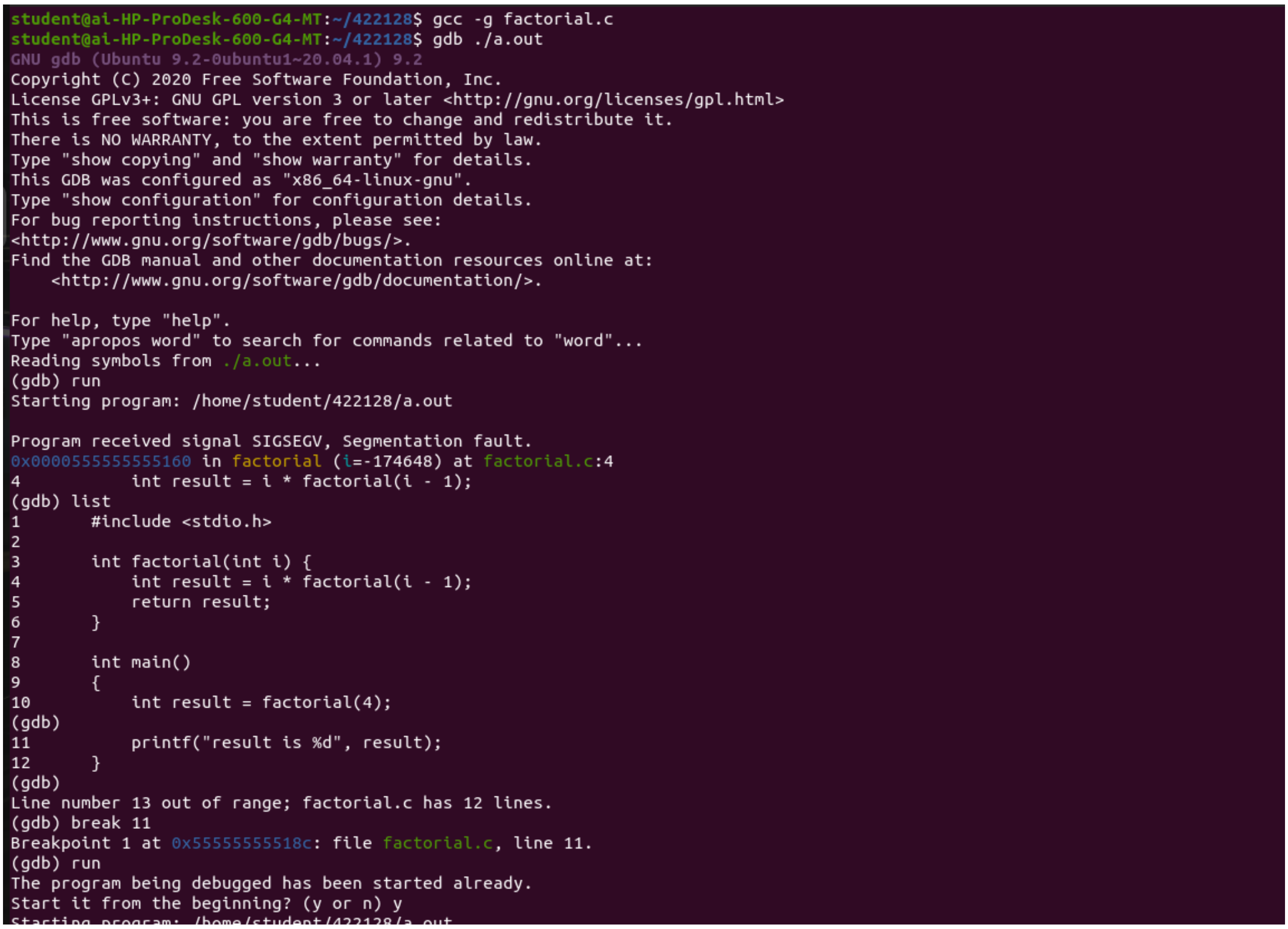
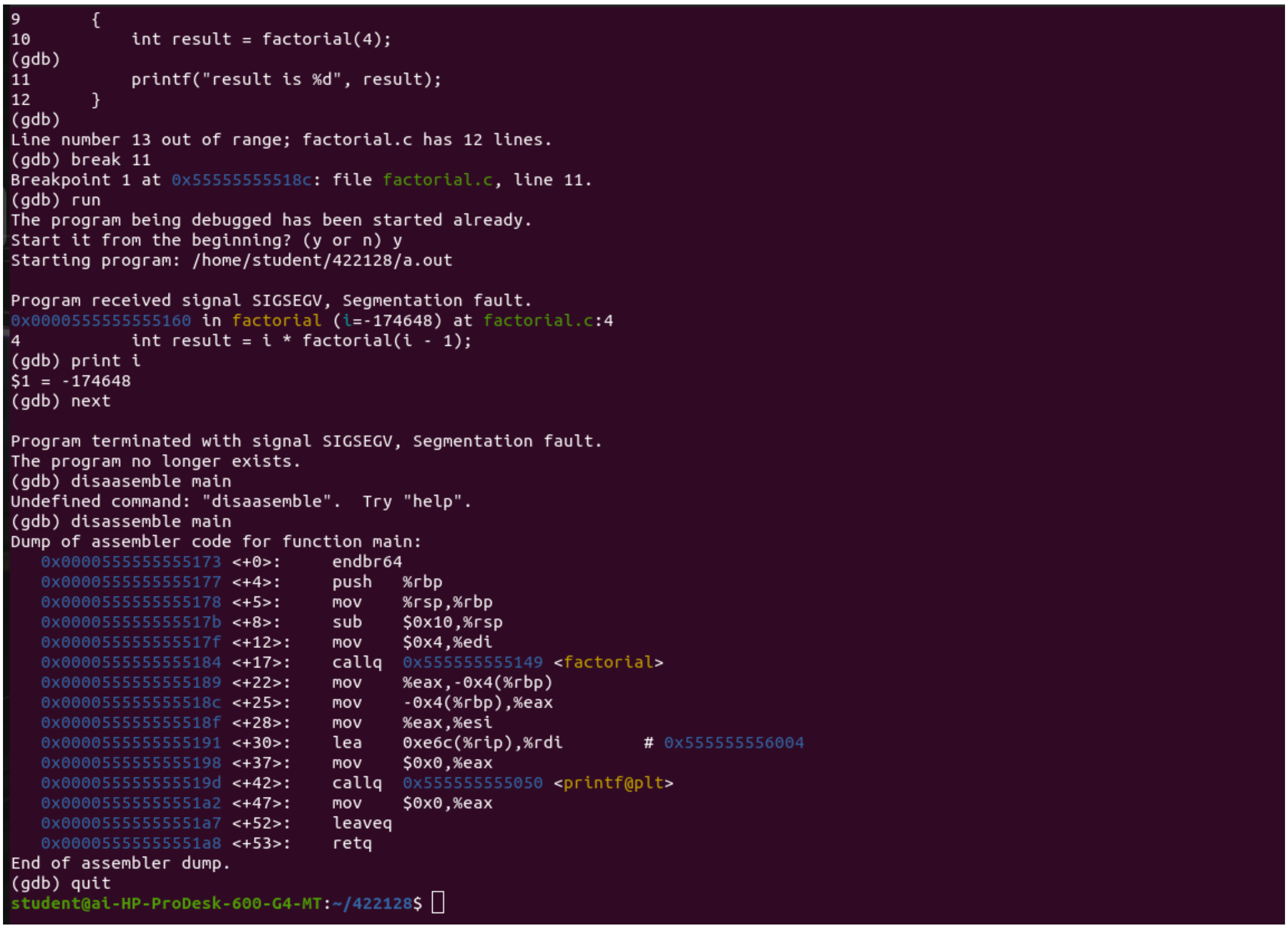
NAME : C.Sai Charitha
ROLL NO : 422128
SECTION : A

FACTORIAL :

```
#include <stdio.h>

int factorial(int i) {
    int result = i * factorial(i - 1);
    return result;
}

int main()
{
    int result = factorial(4);
    printf("result is %d", result);
}
```



```
student@ai-HP-ProDesk-600-G4-MT:~/422128$ gcc -g factorial.c
student@ai-HP-ProDesk-600-G4-MT:~/422128$ 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/422128/a.out

Program received signal SIGSEGV, Segmentation fault.
