

## Operating Systems Assignment #1 - Submission

<u>Question 1</u>	2
<u>GDB - Textual debugger</u>	3
<u>1. Stack overflow</u>	3
• Without debug info	3
• With debug info	4
<u>2. Division by zero - Floating point exception</u>	5
• Without debug info	5
• With debug info	6
<u>3. Out of memory</u>	7
• Without debug info	7
• With debug info	8
<u>ddd - Graphical debugger</u>	9
<u>1. Stack overflow</u>	9
• Without debug info	9
• With debug info	10
<u>2. Division by zero - Floating point exception</u>	11
• Without debug info	11
• With debug info	12
<u>3. Out of memory</u>	13
• Without debug info	13
• With debug info	14
<u>Question 2</u>	15
<u>Question 3</u>	16
<u>Question 4</u>	19
<u>Question 5</u>	23
<u>Question 6</u>	26

## Question 1

We turned off the limit for the core dump file size using 'ulimit -c unlimited'

```
 sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/question-1$ ulimit -a
real-time non-blocking time (microseconds, -R) unlimited
core file size          (blocks, -c) 0
data seg size            (kbytes, -d) unlimited
scheduling priority      (-e) 0
file size                (blocks, -f) unlimited
pending signals           (-i) 31431
max locked memory        (kbytes, -l) 1015388
max memory size          (kbytes, -m) unlimited
open files               (-n) 1024
pipe size                (512 bytes, -p) 8
POSIX message queues     (bytes, -q) 819200
real-time priority        (-r) 0
stack size               (kbytes, -s) 8192
cpu time                 (seconds, -t) unlimited
max user processes        (-u) 31431
virtual memory            (kbytes, -v) unlimited
file locks               (-x) unlimited
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/question-1$ ulimit -c unlimited
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/question-1$ ulimit -a
real-time non-blocking time (microseconds, -R) unlimited
core file size          (blocks, -c) unlimited
data seg size            (kbytes, -d) unlimited
scheduling priority      (-e) 0
file size                (blocks, -f) unlimited
pending signals           (-i) 31431
max locked memory        (kbytes, -l) 1015388
max memory size          (kbytes, -m) unlimited
open files               (-n) 1024
pipe size                (512 bytes, -p) 8
POSIX message queues     (bytes, -q) 819200
real-time priority        (-r) 0
stack size               (kbytes, -s) 8192
cpu time                 (seconds, -t) unlimited
max user processes        (-u) 31431
virtual memory            (kbytes, -v) unlimited
file locks               (-x) unlimited
```

Changed the default directory for core file to be the current directory using:  
'echo "kernel.core\_pattern=core" | sudo tee -a /etc/sysctl.conf && sudo sysctl -p'  
(Thanks to GPT)

The answer is organized in the next pages, a page per scenserio.

IDs: 53036281, 211791041

### GDB - Textual debugger

#### 1. Stack overflow

- Without debug info: core.6066

```
 sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/question-1$ make stack_overflow
gcc -Wall -c stack_overflow.c -o stack_overflow.o
gcc -Wall -o stack_overflow stack_overflow.o
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/question-1$ ./stack_overflow
AGHAFASHFASH OVERFLOW INCOMING!!!
Segmentation fault (core dumped)
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/question-1$ ls -lart
total 8360
drwxrwxr-x 4 sam sam    4096 May  3 13:21 ...
-rw-rw-r-- 1 sam sam     302 May  3 13:53 division_by_zero.c
-rw-rw-r-- 1 sam sam    168 May  3 13:53 out_of_memory.c
-rw-rw-r-- 1 sam sam    278 May  5 11:51 stack_overflow.c
drwxrwxr-x 2 sam sam    4096 May  5 12:41 core_files
-rw-rw-r-- 1 sam sam    370 May  5 12:41 makefile
-rw-rw-r-- 1 sam sam   1832 May  5 12:43 stack_overflow.o
-rwxrwxr-x 1 sam sam  16048 May  5 12:43 stack_overflow
-rw----- 1 sam sam 8695808 May  5 12:43 core.6066
drwxrwxr-x 3 sam sam    4096 May  5 12:44 .
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/question-1$ gdb ./stack_overflow -c core.6066
GNU gdb (Ubuntu 12.1-0ubuntu1~22.04) 12.1
Copyright (C) 2022 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:
<https://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 ./stack_overflow...
(No debugging symbols found in ./stack_overflow)
[New LWP 6066]
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Core was generated by `./stack_overflow'.
Program terminated with signal SIGSEGV, Segmentation fault.
#0 0x00005c8eed8d01c9 in stack_overflow ()
(gdb) where
#0 0x00005c8eed8d01c9 in stack_overflow ()
#1 0x00005c8eed8d01ce in stack_overflow ()
#2 0x00005c8eed8d01ce in stack_overflow ()
#3 0x00005c8eed8d01ce in stack_overflow ()
#4 0x00005c8eed8d01ce in stack_overflow ()
#5 0x00005c8eed8d01ce in stack_overflow ()
#6 0x00005c8eed8d01ce in stack_overflow ()
#7 0x00005c8eed8d01ce in stack_overflow ()
#8 0x00005c8eed8d01ce in stack_overflow ()
#9 0x00005c8eed8d01ce in stack_overflow ()
#10 0x00005c8eed8d01ce in stack_overflow ()
#11 0x00005c8eed8d01ce in stack_overflow ()
#12 0x00005c8eed8d01ce in stack_overflow ()
```

The core file doesn't show exactly where the error happened (specifically to the line), but we see the failure happened in the 'stack\_overflow' function.

Using the '**where**' command shows the stack trace, which at the end (after 5233838 recursive calls) shows our function was called from 'main'.

```
#523837 0x00005c8eed8d01ce in stack_overflow ()
#523838 0x00005c8eed8d019c in main ()
(gdb) 
```

