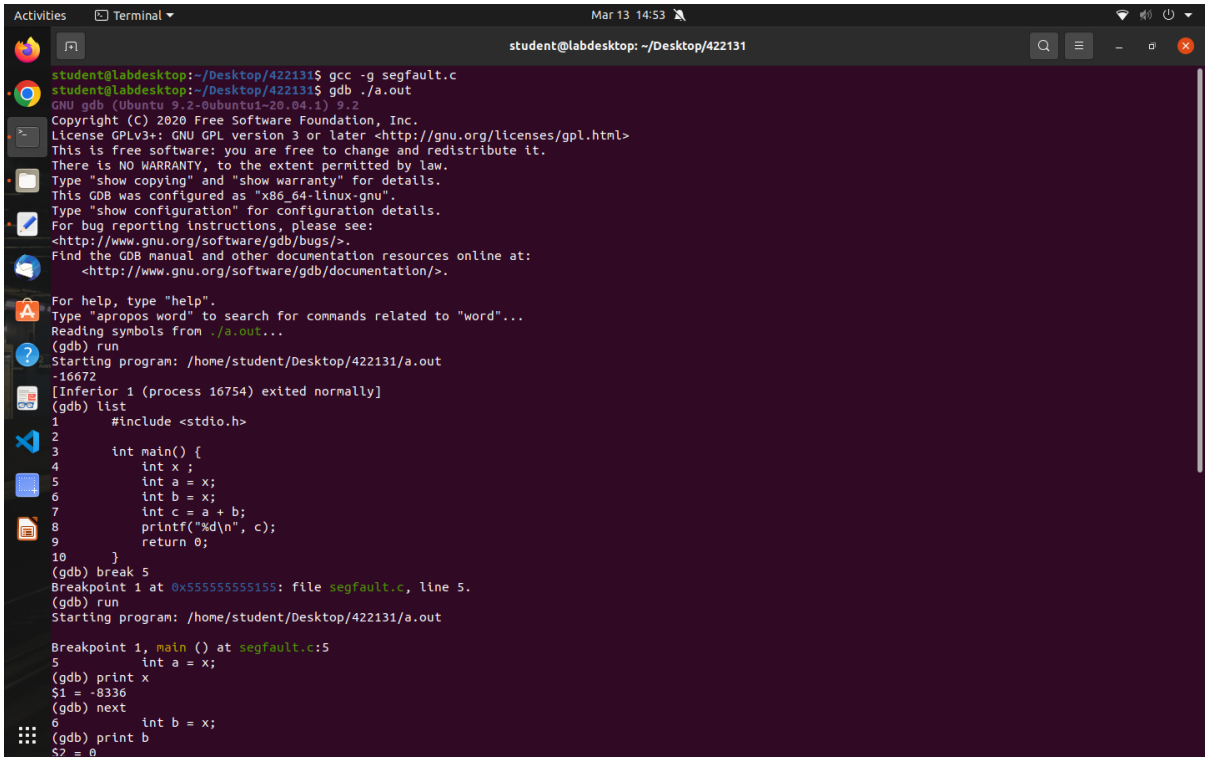
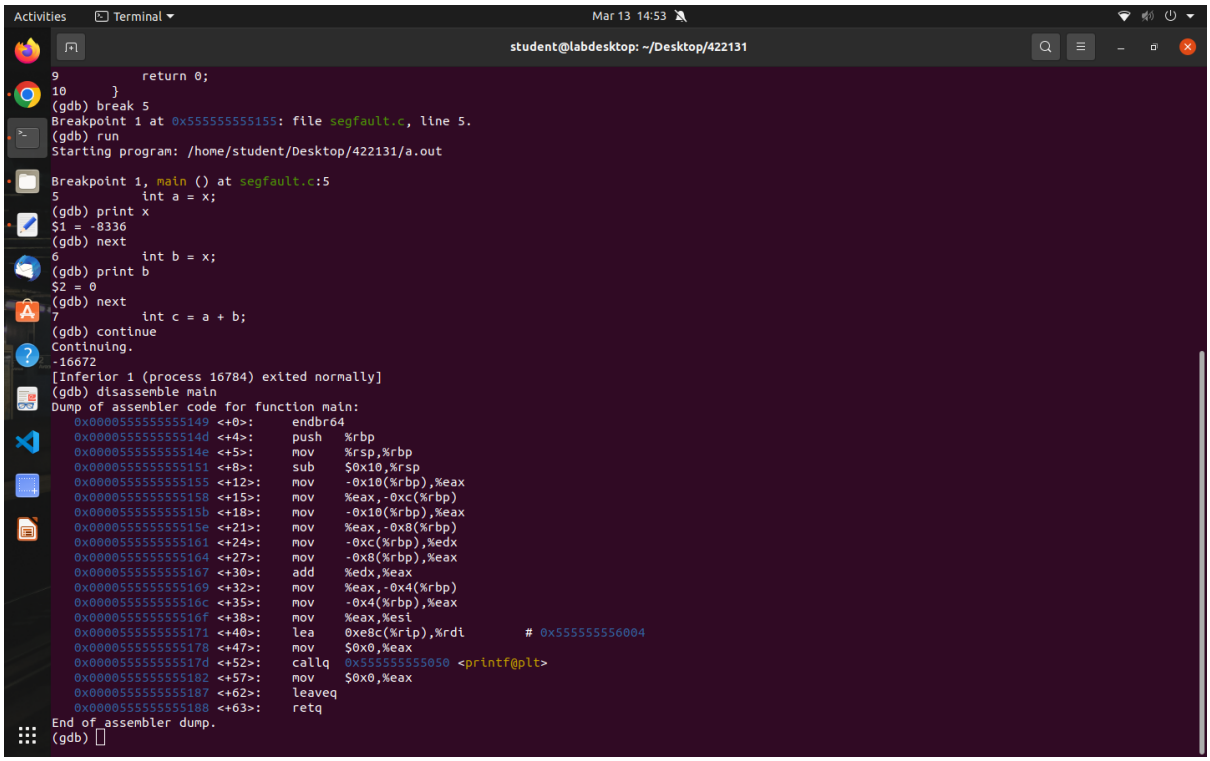
NAME: CH.YESWANTH