0x0000555555555160 in factorial (i=-174648) at factorial.c:4
4           int result = i * factorial(i - 1);
(gdb) list
1       #include <stdio.h>
2
3       int factorial(int i) {
4           int result = i * factorial(i - 1);
5           return result;
6       }
7
8       int main()
9       {
10          int result = factorial(4);
(gdb)
11          printf("result is %d", result);
12      }
(gdb)
Line number 13 out of range; factorial.c has 12 lines.
(gdb) break 11
Breakpoint 1 at 0x55555555518c: file factorial.c, line 11.
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/student/422128/a.out
```

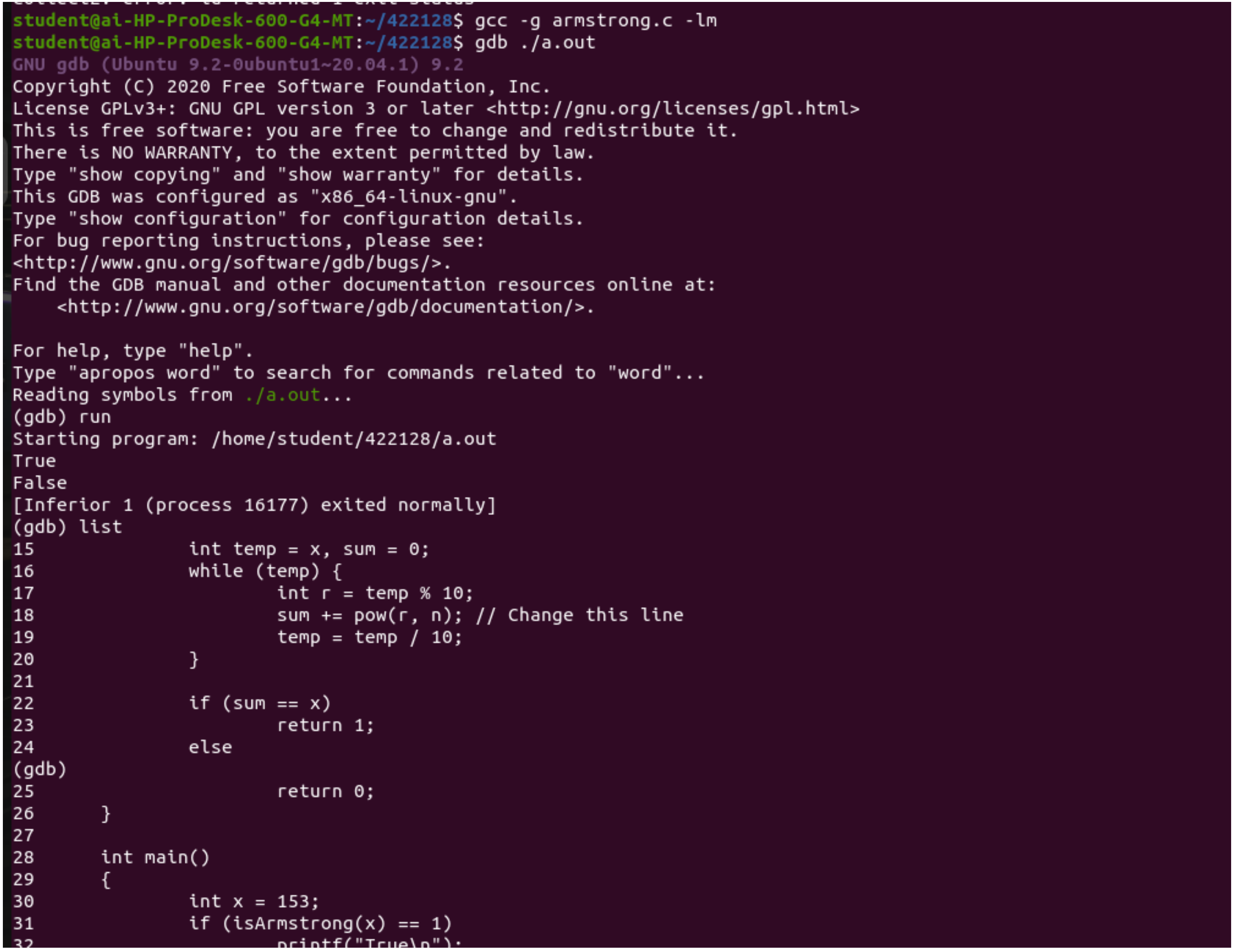


```
9       {
10          int result = factorial(4);
(gdb)
11          printf("result is %d", result);
12      }
(gdb)
Line number 13 out of range; factorial.c has 12 lines.
(gdb) break 11
Breakpoint 1 at 0x55555555518c: file factorial.c, line 11.
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/student/422128/a.out

Program received signal SIGSEGV, Segmentation fault.
0x0000555555555160 in factorial (i=-174648) at factorial.c:4