IDs: 53036281, 211791041

## 1. Stack overflow

- With debug info (-g flag in compilation): core.4530

```
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/question-1$ make stack_overflow
gcc -Wall -g -c stack_overflow.c -o stack_overflow.o
gcc -Wall -g -o stack_overflow stack_overflow.o
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/question-1$ ./stack_overflow
AGHAFASHFASH OVERFLOW INCOMING!!!
Segmentation fault (core dumped)
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/question-1$ ls
core.4530  division_by_zero.c  makefile  out_of_memory.c  stack_overflow  stack_overflow.c  stack_overflow.o
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/question-1$ gdb ./stack_overflow -c core.4530
GNU gdb (Ubuntu 12.1-0ubuntu1~22.04) 12.1
Copyright (C) 2022 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:
<https://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 ./stack_overflow...
[New LWP 4530]
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Core was generated by `./stack_overflow'.
Program terminated with signal SIGSEGV, Segmentation fault.
#0 0x00005c41b490b1c9 in stack_overflow () at stack_overflow.c:15
15      stack_overflow();
(gdb) where
#0 0x00005c41b490b1c9 in stack_overflow () at stack_overflow.c:15
#1 0x00005c41b490b1ce in stack_overflow () at stack_overflow.c:15
#2 0x00005c41b490b1ce in stack_overflow () at stack_overflow.c:15
#3 0x00005c41b490b1ce in stack_overflow () at stack_overflow.c:15
#4 0x00005c41b490b1ce in stack_overflow () at stack_overflow.c:15
#5 0x00005c41b490b1ce in stack_overflow () at stack_overflow.c:15
#6 0x00005c41b490b1ce in stack_overflow () at stack_overflow.c:15
#7 0x00005c41b490b1ce in stack_overflow () at stack_overflow.c:15
#8 0x00005c41b490b1ce in stack_overflow () at stack_overflow.c:15
#9 0x00005c41b490b1ce in stack_overflow () at stack_overflow.c:15
#10 0x00005c41b490b1ce in stack_overflow () at stack_overflow.c:15
#11 0x00005c41b490b1ce in stack_overflow () at stack_overflow.c:15
#12 0x00005c41b490b1ce in stack_overflow () at stack_overflow.c:15
#13 0x00005c41b490b1ce in stack_overflow () at stack_overflow.c:15
#14 0x00005c41b490b1ce in stack_overflow () at stack_overflow.c:15
#15 0x00005c41b490b1ce in stack_overflow () at stack_overflow.c:15
#16 0x00005c41b490b1ce in stack_overflow () at stack_overflow.c:15
#17 0x00005c41b490b1ce in stack_overflow () at stack_overflow.c:15
#18 0x00005c41b490b1ce in stack_overflow () at stack_overflow.c:15
#19 0x00005c41b490b1ce in stack_overflow () at stack_overflow.c:15
#20 0x00005c41b490b1ce in stack_overflow () at stack_overflow.c:15
#21 0x00005c41b490b1ce in stack_overflow () at stack_overflow.c:15
#22 0x00005c41b490b1ce in stack_overflow () at stack_overflow.c:15
```

We can see that line 15 in the 'stack\_overflow' function is causing the stack overflow from #0.

Using the 'where' command also shows there is a stack overflow problem with a lot of calls for the problematic function. (we see the stack trace is full - starting from a call in the 'main' function in line 9).

The screenshot above shows the first page only, this goes on until #523527.

```
#523526 0x00005c41b490b1ce in stack_overflow () at stack_overflow.c:15
#523527 0x00005c41b490b1ce in main (argc=1, argv=0x7ffe2f4e81a8) at stack_overflow.c:9
(gdb) 
```

IDs: 53036281, 211791041

## 2. Division by zero - Floating point exception

- Without debug info: core.6319

```
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/question-1$ make division_by_zero
gcc -Wall -c division_by_zero.c -o division_by_zero.o
gcc -Wall -o division_by_zero division_by_zero.o
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/question-1$ ./division_by_zero
Floating point exception (core dumped)
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/question-1$ ls -lart
total 8504
drwxrwxr-x 4 sam sam 4096 May  3 13:21 .
-rw-rw-r-- 1 sam sam 302 May  3 13:53 division_by_zero.c
-rw-rw-r-- 1 sam sam 168 May  3 13:53 out_of_memory.c
-rw-rw-r-- 1 sam sam 278 May  5 11:51 stack_overflow.c
drwxrwxr-x 2 sam sam 4096 May  5 12:41 core_files
-rw-rw-r-- 1 sam sam 370 May  5 12:41 makefile
-rw-rw-r-- 1 sam sam 1832 May  5 12:43 stack_overflow.o
-rwxrwxr-x 1 sam sam 16048 May  5 12:43 stack_overflow
-rw----- 1 sam sam 8695808 May  5 12:43 core.6066
-rw-rw-r-- 1 sam sam 1536 May  5 12:51 division_by_zero.o
-rwxrwxr-x 1 sam sam 15976 May  5 12:51 division_by_zero
-rw----- 1 sam sam 307200 May  5 12:52 core.6319
drwxrwxr-x 3 sam sam 4096 May  5 12:52 .
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/question-1$ gdb ./division_by_zero -c core.6319
GNU gdb (Ubuntu 12.1-0ubuntu1~22.04) 12.1
Copyright (C) 2022 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:
<https://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 ./division_by_zero...
(No debugging symbols found in ./division_by_zero)
[New LWP 6319]
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Core was generated by './division_by_zero'.
Program terminated with signal SIGFPE, Arithmetic exception.
#0 0x0000640ee51f0169 in main ()
(gdb) where
#0 0x0000640ee51f0169 in main ()
```

All we know is that the exception happened in the 'main' function.

```
(gdb) where
#0 0x0000640ee51f0169 in main ()
(gdb) print num
$1 = num
(gdb) print zero
No symbol "zero" in current context.
(gdb) breakpoint 1
Undefined command: "breakpoint". Try "help".
(gdb) break 1
No line 1 in the current file.
Make breakpoint pending on future shared library load? (y or [n]) n
(gdb) run
Starting program: /home/sam/Desktop/git/Operation-Systems-Ex1/question-1/division_by_zero
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Program received signal SIGFPE, Arithmetic exception.
0x0000555555555169 in main ()
(gdb) break 6
No line 6 in the current file.
Make breakpoint pending on future shared library load? (y or [n]) n
(gdb) break 10
No line 10 in the current file.
Make breakpoint pending on future shared library load? (y or [n]) n
(gdb) break 3
No line 3 in the current file.
Make breakpoint pending on future shared library load? (y or [n]) n
(gdb) 
```

Because the -g flag is not included in compilation - we have very limited options when debugging it.

Using the '**where**' command shows the same thing - a problem in 'main'.

Trying to '**print**' the values of num and zero also result in nothing (we only know they are the problem because we caused it - otherwise, good luck finding the variables with the problems ;P).

And trying place 'breakpoint's doesn't work as well, so we can't step line by line.

IDs: 53036281, 211791041

## 2. Division by zero - Floating point exception

- With debug info (-g flag in compilation): core.5133

```
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/question-1$ make division_by_zero
gcc -Wall -g -c division_by_zero.c -o division_by_zero.o
gcc -Wall -g -o division_by_zero division_by_zero.o
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/question-1$ ./division_by_zero
Floating point exception (core dumped)
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/question-1$ ls -lart
total 8508
drwxrwxr-x 4 sam sam 4096 May 3 13:21 ..
-rw-rw-r-- 1 sam sam 302 May 3 13:53 division_by_zero.c
-rw-rw-r-- 1 sam sam 168 May 3 13:53 out_of_memory.c
-rw-rw-r-- 1 sam sam 373 May 3 14:14 makefile
-rw-rw-r-- 1 sam sam 278 May 5 11:51 stack_overflow.c
-rw-rw-r-- 1 sam sam 4460 May 5 11:59 stack_overflow.o
-rwxrwxr-x 1 sam sam 17448 May 5 11:59 stack_overflow
-rw-rw-r-- 1 sam sam 8695808 May 5 11:59 core.4530
-rw-rw-r-- 1 sam sam 4024 May 5 12:07 division_by_zero.o
-rwxrwxr-x 1 sam sam 17336 May 5 12:07 division_by_zero
-rw-rw-r-- 1 sam sam 307200 May 5 12:07 core.5133
drwxrwxr-x 2 sam sam 4096 May 5 12:07 .
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/question-1$ gdb ./division_by_zero -c core.5133
GNU gdb (Ubuntu 12.1-0ubuntu1+22.04) 12.1
Copyright (C) 2022 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:
<https://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 ./division_by_zero...
[New LWP 5133]
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Core was generated by `./division_by_zero'.
Program terminated with signal SIGFPE, Arithmetic exception.
#0 0x00005f0b3a976169 in main (argc=1, argv=0x7ffe0ac516d8) at division_by_zero.c:10
10      num = 5/zero;
(gdb) where
#0 0x00005f0b3a976169 in main (argc=1, argv=0x7ffe0ac516d8) at division_by_zero.c:10
(gdb) print num
$1 = 0
(gdb) print zero
$2 = 0
(gdb) break 10
Breakpoint 1 at 0x5f0b3a976169: file division_by_zero.c, line 10.
(gdb) run
Starting program: /home/sam/Desktop/git/Operation-Systems-Ex1/question-1/division_by_zero
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
```