ROLLNO : 422131

First program

```
#include <stdio.h>


int main() {
   int x = 10;
   int a = x;
   int b = x;
   int c = a + b;
   printf("%d\n", c);
   return 0;
}
```

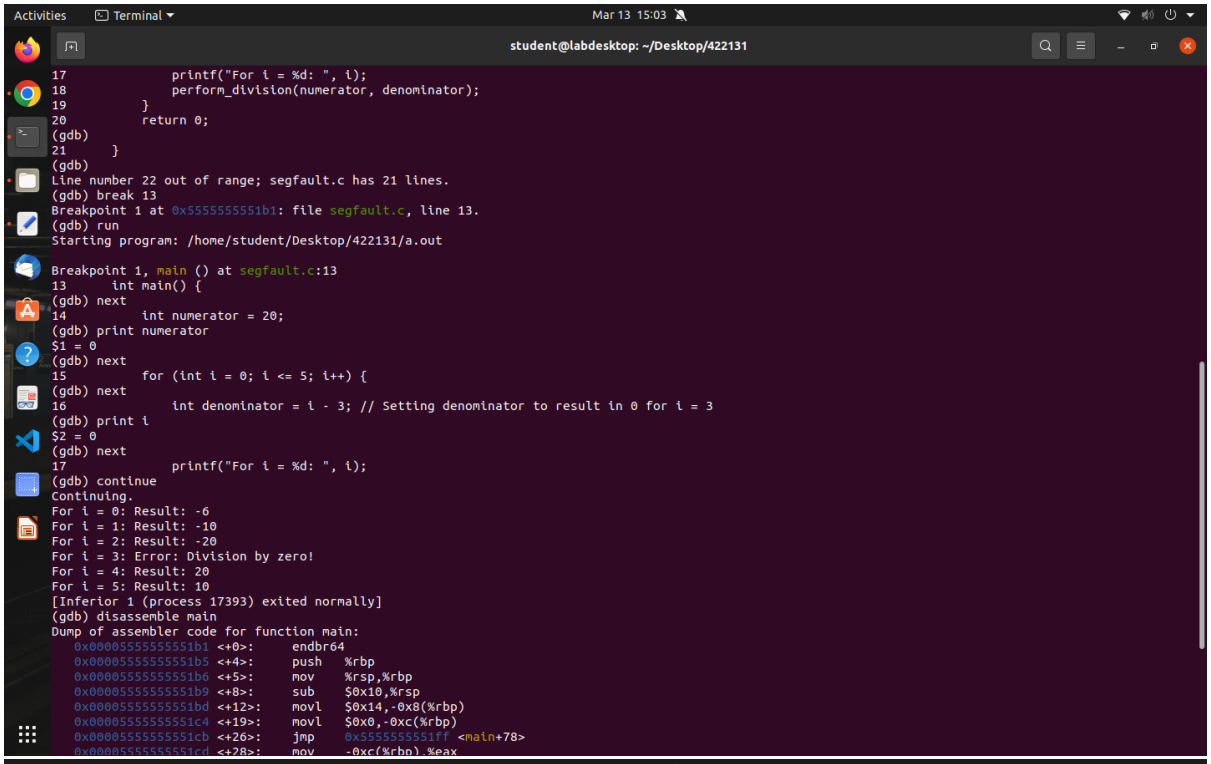

Activities
Terminal
Mar 13 14:53
student@labdesktop: ~/Desktop/422131
student@labdesktop:~/Desktop/422131$ gcc -g segfault.c
student@labdesktop:~/Desktop/422131$ gdb ./a.out
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04.1) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or 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/Desktop/422131/a.out
-16672
[Inferior 1 (process 16754) exited normally]
(gdb) list
1 #include <stdio.h>
2
3 int main() {
4 int x ;
5 int a = x;
6 int b = x;
7 int c = a + b;
8 printf("%d\n", c);
9 return 0;
10 }
(gdb) break 5
Breakpoint 1 at 0x555555555155: file segfault.c, line 5.
(gdb) run
Starting program: /home/student/Desktop/422131/a.out
Breakpoint 1, main () at segfault.c:5
5 int a = x;
(gdb) print x
$1 = -8336
(gdb) next
6 int b = x;
(gdb) print b
$2 = 0

