ASSIGNMENT - 4

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.

PROG - 1

```
student@ai-HP-ProDesk-600-G4-MT:~/Desktop/422133$ gcc prog.c
student@ai-HP-ProDesk-600-G4-MT:~/Desktop/422133$ ./a.out
Enter a number: 85742361
Segmentation fault (core dumped)
student@ai-HP-ProDesk-600-G4-MT:~/Desktop/422133$ gcc -g prog.c
student@ai-HP-ProDesk-600-G4-MT:~/Desktop/422133$ 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       int main() {
4           char num[10];
5           printf("Enter a number: ");
6           scanf("%s", num);
7           int sum = 0;
8           for (int i = 0; num[i] != '\0'; i++) {
9               sum += num[i] - '0';
10          }
(gdb) list
11
12          int *ptr = NULL;
13          *ptr = sum;
14          printf("Sum of digits: %d\n", sum);
15          return 0;
16      }
(gdb) list
Line number 17 out of range; prog.c has 16 lines.
```

```
(gdb) list
Line number 17 out of range; prog.c has 16 lines.
(gdb) break 3
Breakpoint 1 at 0x1189: file prog.c, line 3.
(gdb) run
Starting program: /home/student/Desktop/422133/a.out

Breakpoint 1, main () at prog.c:3
3       int main() {
(gdb) print i
No symbol "i" in current context.
(gdb) next
5           printf("Enter a number: ");
(gdb) next
6           scanf("%s", num);
(gdb) next
Enter a number: 714
7           int sum = 0;
(gdb) print num
$1 = "714\000\377\377\377\177\000"
(gdb) print sum
$2 = 1431654992
(gdb) next
8           for (int i = 0; num[i] != '\0'; i++) {
(gdb) print i
$3 = 21845
(gdb) print sum
$4 = 0
(gdb) print num
$5 = "714\000\377\377\377\177\000"
(gdb) next
9               sum += num[i] - '0';
(gdb) next
8           for (int i = 0; num[i] != '\0'; i++) {
(gdb) next
9               sum += num[i] - '0';
(gdb) next
8           for (int i = 0; num[i] != '\0'; i++) {
(gdb) next
9               sum += num[i] - '0';
(gdb) continue
Continuing.
```

```
(gdb) continue
Continuing.

Program received signal SIGSEGV, Segmentation fault.
0x0000555555555211 in main () at prog.c:13
13          *ptr = sum;
(gdb) continue
Continuing.

Program terminated with signal SIGSEGV, Segmentation fault.
The program no longer exists.
(gdb)
```