We can see the program is terminated in main, line 10: num = 5/zero;

Using the ‘where’ command shows the same problem.

```
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, main (argc=1, argv=0xfffffffffe078) at division_by_zero.c:10
10      num = 5/zero;
(gdb) print zero
$3 = 0
(gdb) print num
$4 = 0
(gdb) continue
Continuing.

Program received signal SIGFPE, Arithmetic exception.
0x0000555555555569 in main (argc=1, argv=0xfffffffffe078) at division_by_zero.c:10
10      num = 5/zero;
(gdb) 
```

By setting a ‘breakpoint’ on this line, we can ‘run’ the program and when it’s paused on the breakpoint, ‘print’ the values at the moment of crashing: we see zero equal to 0 -> so we have a division by 0.

Then by typing ‘continue’, we see the crash happens right after the execution of the problematic line.

IDs: 53036281, 211791041

### 3. Out of memory

- Without debug info: core.6590

```
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/question-1$ make out_of_memory
gcc -Wall -c out_of_memory.c -o out_of_memory.o
gcc -Wall -o out_of_memory out_of_memory.o
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/question-1$ ./out_of_memory
Segmentation fault (core dumped)
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/question-1$ ls -lart
total 8648
drwxrwxr-x 4 sam sam    4096 May  3 13:21 ..
-rw-rw-r-- 1 sam sam     302 May  3 13:53 division_by_zero.c
-rw-rw-r-- 1 sam sam    168 May  3 13:53 out_of_memory.c
-rw-rw-r-- 1 sam sam    278 May  5 11:51 stack_overflow.c
drwxrwxr-x 2 sam sam    4096 May  5 12:41 core_files
-rw-rw-r-- 1 sam sam    370 May  5 12:41 makefile
-rw-rw-r-- 1 sam sam   1832 May  5 12:43 stack_overflow.o
-rwxrwxr-x 1 sam sam  16048 May  5 12:43 stack_overflow
-rw----- 1 sam sam 8695808 May  5 12:43 core.6066
-rw-rw-r-- 1 sam sam   1536 May  5 12:51 division_by_zero.o
-rwxrwxr-x 1 sam sam   15976 May  5 12:51 division_by_zero
-rw----- 1 sam sam 307200 May  5 12:52 core.6319
-rw-rw-r-- 1 sam sam   1544 May  5 13:00 out_of_memory.o
-rwxrwxr-x 1 sam sam   15968 May  5 13:00 out_of_memory
-rw----- 1 sam sam 307200 May  5 13:00 core.6590
drwxrwxr-x 3 sam sam   4096 May  5 13:00 .
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/question-1$ gdb ./out_of_memory -c core.6590
GNU gdb (Ubuntu 12.1-0ubuntu1-22.04) 12.1
Copyright (C) 2022 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:
<https://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 ./out_of_memory...
(No debugging symbols found in ./out_of_memory)
[New LWP 6590]
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Core was generated by `./out_of_memory'.
Program terminated with signal SIGSEGV, Segmentation fault.
#0 0x00000529bc1a3171 in main ()
```

The core file doesn't show exactly where the error happened but we see the failure happened in the 'main' function.

```
(gdb) where
#0 0x00000529bc1a3171 in main ()
(gdb) print c
No symbol "c" in current context.
(gdb) print *c
No symbol "c" in current context.
(gdb) break 1
No line 1 in the current file.
Make breakpoint pending on future shared library load? (y or [n]) n
(gdb) break 5
No line 5 in the current file.
Make breakpoint pending on future shared library load? (y or [n]) n
(gdb) break 7
No line 7 in the current file.
Make breakpoint pending on future shared library load? (y or [n]) n
(gdb) run
Starting program: /home/sam/Desktop/git/Operation-Systems-Ex1/question-1/out_of_memory
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Program received signal SIGSEGV, Segmentation fault.
0x0000055555555171 in main ()
```

Using the '**where**' command shows the same thing, because there is nothing else in the stack trace to show.

Any attempt to debug doesn't work - trying to '**print**' the value and set breakpoints result in nothing.

### 3. Out of memory

- With debug info (-g flag in compilation): core.5384

```
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/question-1$ make out_of_memory
gcc -Wall -g -c out_of_memory.c -o out_of_memory.o
gcc -Wall -g -o out_of_memory out_of_memory.o
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/question-1$ ./out_of_memory
Segmentation fault (core dumped)
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/question-1$ ls -lart
total 8656
drwxrwxr-x 4 sam sam    4096 May  3 13:21 ..
-rw-rw-r-- 1 sam sam     302 May  3 13:53 division_by_zero.c
-rw-rw-r-- 1 sam sam    168 May  3 13:53 out_of_memory.c
-rw-rw-r-- 1 sam sam    373 May  3 14:14 makefile
-rw-rw-r-- 1 sam sam    278 May  5 11:51 stack_overflow.c
-rw-rw-r-- 1 sam sam   4400 May  5 11:59 stack_overflow.o
-rwxrwxr-x 1 sam sam   17448 May  5 11:59 stack_overflow
-rw----- 1 sam sam 8695808 May  5 11:59 core.4530
-rw-rw-r-- 1 sam sam   4024 May  5 12:07 division_by_zero.o
-rwxrwxr-x 1 sam sam   17336 May  5 12:07 division_by_zero
-rw----- 1 sam sam 307200 May  5 12:07 core.5133
-rw-rw-r-- 1 sam sam   3984 May  5 12:16 out_of_memory.o
-rwxrwxr-x 1 sam sam   17312 May  5 12:16 out_of_memory
-rw----- 1 sam sam 307200 May  5 12:17 core.5384
drwxrwxr-x 2 sam sam    4096 May  5 12:17 .
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/question-1$ gdb ./out_of_memory -c core.5384
GNU gdb (Ubuntu 12.1-0ubuntu1-22.04) 12.1
Copyright (C) 2022 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 warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://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 ./out_of_memory...
[New LWP 5384]
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Core was generated by `./out_of_memory'.
Program terminated with signal SIGSEGV, Segmentation fault.
#0 0x00005a45830a6171 in main (argc=1, argv=0x7fff81a63cf8) at out_of_memory.c:7
7      printf("%c \n", *c);
(gdb) where
#0 0x00005a45830a6171 in main (argc=1, argv=0x7fff81a63cf8) at out_of_memory.c:7
```

We can see the program is terminated in main, line 7: printf("%c \n", \*c);