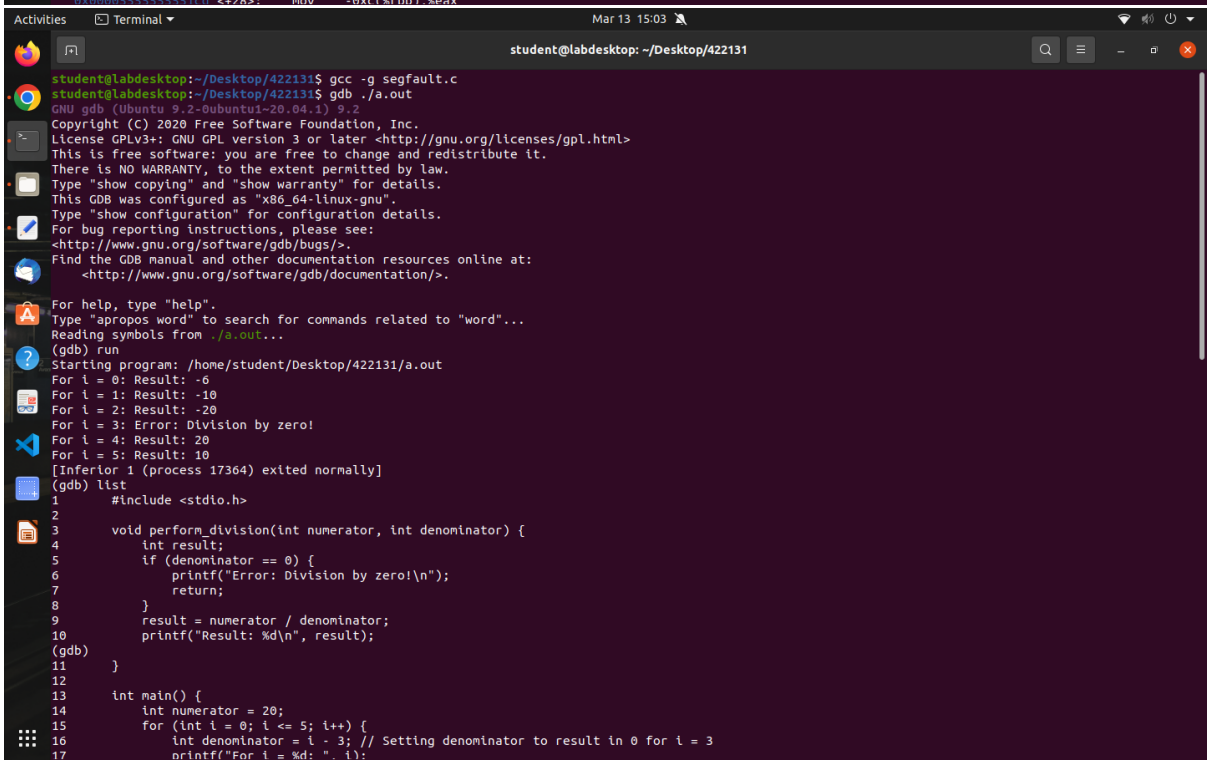

Activities
Terminal
Mar 13 14:53
student@labdesktop: ~/Desktop/422131
9 return 0;
10 }
(gdb) break 5
Breakpoint 1 at 0x555555555155: file segfault.c, line 5.
(gdb) run
Starting program: /home/student/Desktop/422131/a.out
Breakpoint 1, main () at segfault.c:5
5 int a = x;
(gdb) print x
$1 = -8336
(gdb) next
6 int b = x;
(gdb) print b
$2 = 0
(gdb) next
7 int c = a + b;
(gdb) continue
Continuing.
-16672
[Inferior 1 (process 16784) exited normally]
(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>: mov -0x10(%rbp),%eax
0x0000555555555158 <+15>: mov %eax,-0xc(%rbp)
0x000055555555515b <+18>: mov -0x10(%rbp),%eax
0x000055555555515e <+21>: mov %eax,-0x8(%rbp)
0x0000555555555161 <+24>: mov -0xc(%rbp),%edx
0x0000555555555164 <+27>: mov -0x8(%rbp),%eax
0x0000555555555167 <+30>: add %edx,%eax
0x0000555555555169 <+32>: mov %eax,-0x4(%rbp)
0x000055555555516c <+35>: mov -0x4(%rbp),%eax
