line 4.
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/student/a.out

Breakpoint 1, main () at ptr_gdb.c:4
4           *ptr=5;//trying to access memory location pointed by NULL
(gdb) next

Program received signal SIGSEGV, Segmentation fault.
0x0000555555555161 in main () at ptr_gdb.c:4
4           *ptr=5;//trying to access memory location pointed by NULL
(gdb) next

Program terminated with signal SIGSEGV, Segmentation fault.
The program no longer exists.
(gdb) disassemble main
Dump of assembler code for function main:
   0x0000555555555149 <+0>:     endbr64
   0x000055555555514d <+4>:     push   %rbp
   0x000055555555514e <+5>:     mov    %rsp,%rbp
   0x0000555555555151 <+8>:     sub    $0x10,%rsp
   0x0000555555555155 <+12>:    movq   $0x0,-0x8(%rbp)
   0x000055555555515d <+20>:    mov    -0x8(%rbp),%rax
   0x0000555555555161 <+24>:    movl   $0x5,(%rax)
   0x0000555555555167 <+30>:    lea    0xe9a(%rip),%rdi        # 0x555555556008
   0x000055555555516e <+37>:    callq  0x555555555050 <puts@plt>
   0x0000555555555173 <+42>:    mov    $0x0,%eax
   0x0000555555555178 <+47>:    leaveq
   0x0000555555555179 <+48>:    retq
End of assembler dump.
(gdb) quit
student@ai-HP-ProDesk-600-G4-MT:~$ 
```

3.

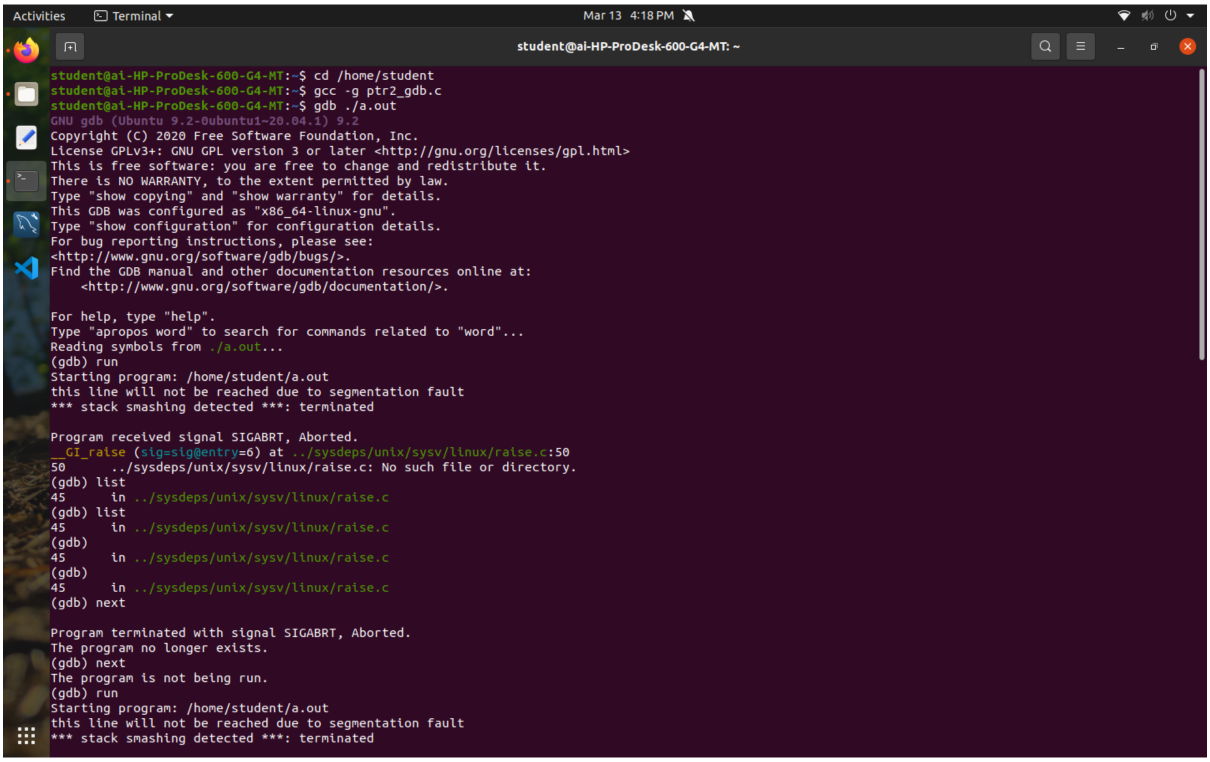