Using the ‘where’ command shows the same problem.

```
(gdb) break 7
Breakpoint 1 at 0x5a45830a616d: file out_of_memory.c, line 7.
(gdb) run
Starting program: /home/sam/Desktop/git/Operation-Systems-Ex1/question-1/out_of_memory
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, main (argc=1, argv=0xfffffffffe078) at out_of_memory.c:7
7      printf("%c \n", *c);
(gdb) print c
$1 = 0xdeadbeef <error: Cannot access memory at address 0xdeadbeef>
(gdb) print *c
Cannot access memory at address 0xdeadbeef
(gdb) continue
Continuing.

Program received signal SIGSEGV, Segmentation fault.
0x000055555555171 in main (argc=1, argv=0x7fffffe078) at out_of_memory.c:7
7      printf("%c \n", *c);
(gdb) 
```

By setting a ‘breakpoint’ on this line, we can ‘run’ the program and when it’s paused on the breakpoint, try to ‘print’ the value of ‘c’ at the moment of crashing:

We notice there is an error trying to print it because we cannot access the memory at this address (According to Google “There was a practice in the '70s to use the hexadecimal code 0xDEADBEEF to indicate an invalid value”)

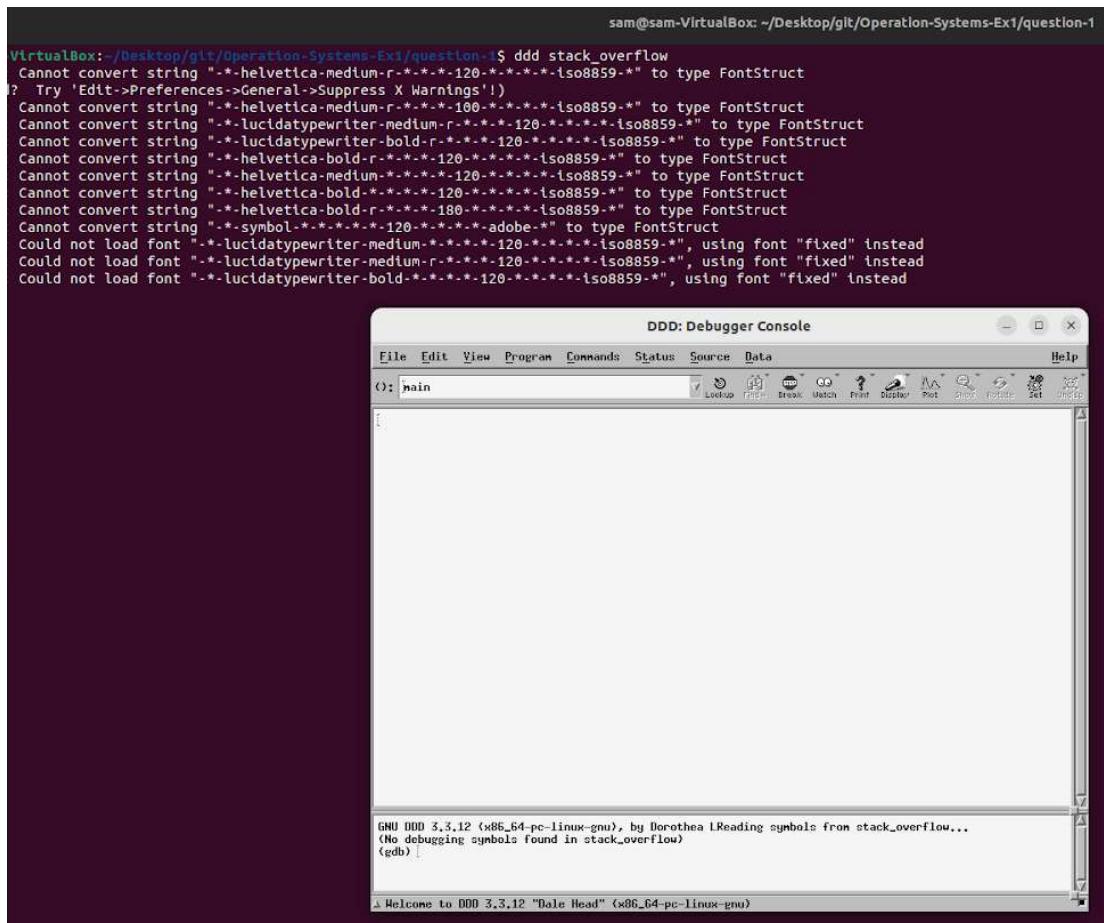
Then by typing ‘continue’, we see the crash happens right after the execution of the problematic line.

IDs: 53036281, 211791041

ddd - Graphical debugger

## 1. Stack overflow

- Without debug info:



The screenshot shows a terminal window and a DDD (Debugger) interface. The terminal window at the top displays the command `ddd stack_overflow` followed by numerous error messages indicating font conversion failures. Below it, the DDD interface has a menu bar with File, Edit, View, Program, Commands, Status, Source, Data, and Help. A toolbar below the menu contains icons for Lock Up, Print, Break, Match, Disp, Prof, and other debugger functions. The main window is titled "DDD: Debugger Console" and shows the message "GNU DDD 3.3.12 (x86\_64-pc-linux-gnu), by Dorothea LReading symbols from stack\_overflow... (No debugging symbols found in stack\_overflow...) (gdb) [ ]". At the bottom, a status bar says "Welcome to DDD 3.3.12 "Dale Head" (x86\_64-pc-linux-gnu)".

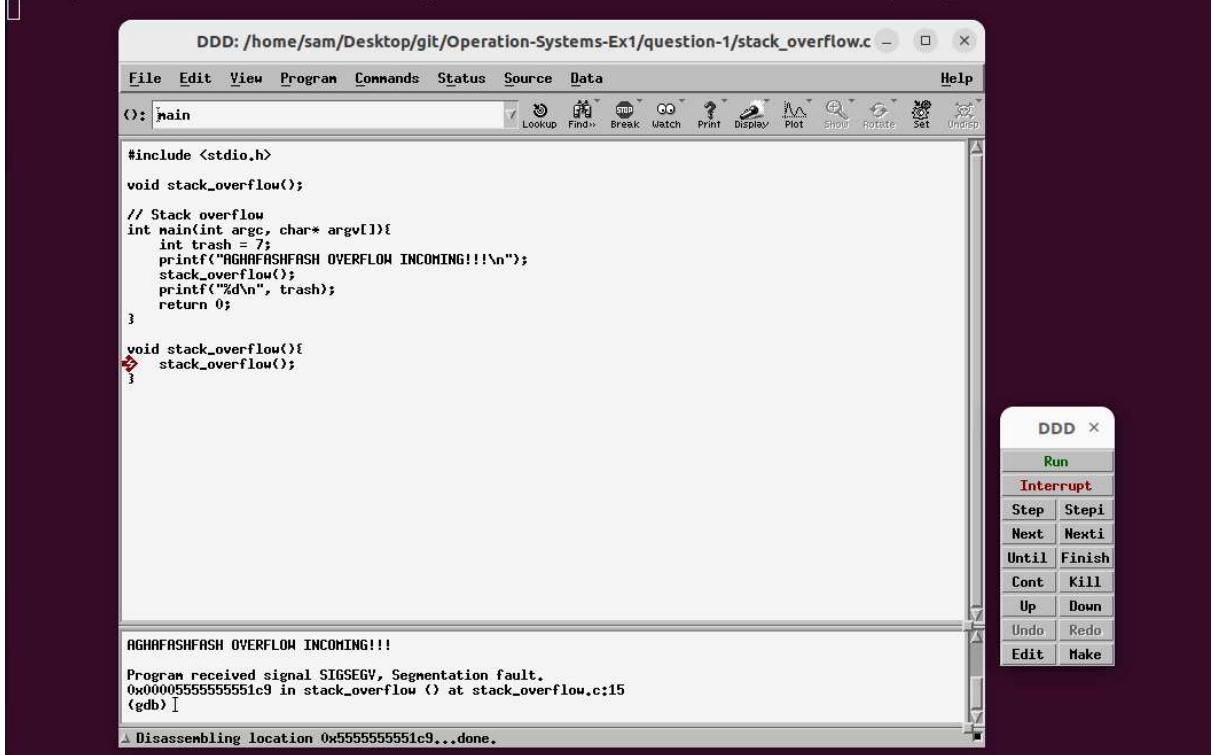
The code doesn't show and “No debugging symbols found in \_\_\_\_” - We are unable to debug because the -g flag was not included in compilation.