0x000055555555516f <+38>: mov %eax,%esi
0x0000555555555171 <+40>: lea 0xe8c(%rip),%rdi # 0x555555556004
0x0000555555555178 <+47>: mov $0x0,%eax
0x000055555555517d <+52>: callq 0x555555555050 <printf@plt>
0x0000555555555182 <+57>: mov $0x0,%eax
0x0000555555555187 <+62>: leaveq
0x0000555555555188 <+63>: retq
End of assembler dump.
(gdb)

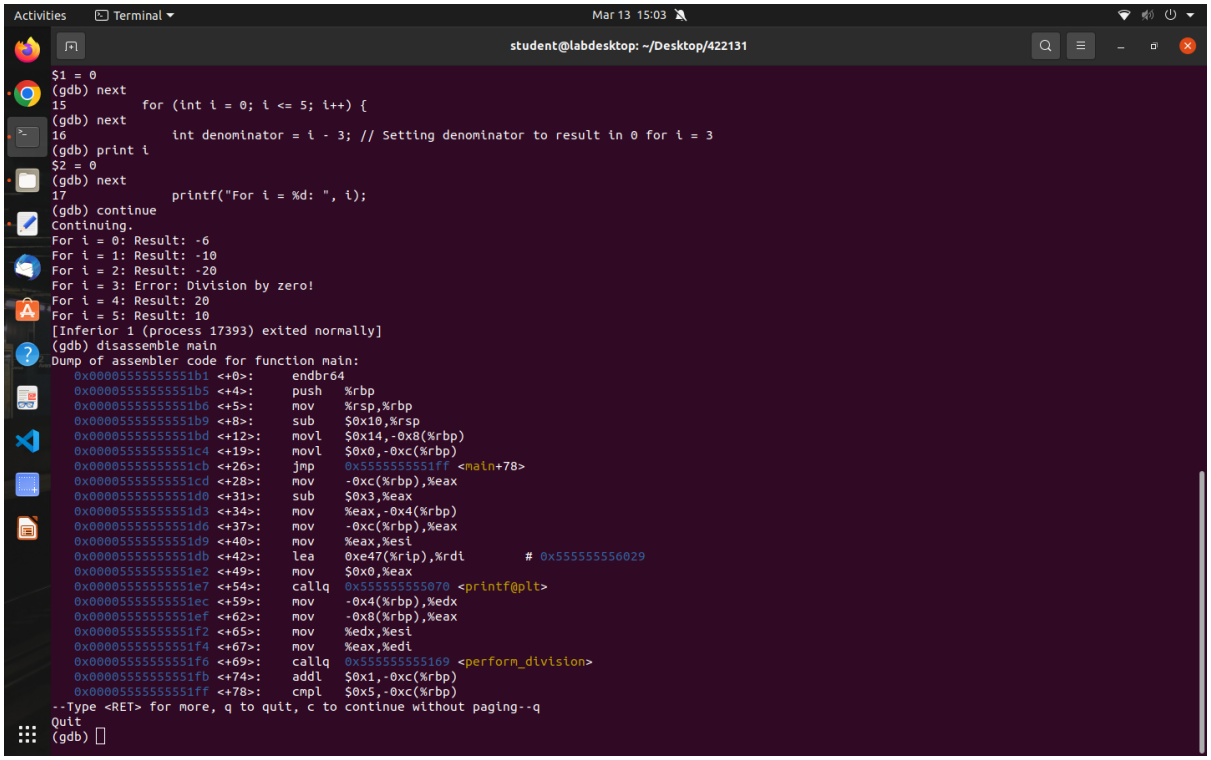
Second code:

```
#include <stdio.h>

void perform_division(int numerator, int denominator) {
   int result;
   if (denominator == 0) {
      printf("Error: Division by zero!\n");
      return;
   }
   result = numerator / denominator;
   printf("Result: %d\n", result);
}

int main() {
   int numerator = 20;
   for (int i = 0; i <= 5; i++) {
      int denominator = i - 3; // Setting denominator to result in 0 for i = 3
      printf("For i = %d: ", i);
      perform_division(numerator, denominator);
   return 0;
}
```