```c
#include <stdio.h>
int main(){
    int arr[5];
    int i;
    for (i=0;i<=10;i++){
        arr[i]=i;
    }
    printf("this line will not be reached due to segmentation fault\n");
    return 0;
}

```

Loading file "/home/student/ptr2_gdb.c"... C Tab Width: 8 Ln 8, Col 73 INS

OUTPUT:

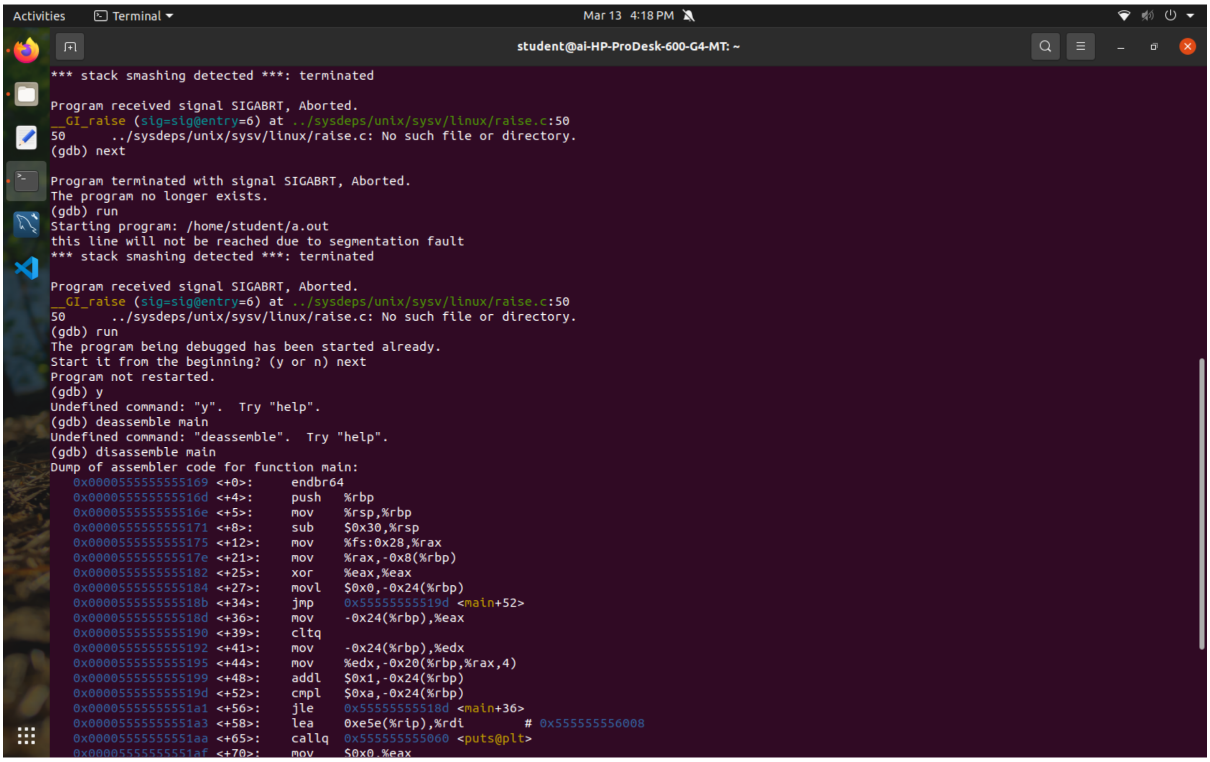


```
student@ai-HP-ProDesk-600-G4-MT:~$ cd /home/student
student@ai-HP-ProDesk-600-G4-MT:~$ gcc -g ptr2_gdb.c
student@ai-HP-ProDesk-600-G4-MT:~$ gdb ./a.out
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04.1) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
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) run
Starting program: /home/student/a.out
this line will not be reached due to segmentation fault
*** stack smashing detected ***: terminated