IDs: 53036281, 211791041

## 1. Stack overflow

- With debug info (-g flag in compilation):

```
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/question-1$ ddd stack_overflow
Warning: Cannot convert string "-*-helvetica-medium-r-*-*-120-*-*-iso8859-*" to type FontStruct
(Annoyed? Try 'Edit->Preferences->General->Suppress X Warnings'!)
Warning: Cannot convert string "-*-helvetica-medium-r-*-*-100-*-*-iso8859-*" to type FontStruct
Warning: Cannot convert string "-*-lucidatypewriter-medium-r-*-*-120-*-*-iso8859-*" to type FontStruct
Warning: Cannot convert string "-*-lucidatypewriter-bold-r-*-*-120-*-*-iso8859-*" to type FontStruct
Warning: Cannot convert string "-*-helvetica-bold-r-*-*-120-*-*-iso8859-*" to type FontStruct
Warning: Cannot convert string "-*-helvetica-medium-*-*-120-*-*-iso8859-*" to type FontStruct
Warning: Cannot convert string "-*-helvetica-bold-*-*-120-*-*-iso8859-*" to type FontStruct
Warning: Cannot convert string "-*-symbol-*-*-120-*-*-adobe-*" to type FontStruct
Warning: Could not load font "-*-lucidatypewriter-medium-*-*-120-*-*-iso8859-*", using font "fixed" instead
Warning: Could not load font "-*-lucidatypewriter-medium-r-*-*-120-*-*-iso8859-*", using font "fixed" instead
Warning: Could not load font "-*-lucidatypewriter-bold-*-*-120-*-*-iso8859-*", using font "fixed" instead
```



Shows exactly where the fault occurs and we are able to fully debug the program by setting breakpoints and print values (right-click var/func to see options).

We see there is a Segmentation fault in the 'stack\_overflow' function, line 15.

The stack trace is full so the program crashed.

## 2. Division by zero - Floating point exception

- Without debug info:

The terminal window shows the command `ddd division_by_zero` being run, which results in numerous errors related to font conversion and loading. The DDD Debugger Console window shows a warning message: "No debugging symbols found in division\_by\_zero..."

```
VirtualBox:~/Desktop/git/Operation-Systems-Ex1/question-1$ ddd division_by_zero
Cannot convert string "-*-helvetica-medium-r-*-*-120-*-*-*-iso8859-*" to type FontStruct
? Try 'Edit->Preferences->General->Suppress X Warnings!'
Cannot convert string "-*-helvetica-medium-r-*-*-120-*-*-*-iso8859-*" to type FontStruct
Cannot convert string "-*-lucidatypewriter-medium-r-*-*-120-*-*-*-iso8859-*" to type FontStruct
Cannot convert string "-*-lucidatypewriter-bold-r-*-*-120-*-*-*-iso8859-*" to type FontStruct
Cannot convert string "-*-helvetica-bold-r-*-*-120-*-*-*-iso8859-*" to type FontStruct
Cannot convert string "-*-helvetica-medium-r-*-*-120-*-*-*-iso8859-*" to type FontStruct
Cannot convert string "-*-helvetica-bold-r-*-*-180-*-*-*-iso8859-*" to type FontStruct
Cannot convert string "-*-symbol-*-*-120-*-*-*-adobe-*" to type FontStruct
Could not load font "-*-lucidatypewriter-medium-r-*-*-120-*-*-*-iso8859-*", using font "fixed" instead
Could not load font "-*-lucidatypewriter-medium-r-*-*-120-*-*-*-iso8859-*", using font "fixed" instead
Could not load font "-*-lucidatypewriter-bold-r-*-*-120-*-*-*-iso8859-*", using font "fixed" instead
```

GNU DDD 3.3.12 (x86\_64-pc-linux-gnu), by Dorothea (Reading symbols from division\_by\_zero...
(No debugging symbols found in division\_by\_zero)
(gdb) [redacted]

Welcome to DDD 3.3.12 "Dale Head" (x86\_64-pc-linux-gnu)

The code doesn't show and “No debugging symbols found in \_\_\_\_” - We are unable to debug because the -g flag was not included in compilation.

IDs: 53036281, 211791041

## 2. Division by zero - Floating point exception

- With debug info (-g flag in compilation):

We're able to see exactly where the exception occurred just by pressing 'run'.

We see there is an Arithmetic exception in main, line 10.

The screenshot shows the DDD (Debian Debug) interface. The left pane displays the source code of `division_by_zero.c`:

```
#include <stdio.h>

// Division by zero - Floating point exception
int main(int argc, char* argv[])
{
    int num;
    // num = 5/0;      // shows warning when compiling
    // division by 0 without a warning when compiling
    int zero = 0;
    num = 5/zero;
    printf("%d \n", num);
    return 0;
}
```

The right pane shows the DDD control panel with the "Run" button highlighted. The bottom status bar indicates:

- Using host libthread\_db library "/lib/x86\_64-linux-gnu/libthread\_db.so.1".
- Program received signal SIGFPE, Arithmetic exception.
- 0x00005555555169 in main (argc=1, argv=0xfffffff0e028) at division\_by\_zero.c:10
- (gdb)

A message at the bottom says: "Starting program: /home/sam/Desktop/git/Operation-Systems-Ex1/question-1/division\_by\_zero"