}

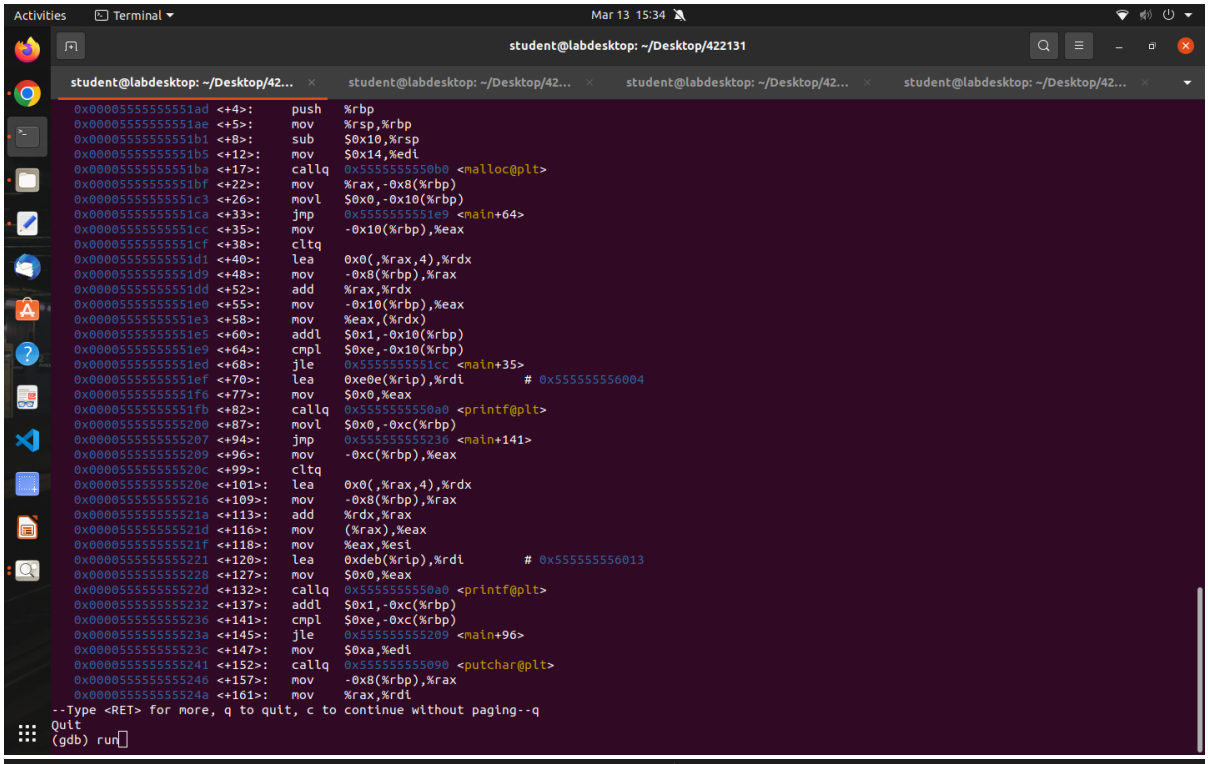

Activities
Terminal
Mar 13 15:03
student@labdesktop: ~/Desktop/422131
17 printf("For i = %d: ", i);
18 perform_division(numerator, denominator);
19 }
20 return 0;
(gdb)
21 }
(gdb)
Line number 22 out of range; segfault.c has 21 lines.
(gdb) break 13
Breakpoint 1 at 0x5555555551b1: file segfault.c, line 13.
(gdb) run
Starting program: /home/student/Desktop/422131/a.out
Breakpoint 1, main () at segfault.c:13
13 int main() {
(gdb) next
14 int numerator = 20;
(gdb) print numerator
$1 = 0
(gdb) next
15 for (int i = 0; i <= 5; i++) {
(gdb) next
16 int denominator = i - 3; // Setting denominator to result in 0 for i = 3
(gdb) print i
$2 = 0
(gdb) next
17 printf("For i = %d: ", i);
(gdb) continue
Continuing.
For i = 0: Result: -6
For i = 1: Result: -10
For i = 2: Result: -20
For i = 3: Error: Division by zero!
For i = 4: Result: 20
For i = 5: Result: 10
[Inferior 1 (process 17393) exited normally]
(gdb) disassemble main
Dump of assembler code for function main:
0x00005555555551b1 <+0>: endbr64
0x00005555555551b5 <+4>: push %rbp
0x00005555555551b6 <+5>: mov %rsp,%rbp
0x00005555555551b9 <+8>: sub $0x10,%rsp
0x00005555555551bd <+12>: movl $0x14,-0x8(%rbp)
0x00005555555551c4 <+19>: movl $0x0,-0xc(%rbp)
0x00005555555551cb <+26>: jmp 0x555555551ff <main+78>
0x00005555555551cd <+28>: mov -0xc(%rbp),%eax

