NAME : J.RAMYA

ROLL NO. : 422160

# ASSIGNMENT 04

Generate different C programs that induce a segmentation fault error, select these examples of your choice, and employ the GDB utility for debugging on Linux.

Note:

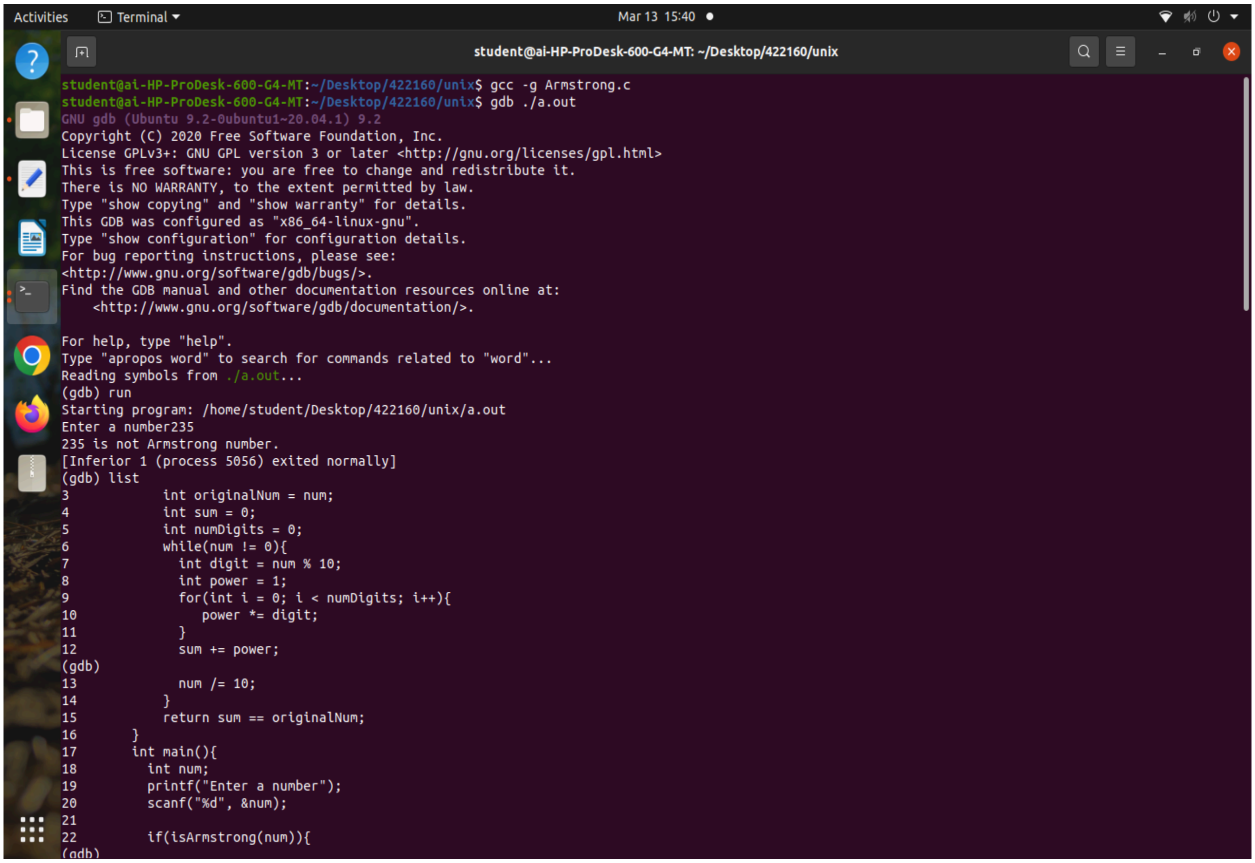
1. Include multiple breakpoints while debugging