4           int result = i * factorial(i - 1);
(gdb) print i
$1 = -174648
(gdb) next

Program terminated with signal SIGSEGV, Segmentation fault.
The program no longer exists.
(gdb) disaasemble main
Undefined command: "disaasemble".  Try "help".
(gdb) disassemble main
Dump of assembler code for function main:
   0x0000555555555173 <+0>:     endbr64
   0x0000555555555177 <+4>:     push   %rbp
   0x0000555555555178 <+5>:     mov    %rsp,%rbp
   0x000055555555517b <+8>:     sub    $0x10,%rsp
   0x000055555555517f <+12>:    mov    $0x4,%edi
   0x0000555555555184 <+17>:    callq  0x555555555149 <factorial>
   0x0000555555555189 <+22>:    mov    %eax,-0x4(%rbp)
   0x000055555555518c <+25>:    mov    -0x4(%rbp),%eax
   0x000055555555518f <+28>:    mov    %eax,%esi
   0x0000555555555191 <+30>:    lea    0xe6c(%rip),%rdi        # 0x555555556004
   0x0000555555555198 <+37>:    mov    $0x0,%eax
   0x000055555555519d <+42>:    callq  0x555555555050 <printf@plt>
   0x00005555555551a2 <+47>:    mov    $0x0,%eax
   0x00005555555551a7 <+52>:    leaveq
   0x00005555555551a8 <+53>:    retq
End of assembler dump.