PROG - 2

```
student@ai-HP-ProDesk-600-G4-MT:~$ cd Desktop
student@ai-HP-ProDesk-600-G4-MT:~/Desktop$ cd 422133
student@ai-HP-ProDesk-600-G4-MT:~/Desktop/422133$ gcc prog2.c
student@ai-HP-ProDesk-600-G4-MT:~/Desktop/422133$ ./a.out
10
1
0
free(): double free detected in tcache 2
Aborted (core dumped)
student@ai-HP-ProDesk-600-G4-MT:~/Desktop/422133$ gcc -g prog2.c
student@ai-HP-ProDesk-600-G4-MT:~/Desktop/422133$ 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
11          new_node->data = data;
12          new_node->next = NULL;
13
14          if (*head_ref == NULL) {
15              *head_ref = new_node;
16          } else {
17              struct Node* current = *head_ref;
18              while (current->next != NULL) {
19                  current = current->next;
20              }
(gdb) list
21              current->next = new_node;
22          }
23      }
24
25      int main() {
26          struct Node* head = NULL;
27          insertNode(&head, 10);
```

```
25      int main() {
26          struct Node* head = NULL;
27          insertNode(&head, 10);
28          insertNode(&head, 20);
29
30          struct Node* current = head;
(gdb) list
31          while (current != NULL) {
32              printf("%d\n", current->data);
33              free(current);
34              current = current->next;
35          }
36
37          return 0;
38      }
(gdb) list
Line number 39 out of range; prog2.c has 38 lines.
(gdb) break 8
Breakpoint 1 at 0x11a9: file prog2.c, line 9.
(gdb) run
Starting program: /home/student/Desktop/422133/a.out

Breakpoint 1, insertNode (head_ref=0x7fffffffde97, data=0) at prog2.c:9
9       void insertNode(struct Node** head_ref, int data) {
(gdb) nex
Ambiguous command "nex": next, nexti.
(gdb) next
10          struct Node* new_node = (struct Node*)malloc(sizeof(struct Node));
(gdb) next
11          new_node->data = data;
(gdb) next
12          new_node->next = NULL;
(gdb) next
14          if (*head_ref == NULL) {
(gdb) next
15              *head_ref = new_node;
(gdb) next
23      }
(gdb) print new_node
$1 = (struct Node *) 0x5555555592a0
(gdb) next
main () at prog2.c:28
28          insertNode(&head, 20);
```

```
28          insertNode(&head, 20);
(gdb) next

Breakpoint 1, insertNode (head_ref=0x7fffffffde98, data=10) at prog2.c:9
9       void insertNode(struct Node** head_ref, int data) {
(gdb) print head
No symbol "head" in current context.
(gdb) print &Node
No symbol "Node" in current context.
(gdb) print &head
No symbol "head" in current context.
(gdb) next
10          struct Node* new_node = (struct Node*)malloc(sizeof(struct Node));
(gdb) next
11          new_node->data = data;
(gdb) next
12          new_node->next = NULL;
(gdb) next
14          if (*head_ref == NULL) {
(gdb) next
17              struct Node* current = *head_ref;
(gdb) next
18              while (current->next != NULL) {
(gdb) next
21              current->next = new_node;
(gdb) next
23      }
(gdb) next
main () at prog2.c:30
30          struct Node* current = head;
(gdb) next
31          while (current != NULL) {
(gdb) nex
Ambiguous command "nex": next, nexti.
(gdb) next
32              printf("%d\n", current->data);
(gdb) next
10
33              free(current);
(gdb) next
34              current = current->next;
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) n
Program not restarted.
(gdb) next
```

```
34              current = current->next;
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) n
Program not restarted.
(gdb) next
31          while (current != NULL) {
(gdb) next
32              printf("%d\n", current->data);
(gdb) print data
$2 = (struct here_cg_arc_record *) 0x0
(gdb) print current
$3 = (struct Node *) 0x555555559010
(gdb) next
1
33              free(current);
(gdb) next data
main () at prog2.c:33
33              free(current);
(gdb) print current
$4 = (struct Node *) 0x555555559010
(gdb) next
34              current = current->next;
(gdb) print current
$5 = (struct Node *) 0x555555559010
(gdb) print data
$6 = (struct here_cg_arc_record *) 0x0
(gdb) next