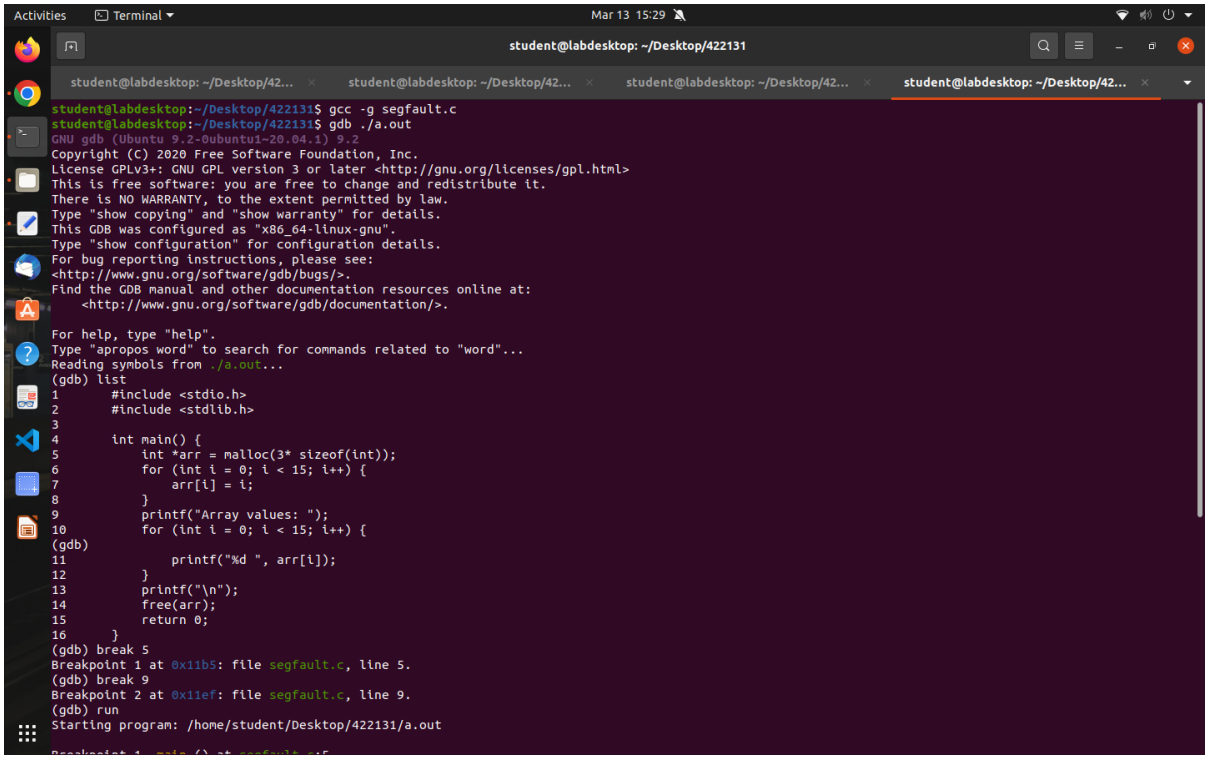

Activities
Terminal
Mar 13 15:03
student@labdesktop: ~/Desktop/422131
student@labdesktop:~/Desktop/422131$ gcc -g segfault.c
student@labdesktop:~/Desktop/422131$ gdb ./a.out
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04.1) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or 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/Desktop/422131/a.out
For i = 0: Result: -6
For i = 1: Result: -10
For i = 2: Result: -20
For i = 3: Error: Division by zero!
For i = 4: Result: 20
For i = 5: Result: 10
[Inferior 1 (process 17364) exited normally]
(gdb) list
1 #include <stdio.h>
2
3 void perform_division(int numerator, int denominator) {
4 int result;
5 if (denominator == 0) {
6 printf("Error: Division by zero!\n");
7 return;
8 }
9 result = numerator / denominator;
10 printf("Result: %d\n", result);
(gdb)
11 }
12
13 int main() {
14 int numerator = 20;
15 for (int i = 0; i <= 5; i++) {
16 int denominator = i - 3; // Setting denominator to result in 0 for i = 3
17 printf("For i = %d: ", i);