(gdb) quit
student@ai-HP-ProDesk-600-G4-MT:~/422128$ 
```

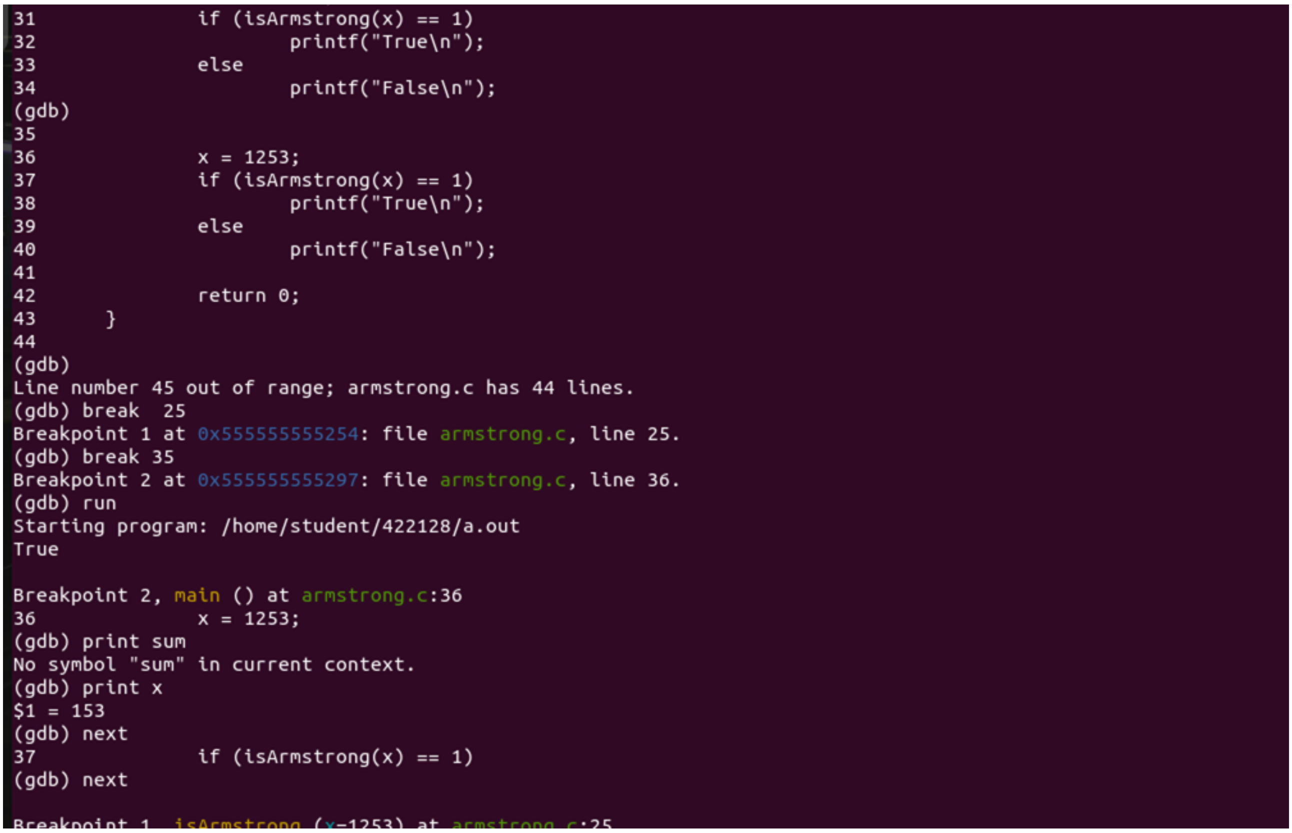
ARMSTRONG NUMBER :

```
#include <stdio.h>
#include <math.h>
int order(int x)
{
        int n = 0;
        while (x) {
                n++;
                x = x / 10;
        }
        return n;
}
int isArmstrong(int x)
{
        int n = order(x);
        int temp = x, sum = 0;
        while (temp) {
                int r = temp % 10;
                sum += pow(r, n); // Change this line
                temp = temp / 10;
        }

        if (sum == x)
                return 1;
        else
```

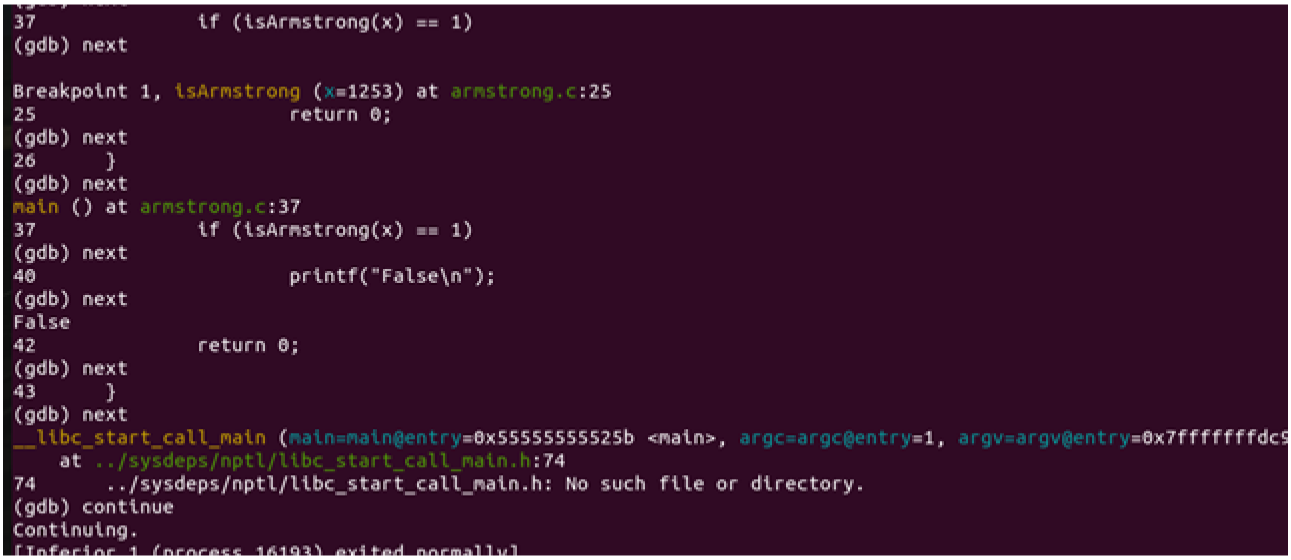
```
		return 0;
}

int main()
{
	int x = 153;
	if (isArmstrong(x) == 1)
		printf("True\n");
	else
		printf("False\n");

	x = 1253;
	if (isArmstrong(x) == 1)
		printf("True\n");
	else
		printf("False\n");

	return 0;