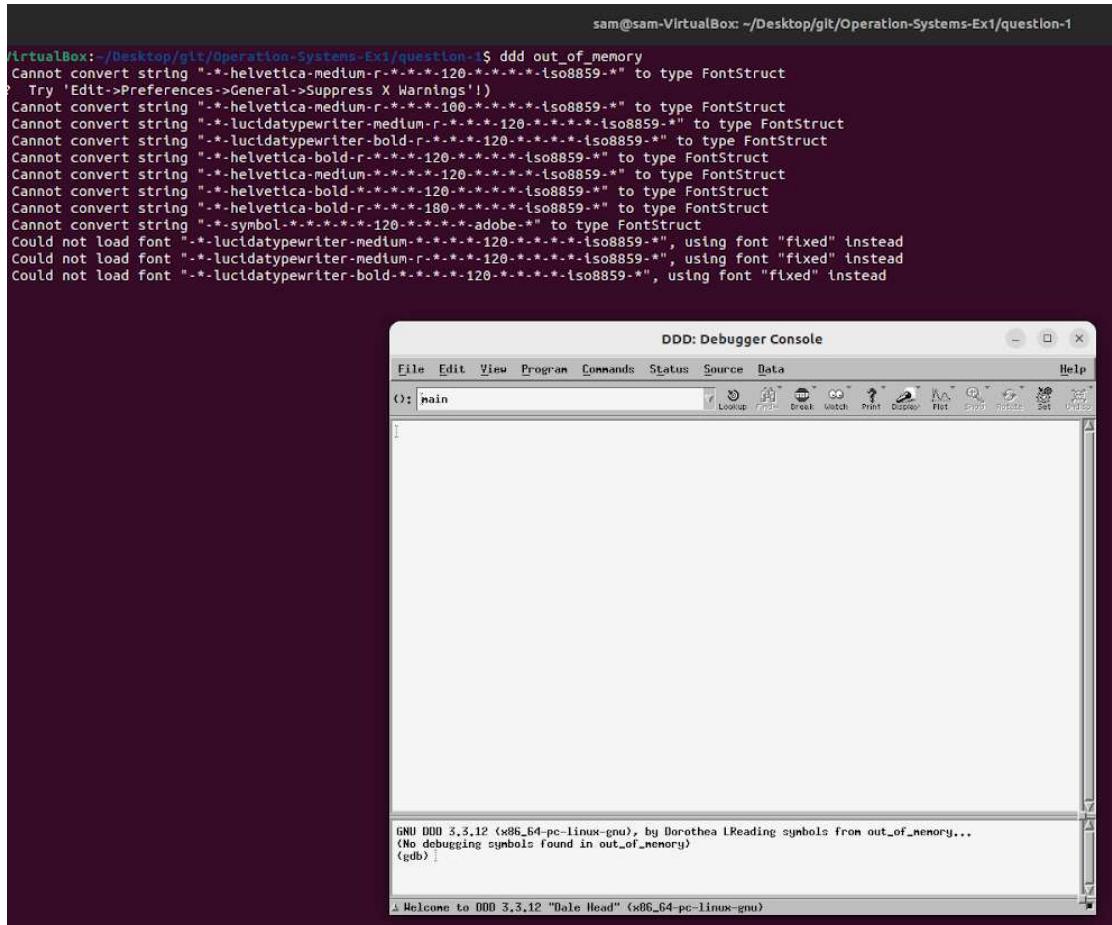
But, we can also debug by placing a 'breakpoint' in main, and stepping line by line.

The screenshot shows two instances of the DDD debugger side-by-side. Both instances have the same source code and control panels. The left instance has a breakpoint set on line 10 of the `main` function. The right instance is at the same stage but has not yet hit the breakpoint. The bottom status bar in both instances shows:

- GDB 3.3.12 (x86\_64-pc-linux-gnu), by Dorothée (Reading symbols from division\_by\_zero...)
- (gdb)
- Set a breakpoint at the selected function

### 3. Out of memory

- Without debug info:



```
VirtualBox:~/Desktop/git/Operation-Systems-Ex1/question-1$ ddd out_of_memory
Cannot convert string "-*-helvetica-medium-r-*-*-120-*-*-*iso8859-*" to type FontStruct
? Try 'Edit->Preferences->General->Suppress Warnings!'
Cannot convert string "-*-helvetica-medium-r-*-*-100-*-*-*iso8859-*" to type FontStruct
Cannot convert string "-*-luciddatapewriter-medium-r-*-*-120-*-*-*iso8859-*" to type FontStruct
Cannot convert string "-*-luciddatapewriter-bold-r-*-*-120-*-*-*iso8859-*" to type FontStruct
Cannot convert string "-*-helvetica-bold-r-*-*-120-*-*-*iso8859-*" to type FontStruct
Cannot convert string "-*-helvetica-medium-r-*-*-120-*-*-*iso8859-*" to type FontStruct
Cannot convert string "-*-helvetica-bold-r-*-*-120-*-*-*iso8859-*" to type FontStruct
Cannot convert string "-*-symbol-*-*-*120-*-*-*adobe-*" to type FontStruct
Could not load font "-*-luciddatapewriter-medium-r-*-*-120-*-*-*iso8859-*", using font "fixed" instead
Could not load font "-*-luciddatapewriter-medium-r-*-*-120-*-*-*iso8859-*", using font "fixed" instead
Could not load font "-*-luciddatapewriter-bold-r-*-*-120-*-*-*iso8859-*", using font "fixed" instead
```

GNU DDD 3.3.12 (x86\_64-pc-linux-gnu), by Dorothea LReading symbols from out\_of\_memory...
(No debugging symbols found in out\_of\_memory)
(gdb)

Welcome to DDD 3.3.12 "Dale Head" (x86\_64-pc-linux-gnu)

The code doesn't show and “No debugging symbols found in \_\_\_\_” - We are unable to debug because the -g flag was not included in compilation.

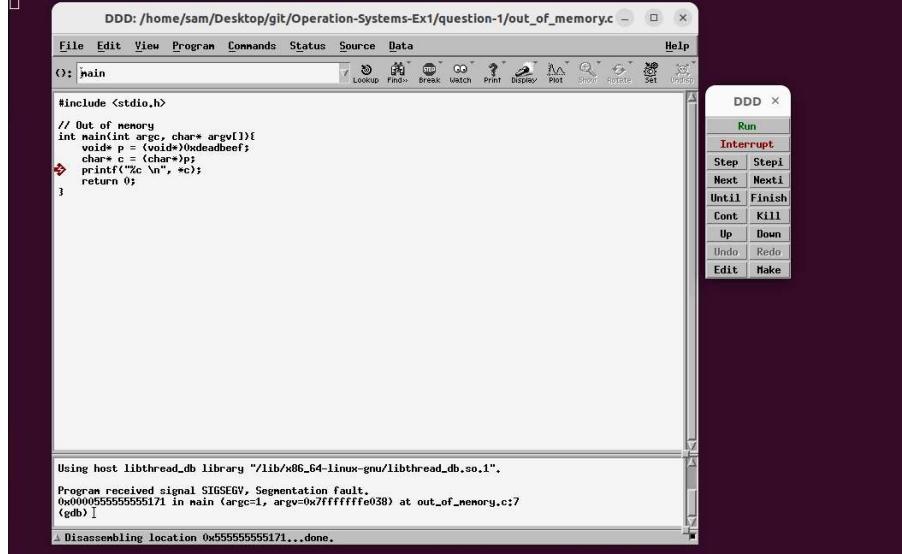
IDs: 53036281, 211791041

### 3. Out of memory

- With debug info (-g flag in compilation):