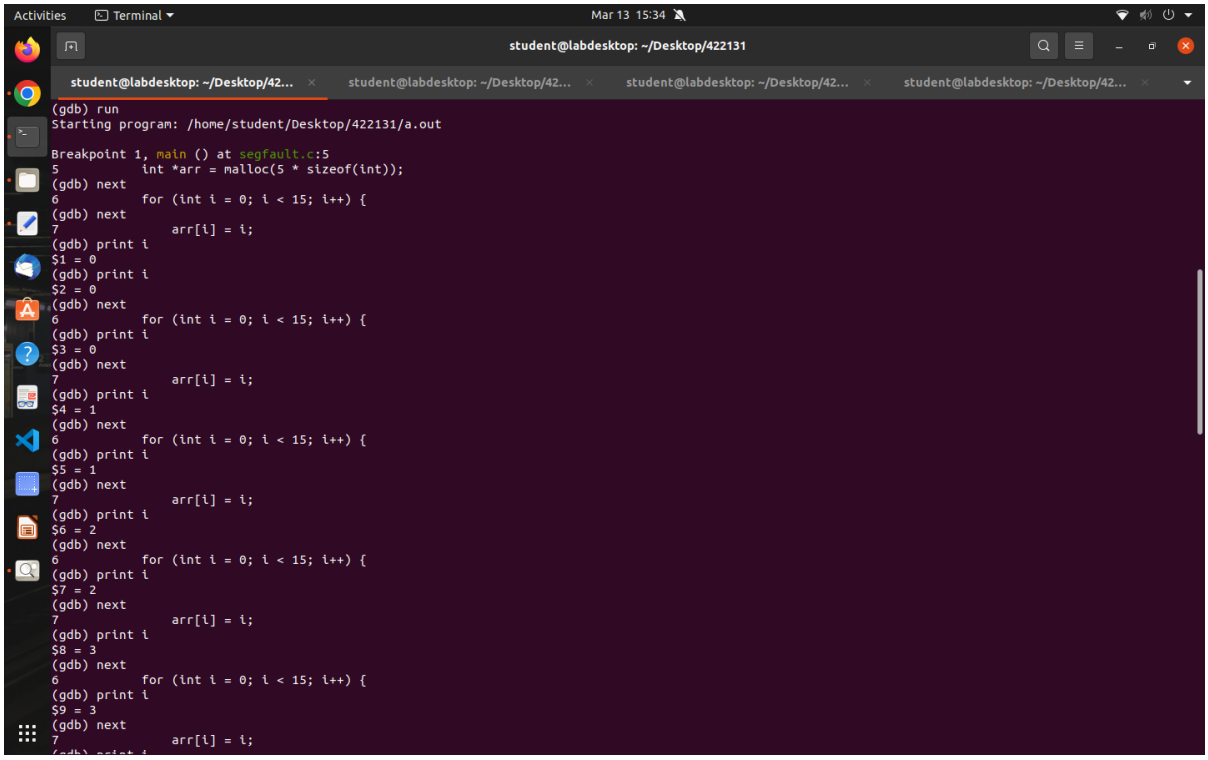


THIRD CODE:

```
#include <stdio.h>

#include <stdlib.h>


int main() {

   int *arr = malloc(10 * sizeof(int));

   for (int i = 0; i < 15; i++) {

      arr[i] = i;

   }

   printf("Array values: ");

   for (int i = 0; i < 15; i++) {

      printf("%d ", arr[i]);

   }

   printf("\n");

   free(arr);

   return 0;

}
```