2. Upload your submission in a format consistent with the example provided in the material.

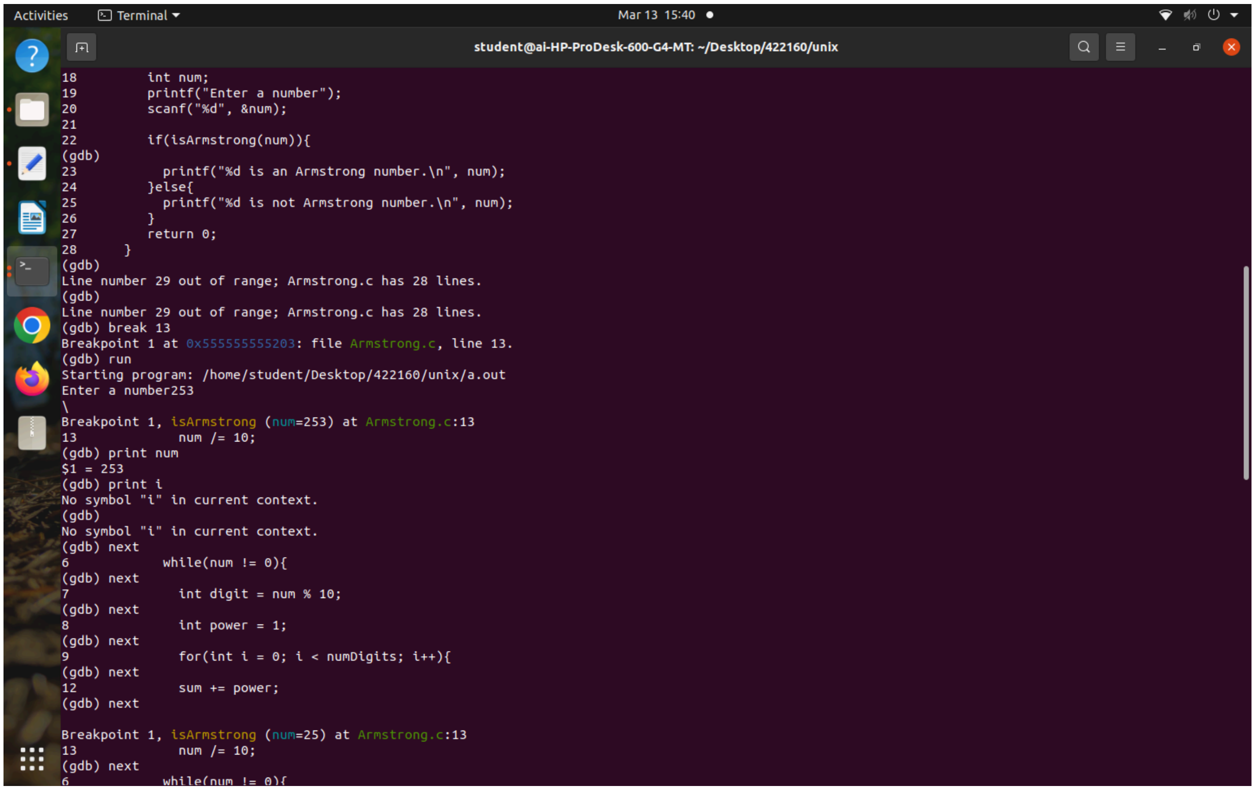
## CODE:

```
#include <stdio.h>
int isArmstrong(int num) {
    int originalNum = num;
    int sum = 0;
    int numDigits = 0;
    while(num != 0){
      int digit = num % 10;
      int power = 1;
      for(int i = 0; i < numDigits; i++){
         power *= digit;
      }
      sum += power;
      num /= 10;
    }
    return sum == originalNum;
 }
 int main(){
   int num;
   printf("Enter a number");
   scanf("%d", &num);

   if(isArmstrong(num)){
     printf("%d is an Armstrong number.\n", num);
   }else{
     printf("%d is not Armstrong number.\n", num);
   }
   return 0;
}
```

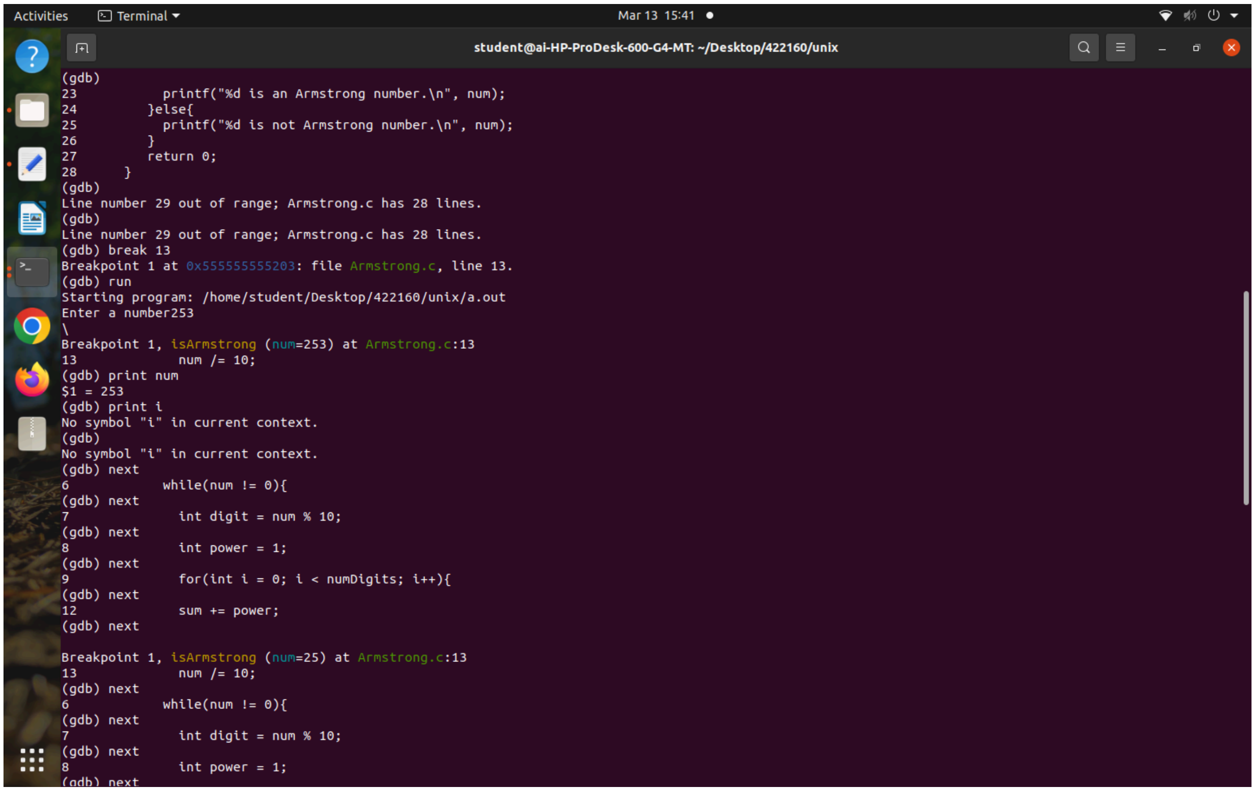
# OUTPUT:



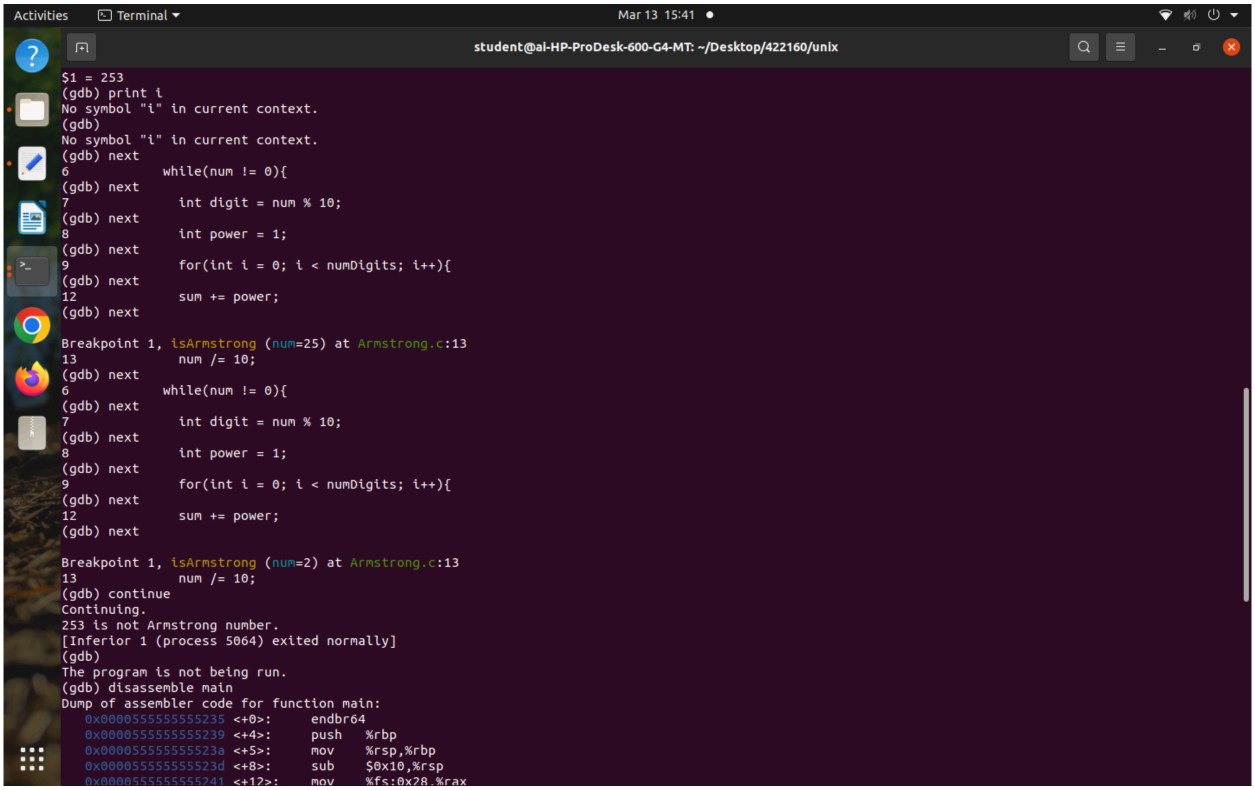
```
student@ai-HP-ProDesk-600-G4-MT:~/Desktop/422160/unix$ gcc -g Armstrong.c
student@ai-HP-ProDesk-600-G4-MT:~/Desktop/422160/unix$ gdb ./a.out
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04.1) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
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/422160/unix/a.out
Enter a number235
235 is not Armstrong number.
[Inferior 1 (process 5056) exited normally]
(gdb) list
            int originalNum = num;
            int sum = 0;
            int numDigits = 0;
            while(num != 0){
              int digit = num % 10;
              int power = 1;
              for(int i = 0; i < numDigits; i++){
                 power *= digit;
              }
              sum += power;
(gdb)
              num /= 10;
            }
            return sum == originalNum;
        }
        int main(){
          int num;
          printf("Enter a number");
          scanf("%d", &num);

          if(isArmstrong(num)){
(gdb)
```



```
18          int num;
19          printf("Enter a number");
20          scanf("%d", &num);
21
22          if(isArmstrong(num)){
(gdb)
23            printf("%d is an Armstrong number.\n", num);
24          }else{
25            printf("%d is not Armstrong number.\n", num);
26          }
27          return 0;
28      }
(gdb)
Line number 29 out of range; Armstrong.c has 28 lines.
(gdb)
Line number 29 out of range; Armstrong.c has 28 lines.
(gdb) break 13
Breakpoint 1 at 0x555555555203: file Armstrong.c, line 13.
(gdb) run
Starting program: /home/student/Desktop/422160/unix/a.out
Enter a number253
\
Breakpoint 1, isArmstrong (num=253) at Armstrong.c:13
13              num /= 10;
(gdb) print num
$1 = 253
(gdb) print i
No symbol "i" in current context.
(gdb)
No symbol "i" in current context.
(gdb) next
6             while(num != 0){
(gdb) next
7               int digit = num % 10;
(gdb) next
8               int power = 1;
(gdb) next
9               for(int i = 0; i < numDigits; i++){
(gdb) next
12              sum += power;
(gdb) next

Breakpoint 1, isArmstrong (num=25) at Armstrong.c:13
13              num /= 10;
(gdb) next
6             while(num != 0){
```



```
(gdb)
23            printf("%d is an Armstrong number.\n", num);
24          }else{