Program received signal SIGABRT, Aborted.
__GI_raise (sig=sig@entry=6) at ../sysdeps/unix/sysv/linux/raise.c:50
50      ../sysdeps/unix/sysv/linux/raise.c: No such file or directory.
(gdb) list
45      in ../sysdeps/unix/sysv/linux/raise.c
(gdb) list
45      in ../sysdeps/unix/sysv/linux/raise.c
(gdb)
45      in ../sysdeps/unix/sysv/linux/raise.c
(gdb)
45      in ../sysdeps/unix/sysv/linux/raise.c
(gdb) next

Program terminated with signal SIGABRT, Aborted.
The program no longer exists.
(gdb) next
The program is not being run.
(gdb) run
Starting program: /home/student/a.out
this line will not be reached due to segmentation fault
*** stack smashing detected ***: terminated
```

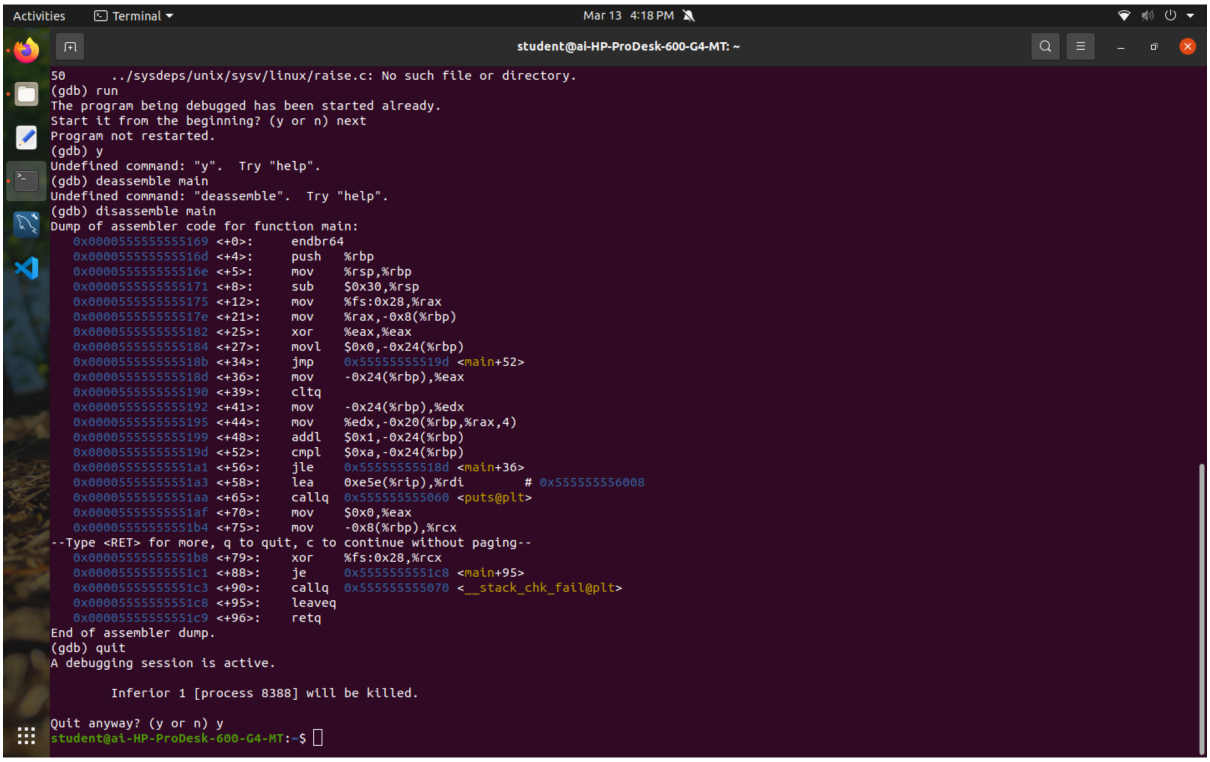


```
*** stack smashing detected ***: terminated

Program received signal SIGABRT, Aborted.
__GI_raise (sig=sig@entry=6) at ../sysdeps/unix/sysv/linux/raise.c:50
50      ../sysdeps/unix/sysv/linux/raise.c: No such file or directory.
(gdb) next

Program terminated with signal SIGABRT, Aborted.
The program no longer exists.
(gdb) run
Starting program: /home/student/a.out
this line will not be reached due to segmentation fault
*** stack smashing detected ***: terminated

