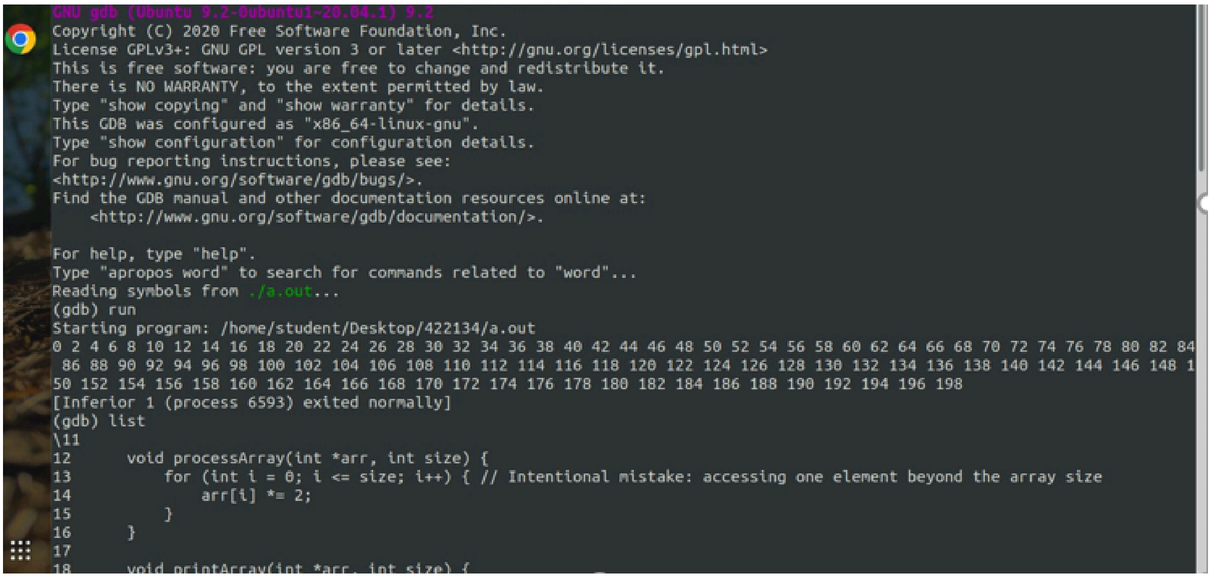
UNIX LAB ASSIGNMENT-WEEK 04

NAME:G.V.Sujith Royal

ROLL NO:422147

1)fact.c

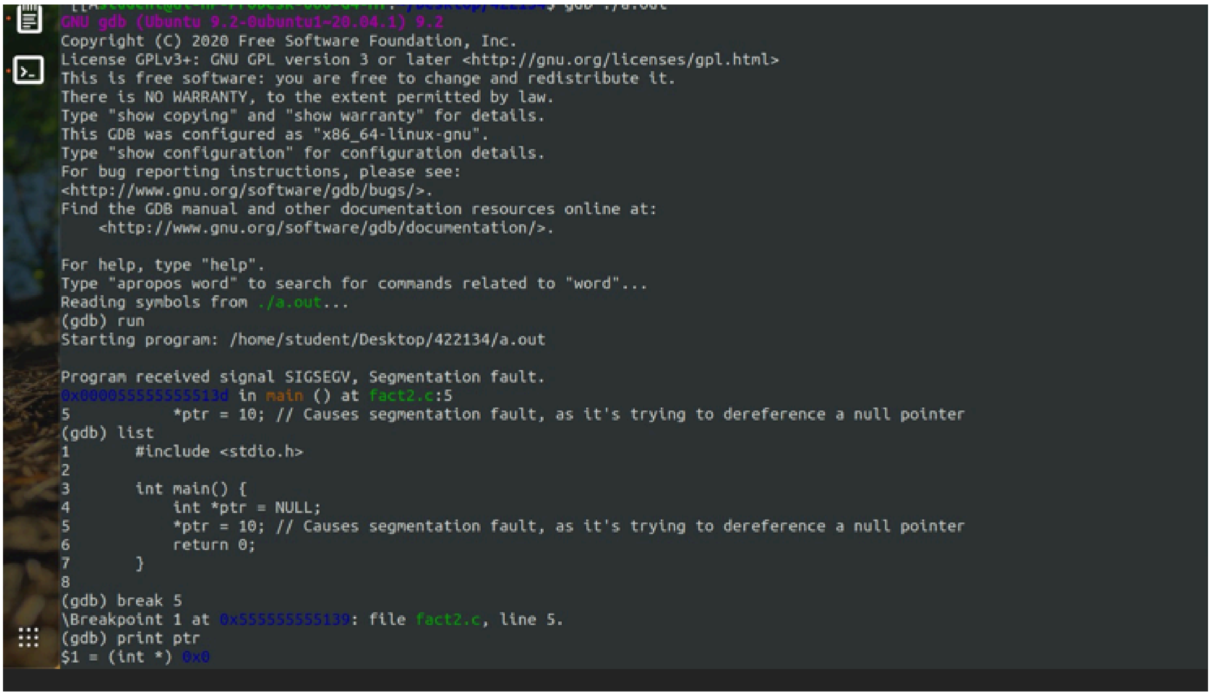
```
#include <stdio.h>
#include <stdlib.h>

#define ARRAY_SIZE 100

void initializeArray(int *arr, int size) {
    for (int i = 0; i < size; i++) {
        arr[i] = i;
    }
}

void processArray(int *arr, int size) {
    for (int i = 0; i <= size; i++) { // Intentional mistake: accessing one element beyond the array size
        arr[i] *= 2;
    }
}

void printArray(int *arr, int size) {
    for (int i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

int main() {
    int *arr = (int *)malloc(ARRAY_SIZE * sizeof(int));
    if (arr == NULL) {
        printf("Memory allocation failed!\n");
        return 1;
    }

    initializeArray(arr, ARRAY_SIZE);
    processArray(arr, ARRAY_SIZE);
    printArray(arr, ARRAY_SIZE);
    free(arr);
    return 0;
}
```



```
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/422134/a.out
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66 68 70 72 74 76 78 80 82 84
 86 88 90 92 94 96 98 100 102 104 106 108 110 112 114 116 118 120 122 124 126 128 130 132 134 136 138 140 142 144 146 148 1
50 152 154 156 158 160 162 164 166 168 170 172 174 176 178 180 182 184 186 188 190 192 194 196 198
[Inferior 1 (process 6593) exited normally]
(gdb) list
\11
12      void processArray(int *arr, int size) {
13          for (int i = 0; i <= size; i++) { // Intentional mistake: accessing one element beyond the array size
14              arr[i] *= 2;
15          }
16      }
17
18      void printArray(int *arr, int size) {
```