}
```



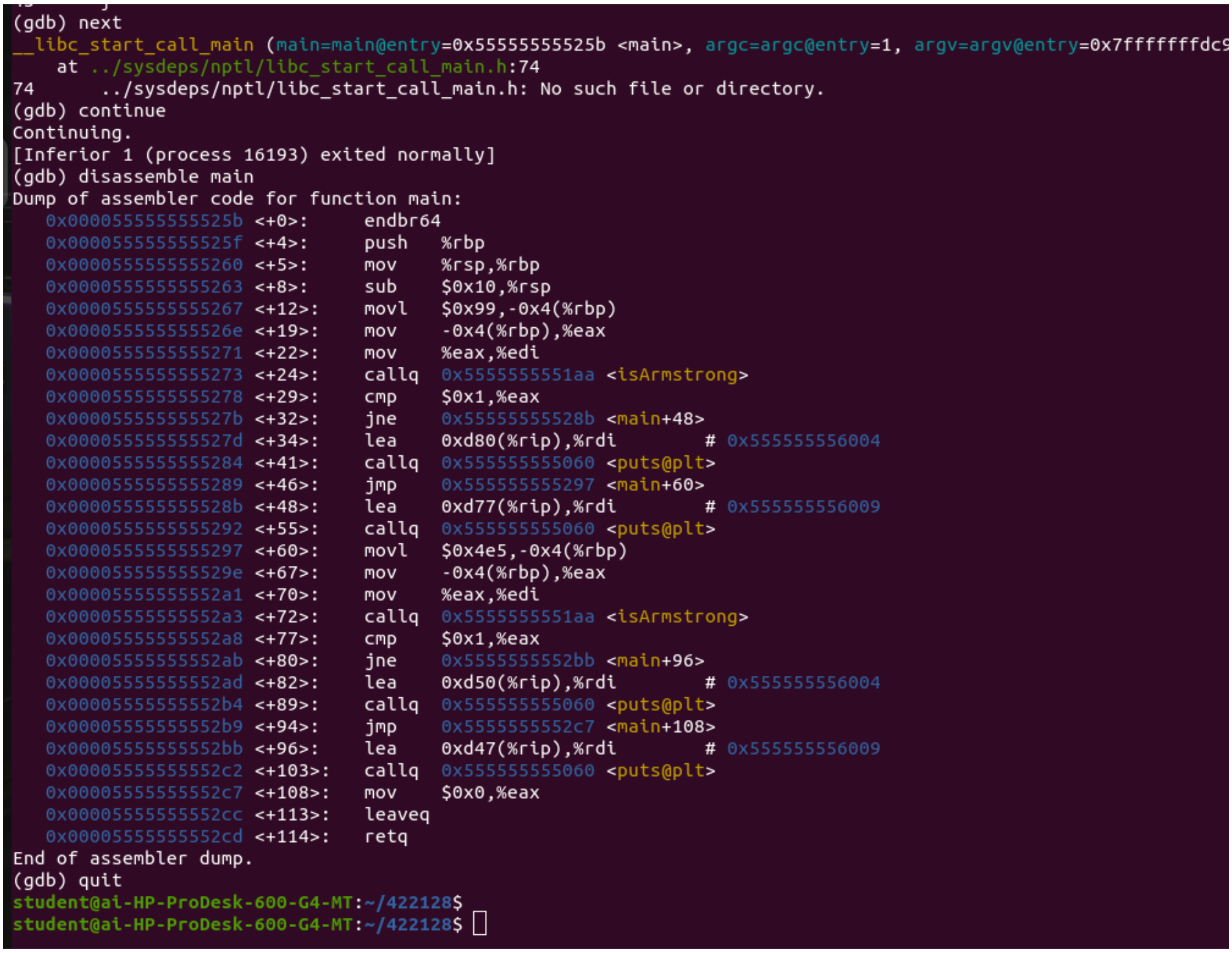
```
student@ai-HP-ProDesk-600-G4-MT:~/422128$ gcc -g armstrong.c -lm
student@ai-HP-ProDesk-600-G4-MT:~/422128$ gdb ./a.out
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04.1) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is 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/422128/a.out
True
False
[Inferior 1 (process 16177) exited normally]
(gdb) list
15              int temp = x, sum = 0;
16              while (temp) {
17                      int r = temp % 10;
18                      sum += pow(r, n); // Change this line
19                      temp = temp / 10;
20              }
21
22              if (sum == x)
23                      return 1;
24              else
(gdb)
25                      return 0;
26      }
27
28      int main()
29      {
30              int x = 153;
31              if (isArmstrong(x) == 1)
32                      printf("True\n");
```



```
31              if (isArmstrong(x) == 1)
32                      printf("True\n");
33              else
34                      printf("False\n");
(gdb)
35
36              x = 1253;
37              if (isArmstrong(x) == 1)
38                      printf("True\n");
39              else
40                      printf("False\n");
41
42              return 0;
43      }
44
(gdb)
Line number 45 out of range; armstrong.c has 44 lines.
(gdb) break  25
Breakpoint 1 at 0x555555555254: file armstrong.c, line 25.
(gdb) break 35
Breakpoint 2 at 0x555555555297: file armstrong.c, line 36.
(gdb) run
Starting program: /home/student/422128/a.out
True

Breakpoint 2, main () at armstrong.c:36
36              x = 1253;
(gdb) print sum
No symbol "sum" in current context.