Activities
Terminal
Mar 13 15:34
student@labdesktop: ~/Desktop/422131
0x00005555555551ad <+4>: push %rbp
0x00005555555551ae <+5>: mov %rsp,%rbp
0x00005555555551b1 <+8>: sub $0x10,%rsp
0x00005555555551b5 <+12>: mov $0x14,%edi
0x00005555555551ba <+17>: callq 0x5555555550b0 <malloc@plt>
0x00005555555551bf <+22>: mov %rax,-0x8(%rbp)
0x00005555555551c3 <+26>: movl $0x0,-0x10(%rbp)
0x00005555555551ca <+33>: jmp 0x5555555551e9 <main+64>
0x00005555555551cc <+35>: mov -0x10(%rbp),%eax
0x00005555555551cf <+38>: cltq
0x00005555555551d1 <+40>: lea 0x0(,%rax,4),%rdx
0x00005555555551d9 <+48>: mov -0x8(%rbp),%rax
0x00005555555551dd <+52>: add %rax,%rdx
0x00005555555551e0 <+55>: mov -0x10(%rbp),%eax
0x00005555555551e3 <+58>: mov %eax,(%rdx)
0x00005555555551e5 <+60>: addl $0x1,-0x10(%rbp)
0x00005555555551e9 <+64>: cmpl $0xe,-0x10(%rbp)
0x00005555555551ed <+68>: jle 0x5555555551cc <main+35>
0x00005555555551ef <+70>: lea 0xe0e(%rip),%rdi # 0x555555556004
0x00005555555551f6 <+77>: mov $0x0,%eax
0x00005555555551fb <+82>: callq 0x5555555550a0 <printf@plt>
0x0000555555555200 <+87>: movl $0x0,-0xc(%rbp)
0x0000555555555207 <+94>: jmp 0x555555555236 <main+141>
0x0000555555555209 <+96>: mov -0xc(%rbp),%eax
0x000055555555520c <+99>: cltq
0x000055555555520e <+101>: lea 0x0(,%rax,4),%rdx
0x0000555555555216 <+109>: mov -0x8(%rbp),%rax
0x000055555555521a <+113>: add %rdx,%rax
0x000055555555521d <+116>: mov (%rax),%eax
0x000055555555521f <+118>: mov %eax,%esi
0x0000555555555221 <+120>: lea 0xdeb(%rip),%rdi # 0x555555556013
0x0000555555555228 <+127>: mov $0x0,%eax
0x000055555555522d <+132>: callq 0x5555555550a0 <printf@plt>
0x0000555555555232 <+137>: addl $0x1,-0xc(%rbp)
0x0000555555555236 <+141>: cmpl $0xe,-0xc(%rbp)
0x000055555555523a <+145>: jle 0x555555555209 <main+96>
0x000055555555523c <+147>: mov $0xa,%edi
0x0000555555555241 <+152>: callq 0x5555555555090 <putchar@plt>
0x0000555555555246 <+157>: mov -0x8(%rbp),%rax
0x000055555555524a <+161>: mov %rax,%rdi
--Type <RET> for more, q to quit, c to continue without paging--q
Quit
(gdb) run


Activities
Terminal
Mar 13 15:29
student@labdesktop: ~/Desktop/422131
student@labdesktop:~/Desktop/422131$ gcc -g segfault.c
student@labdesktop:~/Desktop/422131$ 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 #include <stdlib.h>
3
4 int main() {
5 int *arr = malloc(3* sizeof(int));
6 for (int i = 0; i < 15; i++) {
7 arr[i] = i;
8 }
9 printf("Array values: ");
10 for (int i = 0; i < 15; i++) {
(gdb)
11 printf("%d ", arr[i]);
12 }
13 printf("\n");
14 free(arr);
15 return 0;
16 }
(gdb) break 5
Breakpoint 1 at 0x11b5: file segfault.c, line 5.
(gdb) break 9
Breakpoint 2 at 0x11ef: file segfault.c, line 9.
(gdb) run
Starting program: /home/student/Desktop/422131/a.out


Activities
Terminal
Mar 13 15:34
student@labdesktop: ~/Desktop/422131
student@labdesktop: ~/Desktop/42...
student@labdesktop: ~/Desktop/42...
student@labdesktop: ~/Desktop/42...
student@labdesktop: ~/Desktop/42...
(gdb) run
Starting program: /home/student/Desktop/422131/a.out
Breakpoint 1, main () at segfault.c:5
5 int *arr = malloc(5 * sizeof(int));
(gdb) next
6 for (int i = 0; i < 15; i++) {
(gdb) next
7 arr[i] = i;
(gdb) print i
$1 = 0
(gdb) print i
$2 = 0
(gdb) next
6 for (int i = 0; i < 15; i++) {
(gdb) print i
$3 = 0
(gdb) next
7 arr[i] = i;
(gdb) print i
$4 = 1
(gdb) next
6 for (int i = 0; i < 15; i++) {
(gdb) print i
$5 = 1
(gdb) next
7 arr[i] = i;
(gdb) print i
$6 = 2
(gdb) next
6 for (int i = 0; i < 15; i++) {
(gdb) print i
$7 = 2
(gdb) next
7 arr[i] = i;
(gdb) print i
$8 = 3
(gdb) next
6 for (int i = 0; i < 15; i++) {
(gdb) print i
$9 = 3
(gdb) next
7 arr[i] = i;