Program received signal SIGABRT, Aborted.
__GI_raise (sig=sig@entry=6) at ../sysdeps/unix/sysv/linux/raise.c:50
50      ../sysdeps/unix/sysv/linux/raise.c: No such file or directory.
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) next
Program not restarted.
(gdb) y
Undefined command: "y".  Try "help".
(gdb) deassemble main
Undefined command: "deassemble".  Try "help".
(gdb) disassemble main
Dump of assembler code for function main:
   0x0000555555555169 <+0>:     endbr64
   0x000055555555516d <+4>:     push   %rbp
   0x000055555555516e <+5>:     mov    %rsp,%rbp
   0x0000555555555171 <+8>:     sub    $0x30,%rsp
   0x0000555555555175 <+12>:    mov    %fs:0x28,%rax
   0x000055555555517e <+21>:    mov    %rax,-0x8(%rbp)
   0x0000555555555182 <+25>:    xor    %eax,%eax
   0x0000555555555184 <+27>:    movl   $0x0,-0x24(%rbp)
   0x000055555555518b <+34>:    jmp    0x55555555519d <main+52>
   0x000055555555518d <+36>:    mov    -0x24(%rbp),%eax
   0x0000555555555190 <+39>:    cltq
   0x0000555555555192 <+41>:    mov    -0x24(%rbp),%edx
   0x0000555555555195 <+44>:    mov    %edx,-0x20(%rbp,%rax,4)
   0x0000555555555199 <+48>:    addl   $0x1,-0x24(%rbp)
   0x000055555555519d <+52>:    cmpl   $0xa,-0x24(%rbp)
   0x00005555555551a1 <+56>:    jle    0x55555555518d <main+36>
   0x00005555555551a3 <+58>:    lea    0xe5e(%rip),%rdi        # 0x555555556008
   0x00005555555551aa <+65>:    callq  0x555555555060 <puts@plt>
   0x00005555555551af <+70>:    mov    $0x0,%eax
```



```
50      ../sysdeps/unix/sysv/linux/raise.c: No such file or directory.
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) next
Program not restarted.
(gdb) y
Undefined command: "y".  Try "help".
(gdb) deassemble main
Undefined command: "deassemble".  Try "help".
(gdb) disassemble main
Dump of assembler code for function main:
   0x0000555555555169 <+0>:     endbr64
   0x000055555555516d <+4>:     push   %rbp
   0x000055555555516e <+5>:     mov    %rsp,%rbp
   0x0000555555555171 <+8>:     sub    $0x30,%rsp
   0x0000555555555175 <+12>:    mov    %fs:0x28,%rax
   0x000055555555517e <+21>:    mov    %rax,-0x8(%rbp)
   0x0000555555555182 <+25>:    xor    %eax,%eax
   0x0000555555555184 <+27>:    movl   $0x0,-0x24(%rbp)
   0x000055555555518b <+34>:    jmp    0x55555555519d <main+52>
   0x000055555555518d <+36>:    mov    -0x24(%rbp),%eax
   0x0000555555555190 <+39>:    cltq
   0x0000555555555192 <+41>:    mov    -0x24(%rbp),%edx
   0x0000555555555195 <+44>:    mov    %edx,-0x20(%rbp,%rax,4)
   0x0000555555555199 <+48>:    addl   $0x1,-0x24(%rbp)
   0x000055555555519d <+52>:    cmpl   $0xa,-0x24(%rbp)
   0x00005555555551a1 <+56>:    jle    0x55555555518d <main+36>
   0x00005555555551a3 <+58>:    lea    0xe5e(%rip),%rdi        # 0x555555556008
   0x00005555555551aa <+65>:    callq  0x555555555060 <puts@plt>
   0x00005555555551af <+70>:    mov    $0x0,%eax
   0x00005555555551b4 <+75>:    mov    -0x8(%rbp),%rcx
--Type <RET> for more, q to quit, c to continue without paging--
   0x00005555555551b8 <+79>:    xor    %fs:0x28,%rcx
   0x00005555555551c1 <+88>:    je     0x5555555551c8 <main+95>
   0x00005555555551c3 <+90>:    callq  0x555555555070 <__stack_chk_fail@plt>
   0x00005555555551c8 <+95>:    leaveq
   0x00005555555551c9 <+96>:    retq
End of assembler dump.
(gdb) quit
A debugging session is active.

        Inferior 1 [process 8388] will be killed.

Quit anyway? (y or n) y
student@ai-HP-ProDesk-600-G4-MT:~$ 
```