(gdb) print x
$1 = 153
(gdb) next
37              if (isArmstrong(x) == 1)
(gdb) next

Breakpoint 1, isArmstrong (x=1253) at armstrong.c:25
```



```
(gdb) next
37              if (isArmstrong(x) == 1)
(gdb) next

Breakpoint 1, isArmstrong (x=1253) at armstrong.c:25
25                      return 0;
(gdb) next
26      }
(gdb) next
main () at armstrong.c:37
37              if (isArmstrong(x) == 1)
(gdb) next
40                      printf("False\n");
(gdb) next
False
42              return 0;
(gdb) next
43      }
(gdb) next
__libc_start_call_main (main=main@entry=0x55555555525b <main>, argc=argc@entry=1, argv=argv@entry=0x7fffffffdc9
    at ../sysdeps/nptl/libc_start_call_main.h:74
74      ../sysdeps/nptl/libc_start_call_main.h: No such file or directory.
(gdb) continue
Continuing.
[Inferior 1 (process 16193) exited normally]
```



```
(gdb) next
__libc_start_call_main (main=main@entry=0x55555555525b <main>, argc=argc@entry=1, argv=argv@entry=0x7fffffffdc9
    at ../sysdeps/nptl/libc_start_call_main.h:74
74      ../sysdeps/nptl/libc_start_call_main.h: No such file or directory.
(gdb) continue
Continuing.
[Inferior 1 (process 16193) exited normally]
(gdb) disassemble main
Dump of assembler code for function main:
   0x000055555555525b <+0>:     endbr64
   0x000055555555525f <+4>:     push   %rbp
   0x0000555555555260 <+5>:     mov    %rsp,%rbp
   0x0000555555555263 <+8>:     sub    $0x10,%rsp
   0x0000555555555267 <+12>:    movl   $0x99,-0x4(%rbp)
   0x000055555555526e <+19>:    mov    -0x4(%rbp),%eax
   0x0000555555555271 <+22>:    mov    %eax,%edi
   0x0000555555555273 <+24>:    callq  0x5555555551aa <isArmstrong>
   0x0000555555555278 <+29>:    cmp    $0x1,%eax
   0x000055555555527b <+32>:    jne    0x55555555528b <main+48>
   0x000055555555527d <+34>:    lea    0xd80(%rip),%rdi        # 0x555555556004
   0x0000555555555284 <+41>:    callq  0x555555555060 <puts@plt>
   0x0000555555555289 <+46>:    jmp    0x555555555297 <main+60>
   0x000055555555528b <+48>:    lea    0xd77(%rip),%rdi        # 0x555555556009
   0x0000555555555292 <+55>:    callq  0x555555555060 <puts@plt>
   0x0000555555555297 <+60>:    movl   $0x4e5,-0x4(%rbp)
   0x000055555555529e <+67>:    mov    -0x4(%rbp),%eax
   0x00005555555552a1 <+70>:    mov    %eax,%edi
   0x00005555555552a3 <+72>:    callq  0x5555555551aa <isArmstrong>
   0x00005555555552a8 <+77>:    cmp    $0x1,%eax
   0x00005555555552ab <+80>:    jne    0x5555555552bb <main+96>
   0x00005555555552ad <+82>:    lea    0xd50(%rip),%rdi        # 0x555555556004
   0x00005555555552b4 <+89>:    callq  0x555555555060 <puts@plt>
   0x00005555555552b9 <+94>:    jmp    0x5555555552c7 <main+108>
   0x00005555555552bb <+96>:    lea    0xd47(%rip),%rdi        # 0x555555556009
   0x00005555555552c2 <+103>:   callq  0x555555555060 <puts@plt>
   0x00005555555552c7 <+108>:   mov    $0x0,%eax
   0x00005555555552cc <+113>:   leaveq
   0x00005555555552cd <+114>:   retq
End of assembler dump.
(gdb) quit
student@ai-HP-ProDesk-600-G4-MT:~/422128$
student@ai-HP-ProDesk-600-G4-MT:~/422128$ 
```