Q2)fact2.c

```
#include <stdio.h>

int main() {
    int *ptr = NULL;
    *ptr = 10; // Causes segmentation fault, as it's trying to dereference a null pointer
    return 0;
}
```

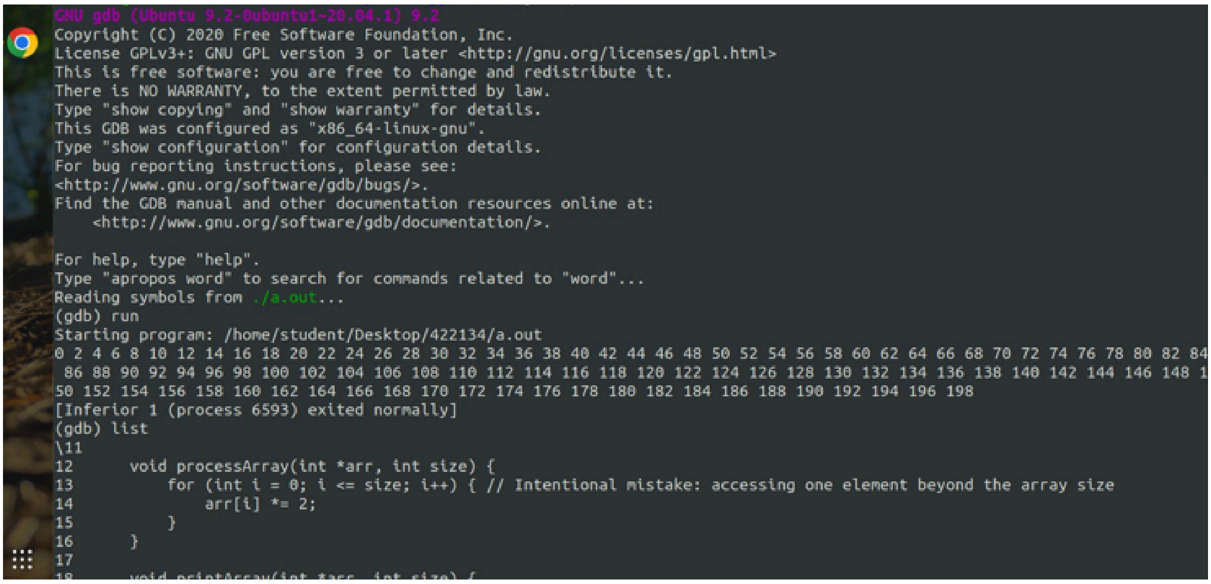


```
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04.1) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show 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/422134/a.out

Program received signal SIGSEGV, Segmentation fault.
0x000055555555513d in main () at fact2.c:5
5           *ptr = 10; // Causes segmentation fault, as it's trying to dereference a null pointer
(gdb) list
1       #include <stdio.h>
2
3       int main() {
4           int *ptr = NULL;
5           *ptr = 10; // Causes segmentation fault, as it's trying to dereference a null pointer
6           return 0;
7       }
8
(gdb) break 5
\Breakpoint 1 at 0x555555555139: file fact2.c, line 5.
(gdb) print ptr
$1 = (int *) 0x0
```

3)fact3.c

```
#include <stdio.h>


int main() {
    int *ptr = NULL;


    // First segmentation fault: Dereferencing a null
pointer
    *ptr = 10;
```

```c
    // Second segmentation fault: Accessing memory beyond the allocated space
    int arr[5];
    arr[10] = 20;

    // Third segmentation fault: Writing to a read-only memory
    char *str = "Segmentation fault";
    str[0] = 's';

    // Fourth segmentation fault: Stack overflow
    main();

    return 0;
}
```



```
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/422134/a.out
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66 68 70 72 74 76 78 80 82 84
 86 88 90 92 94 96 98 100 102 104 106 108 110 112 114 116 118 120 122 124 126 128 130 132 134 136 138 140 142 144 146 148 1
50 152 154 156 158 160 162 164 166 168 170 172 174 176 178 180 182 184 186 188 190 192 194 196 198
[Inferior 1 (process 6593) exited normally]
(gdb) list
\11
12      void processArray(int *arr, int size) {
13          for (int i = 0; i <= size; i++) { // Intentional mistake: accessing one element beyond the array size
14              arr[i] *= 2;
15          }
16      }
17
18      void printArray(int *arr, int size) {
```