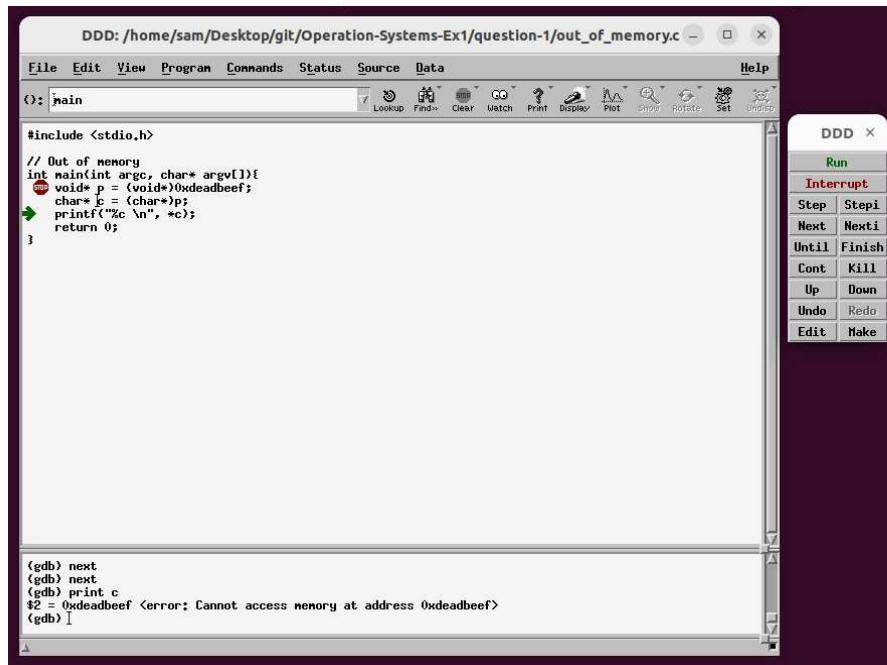
Right away we see the program occurs in main, line 7.

```
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/question-1$ ddd out_of_memory
Warning: Cannot convert string "./*-helvetica-medium-r-*-*-*120-*-*-*-ls08859-*" to type FontStruct
(Annoyed? Try 'Edit->Preferences->General->Suppress X Warnings')
Warning: Cannot convert string "./*-helvetica-medium-r-*-*-*100-*-*-*-ls08859-*" to type FontStruct
Warning: Cannot convert string "./*-lucidatypewriter-medium-r-*-*-*120-*-*-*-ls08859-*" to type FontStruct
Warning: Cannot convert string "./*-lucidatypewriter-bold-r-*-*-*120-*-*-*-ls08859-*" to type FontStruct
Warning: Cannot convert string "./*-helvetica-bold-r-*-*-*120-*-*-*-ls08859-*" to type FontStruct
Warning: Cannot convert string "./*-helvetica-bold-r-*-*-*180-*-*-*-ls08859-*" to type FontStruct
Warning: Cannot convert string "./*-symbol-*-*-*120-*-*-*-adobe-*" to type FontStruct
Warning: Could not load font "./*-lucidatypewriter-medium-*-*-*120-*-*-*-ls08859-*", using font "fixed" instead
Warning: Could not load font "./*-lucidatypewriter-bold-*-*-*120-*-*-*-ls08859-*", using font "fixed" instead
Warning: Could not load font "./*-lucidatypewriter-medium-r-*-*-*120-*-*-*-ls08859-*", using font "fixed" instead
[...]
```



The screenshot shows the DDD debugger interface. The left pane displays the C source code for 'out\_of\_memory.c'. A red diamond symbol marks the breakpoint at line 7. The right pane shows the GDB command line with the error message 'Cannot access memory at address 0xdeadbeef'. A context menu is open over the line of code, with the 'Print' option highlighted.

To see exactly what happened, we can debug the program by setting breakpoints and print values.



The screenshot shows the DDD debugger interface. The left pane displays the C source code for 'out\_of\_memory.c'. A red circle marks a breakpoint on line 7. The right pane shows the GDB command line with the command 'next' entered twice. The output shows an attempt to print variable 'c' results in an error: 'Cannot access memory at address 0xdeadbeef'.

We set a breakpoint on main, ran the programs and pressed next twice to get to the problematic line. Then we tried printing c (by right-clicking c and choosing print) and we got an error saying we cannot access memory at the given address.

## Question 2

Poisson calculation ( $\lambda = 2$ ,  $k=3$ ), (we got the sum through the website:<https://statisticshelper.com/poisson-probability-calculator/#answer>)

### **Solution:**

$P(3)$  Probability of exactly 3 occurrences

If using a calculator, you can enter  $\lambda = 2$  and  $x = 3$  into a poisson probability distribution function (PDF). If doing this by hand, apply the poisson probability formula:

$$P(x) = \frac{e^{-\lambda} \cdot \lambda^x}{x!}$$

where  $x$  is the number of occurrences,  $\lambda$  is the mean number of occurrences, and  $e$  is the constant 2.718. Substituting in values for this problem,  $x = 3$  and  $\lambda = 2$ , we have

$$P(3) = \frac{e^{-2} \cdot 2^3}{3!}$$

Evaluating the expression, we have

$$P(3) = 0.18044704431548$$

After we run ./Poisson from our C program we get the same answer:

```
● lior@ubunto:~/Desktop/OperatingSystem/Ex1_Lior/q2$ make
gcc -Wall -g -c Poisson.c -o Poisson.o
gcc -Wall -g -o Poisson Poisson.o -lm
● lior@ubunto:~/Desktop/OperatingSystem/Ex1_Lior/q2$ ls
makefile  Poisson  Poisson.c  Poisson.o
● lior@ubunto:~/Desktop/OperatingSystem/Ex1_Lior/q2$ ./Poisson 2 3
0.180447041988373
```

```
● lior@ubunto:~/Desktop/OperatingSystem/Ex1_Lior/q2$ make
gcc -Wall -g -c Poisson.c -o Poisson.o
gcc -Wall -g -o Poisson Poisson.o -lm
● lior@ubunto:~/Desktop/OperatingSystem/Ex1_Lior/q2$ ls
makefile  Poisson  Poisson.c  Poisson.o
● lior@ubunto:~/Desktop/OperatingSystem/Ex1_Lior/q2$ ./Poisson 2 3
0.180447
```

**Question 3**

Poisson calculations for the next lambda and k values:

$\lambda$	k	ער
2	1	1
2	10	2
2	2	3
3	3	4
100	3	5

(we got the sum through the website:<https://statisticshelper.com/poisson-probability-calculator/#answer>)

(Lambda = 2, k=1):

**Solution:****P(1) Probability of exactly 1 occurrences**

If using a calculator, you can enter  $\lambda = 2$  and  $x = 1$  into a poisson probability distribution function (PDF). If doing this by hand, apply the poisson probability formula:

$$P(x) = \frac{e^{-\lambda} \cdot \lambda^x}{x!}$$

where  $x$  is the number of occurrences,  $\lambda$  is the mean number of occurrences, and  $e$  is the constant 2.718. Substituting in values for this problem,  $x = 1$  and  $\lambda = 2$ , we have

$$P(1) = \frac{e^{-2} \cdot 2^1}{1!}$$

Evaluating the expression, we have

$$P(1) = 0.27067056647323$$

(Lambda = 2, k=10):