31          while (current != NULL) {
(gdb) next
32              printf("%d\n", current->data);
(gdb) continue
Continuing.
0
free(): double free detected in tcache 2

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)
```

```
student@ai-HP-ProDesk-600-G4-MT:~$ cd Desktop
student@ai-HP-ProDesk-600-G4-MT:~/Desktop$ gcc -g fact_gdb.c
student@ai-HP-ProDesk-600-G4-MT:~/Desktop$ 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/a.out
Enter a number :5
Factorial of 5 = 1
Factorial of 5 = 2
Factorial of 5 = 6
Factorial of 5 = 24
Factorial of 5 = 120
[Inferior 1 (process 14613) exited normally]
(gdb) list
1       #include<stdio.h>
2       int main(){
3               int num;
4               long long factorial=1;
5               printf("Enter a number :");
6               scanf("%d",&num);
7               if (num<0){
8                       printf("Error! Factorial of a negative number doesnt 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) break 11
Breakpoint 1 at 0x5555555551f6: file fact_gdb.c, line 11.
(gdb) run
Starting program: /home/student/Desktop/a.out
Enter a number :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) next
12                              factorial*=i;
(gdb) print num
$5 = 5
(gdb) next
13                              i++;
(gdb) next
14                              printf("Factorial of %d = %lld\n",num,factorial);
(gdb) next
Factorial of 5 = 2
11                      while(i<=num){
(gdb) next
12                              factorial*=i;
(gdb) next
13                              i++;
(gdb) next
```

```
(gdb) next
14                              printf("Factorial of %d = %lld\n",num,factorial);
(gdb) continue
Continuing.
Factorial of 5 = 6
Factorial of 5 = 24
Factorial of 5 = 120
[Inferior 1 (process 14624) 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    0xe4e(%rip),%rdi        # 0x555555556019
   0x00005555555551cb <+66>:    mov    $0x0,%eax
   0x00005555555551d0 <+71>:    callq  0x555555555090 <__isoc99_scanf@plt>
   0x00005555555551d5 <+76>:    mov    -0x18(%rbp),%eax
   0x00005555555551d8 <+79>:    test   %eax,%eax
   0x00005555555551da <+81>:    jns    0x5555555551ef <main+102>
   0x00005555555551dc <+83>:    lea    0xe3d(%rip),%rdi        # 0x555555556020
   0x00005555555551e3 <+90>:    mov    $0x0,%eax
   0x00005555555551e8 <+95>:    callq  0x555555555080 <printf@plt>
--Type <RET> for more, q to quit, c to continue without paging--c
   0x00005555555551ed <+100>:   jmp    0x55555555522f <main+166>
   0x00005555555551ef <+102>:   movl   $0x1,-0x14(%rbp)
   0x00005555555551f6 <+109>:   jmp    0x555555555227 <main+158>
   0x00005555555551f8 <+111>:   mov    -0x14(%rbp),%eax
   0x00005555555551fb <+114>:   cltq
   0x00005555555551fd <+116>:   mov    -0x10(%rbp),%rdx
   0x0000555555555201 <+120>:   imul   %rdx,%rax
   0x0000555555555205 <+124>:   mov    %rax,-0x10(%rbp)
   0x0000555555555209 <+128>:   addl   $0x1,-0x14(%rbp)
   0x000055555555520d <+132>:   mov    -0x18(%rbp),%eax
   0x0000555555555210 <+135>:   mov    -0x10(%rbp),%rdx
   0x0000555555555214 <+139>:   mov    %eax,%esi
   0x0000555555555216 <+141>:   lea    0xe36(%rip),%rdi        # 0x555555556053
```

```
   0x0000555555555216 <+141>:   lea    0xe36(%rip),%rdi        # 0x555555556053
   0x000055555555521d <+148>:   mov    $0x0,%eax
   0x0000555555555222 <+153>:   callq  0x555555555080 <printf@plt>
   0x0000555555555227 <+158>:   mov    -0x18(%rbp),%eax
   0x000055555555522a <+161>:   cmp    %eax,-0x14(%rbp)
   0x000055555555522d <+164>:   jle    0x5555555551f8 <main+111>
   0x000055555555522f <+166>:   mov    $0x0,%eax
   0x0000555555555234 <+171>:   mov    -0x8(%rbp),%rcx
   0x0000555555555238 <+175>:   xor    %fs:0x28,%rcx
   0x0000555555555241 <+184>:   je     0x555555555248 <main+191>
   0x0000555555555243 <+186>:   callq  0x555555555070 <__stack_chk_fail@plt>
   0x0000555555555248 <+191>:   leaveq
   0x0000555555555249 <+192>:   retq
End of assembler dump.
(gdb) quit
student@ai-HP-ProDesk-600-G4-MT:~/Desktop$
```

NAME : CH V CHARAN

ROLL NO : 422133

SECTION : A