25            printf("%d is not Armstrong number.\n", num);
26          }
27          return 0;
28      }
(gdb)
Line number 29 out of range; Armstrong.c has 28 lines.
(gdb)
Line number 29 out of range; Armstrong.c has 28 lines.
(gdb) break 13
Breakpoint 1 at 0x555555555203: file Armstrong.c, line 13.
(gdb) run
Starting program: /home/student/Desktop/422160/unix/a.out
Enter a number253
\
Breakpoint 1, isArmstrong (num=253) at Armstrong.c:13
13              num /= 10;
(gdb) print num
$1 = 253
(gdb) print i
No symbol "i" in current context.
(gdb)
No symbol "i" in current context.
(gdb) next
6             while(num != 0){
(gdb) next
7               int digit = num % 10;
(gdb) next
8               int power = 1;
(gdb) next
9               for(int i = 0; i < numDigits; i++){
(gdb) next
12              sum += power;
(gdb) next

Breakpoint 1, isArmstrong (num=25) at Armstrong.c:13
13              num /= 10;
(gdb) next
6             while(num != 0){
(gdb) next
7               int digit = num % 10;
(gdb) next
8               int power = 1;
(gdb) next
```



```
$1 = 253
(gdb) print i
No symbol "i" in current context.
(gdb)
No symbol "i" in current context.
(gdb) next
6               while(num != 0){
(gdb) next
7                 int digit = num % 10;
(gdb) next
8                 int power = 1;
(gdb) next
9                 for(int i = 0; i < numDigits; i++){
(gdb) next
12                sum += power;
(gdb) next

Breakpoint 1, isArmstrong (num=25) at Armstrong.c:13
13                num /= 10;
(gdb) next
6               while(num != 0){
(gdb) next
7                 int digit = num % 10;
(gdb) next
8                 int power = 1;
(gdb) next
9                 for(int i = 0; i < numDigits; i++){
(gdb) next
12                sum += power;
(gdb) next

Breakpoint 1, isArmstrong (num=2) at Armstrong.c:13
13                num /= 10;
(gdb) continue
Continuing.
253 is not Armstrong number.
[Inferior 1 (process 5064) exited normally]
(gdb)
The program is not being run.
(gdb) disassemble main
Dump of assembler code for function main:
   0x0000555555555235 <+0>:     endbr64
   0x0000555555555239 <+4>:     push   %rbp
   0x000055555555523a <+5>:     mov    %rsp,%rbp
   0x000055555555523d <+8>:     sub    $0x10,%rsp
   0x0000555555555241 <+12>:    mov    %fs:0x28,%rax
```