**Solution:****P(10) Probability of exactly 10 occurrences**

If using a calculator, you can enter  $\lambda = 2$  and  $x = 10$  into a poisson probability distribution function (PDF). If doing this by hand, apply the poisson probability formula:

$$P(x) = \frac{e^{-\lambda} \cdot \lambda^x}{x!}$$

where  $x$  is the number of occurrences,  $\lambda$  is the mean number of occurrences, and  $e$  is the constant 2.718. Substituting in values for this problem,  $x = 10$  and  $\lambda = 2$ , we have

$$P(10) = \frac{e^{-2} \cdot 2^{10}}{10!}$$

Evaluating the expression, we have

$$P(10) = 3.818985064878E - 5$$

**IDs: 53036281, 211791041**

(Lambda = 2, k=2):

**Solution:**

**P(2) Probability of exactly 2 occurrences**

If using a calculator, you can enter  $\lambda = 2$  and  $x = 2$  into a poisson probability distribution function (PDF). If doing this by hand, apply the poisson probability formula:

$$P(x) = \frac{e^{-\lambda} \cdot \lambda^x}{x!}$$

where  $x$  is the number of occurrences,  $\lambda$  is the mean number of occurrences, and  $e$  is the constant 2.718. Substituting in values for this problem,  $x = 2$  and  $\lambda = 2$ , we have

$$P(2) = \frac{e^{-2} \cdot 2^2}{2!}$$

Evaluating the expression, we have

$$P(2) = 0.27067056647323$$

(Lambda = 3, k=3):

**Solution:**

**P(3) Probability of exactly 3 occurrences**

If using a calculator, you can enter  $\lambda = 3$  and  $x = 3$  into a poisson probability distribution function (PDF). If doing this by hand, apply the poisson probability formula:

$$P(x) = \frac{e^{-\lambda} \cdot \lambda^x}{x!}$$

where  $x$  is the number of occurrences,  $\lambda$  is the mean number of occurrences, and  $e$  is the constant 2.718. Substituting in values for this problem,  $x = 3$  and  $\lambda = 3$ , we have

$$P(3) = \frac{e^{-3} \cdot 3^3}{3!}$$

Evaluating the expression, we have

$$P(3) = 0.22404180765539$$

(Lambda = 100, k=3):

**Solution:**

**P(3) Probability of exactly 3 occurrences**

If using a calculator, you can enter  $\lambda = 100$  and  $x = 3$  into a poisson probability distribution function (PDF). If doing this by hand, apply the poisson probability formula:

$$P(x) = \frac{e^{-\lambda} \cdot \lambda^x}{x!}$$

where  $x$  is the number of occurrences,  $\lambda$  is the mean number of occurrences, and  $e$  is the constant 2.718. Substituting in values for this problem,  $x = 3$  and  $\lambda = 100$ , we have

$$P(3) = \frac{e^{-100} \cdot 100^3}{3!}$$

Evaluating the expression, we have

$$P(3) = 6.2001266267014E - 39$$

After we run ./Poisson from our C program we get the same answer:

```
● lior@ubuntu:~/Desktop/OperatingSystem/Ex1_Lior/q3$ make
gcc -Wall -g -c poisson.c
gcc -Wall -g -c libpoisson.c
gcc -Wall -g -shared libpoisson.o -o libpoisson.so
gcc -Wall -g -o Poisson poisson.o ./libpoisson.so -lm
● lior@ubuntu:~/Desktop/OperatingSystem/Ex1_Lior/q3$ ls
libpoisson.c libpoisson.h libpoisson.o libpoisson.so makefile Poisson poisson.c poisson.o
● lior@ubuntu:~/Desktop/OperatingSystem/Ex1_Lior/q3$ ./Poisson
0.270670566473225
0.000038189850649
0.270670566473225
0.224041807655388
0.0000000000000000
```

IDs: 53036281, 211791041

```
● lior@ubunto:~/Desktop/OperatingSystem/Ex1_Lior/q3$ make
gcc -Wall -g -c poisson.c
gcc -Wall -g -c libpoisson.c
gcc -Wall -g -shared libpoisson.o -o libpoisson.so
gcc -Wall -g -o Poisson poisson.o ./libpoisson.so -lm
● lior@ubunto:~/Desktop/OperatingSystem/Ex1_Lior/q3$ ls
libpoisson.c libpoisson.h libpoisson.o libpoisson.so makefile Poisson poisson.c poisson.o
● lior@ubunto:~/Desktop/OperatingSystem/Ex1_Lior/q3$ ./Poisson
0.270671
0.000038
0.270671
0.224042
0.000000
```

IDs: 53036281, 211791041

#### Question 4

We created 7 test input files (.txt), that together, cover all the lines in the file. The test cases are in the test\_inputs folder, and the gcov file outputs are in the test\_gcov\_outputs folder (We move them to the test\_gcov\_outputs folder and rename after running each test).

We also added "|| true" to the end of the execution command to avoid the program from stopping when the errors occurs.

By running this we can see the coverage precent raising with every test run.

Run `make coverage` to run the coverage test:

```
coverage:  
$(CC) $(GCOV) dijkstra.c -o dijkstra  
  
./dijkstra < $(INPUTS_FOLDER)/valid_input.txt || true  
gcov dijkstra.c  
mv dijkstra.c.gcov $(OUTPUTS_FOLDER)/valid_input.txt.gcov  
  
./dijkstra < $(INPUTS_FOLDER)/invalid_input_early_linebreak.txt || true  
gcov dijkstra.c  
mv dijkstra.c.gcov $(OUTPUTS_FOLDER)/invalid_input_early_linebreak.txt.gcov  
  
./dijkstra < $(INPUTS_FOLDER)/invalid_input_vertices_number.txt || true  
gcov dijkstra.c  
mv dijkstra.c.gcov $(OUTPUTS_FOLDER)/invalid_input_vertices_number.txt.gcov  
  
./dijkstra < $(INPUTS_FOLDER)/invalid_input_double_space.txt || true  
gcov dijkstra.c  
mv dijkstra.c.gcov $(OUTPUTS_FOLDER)/invalid_input_double_space.txt.gcov  
  
./dijkstra < $(INPUTS_FOLDER)/invalid_input_negative_weights.txt || true  
gcov dijkstra.c  
mv dijkstra.c.gcov $(OUTPUTS_FOLDER)/invalid_input_negative_weights.txt.gcov  
  
./dijkstra < $(INPUTS_FOLDER)/invalid_input_non_integer.txt || true  
gcov dijkstra.c  
mv dijkstra.c.gcov $(OUTPUTS_FOLDER)/invalid_input_non_integer.txt.gcov  
  
./dijkstra < $(INPUTS_FOLDER)/invalid_input_endline_space.txt || true  
gcov dijkstra.c  
mv dijkstra.c.gcov $(OUTPUTS_FOLDER)/invalid_input_endline_space.txt.gcov
```

(makefile)

IDs: 53036281, 211791041

Example run;

```
 sam@sam-VirtualBox:~/Desktop/git/Operating-Systems/Operation-Systems-Ex1/q4$ make coverage
gcc -fprofile-arcs -ftest-coverage dijkstra.c -o dijkstra
./dijkstra < test_inputs/valid_input.txt || true
How many vertices do you want?
Vertex      Distance from Source
0              0
1              2
2              3
gcov dijkstra.c
File 'dijkstra.c'
Lines executed:81.48% of 54
Creating 'dijkstra.c.gcov'

Lines executed:81.48% of 54
mv dijkstra.c.gcov test_gcov_outputs/valid_input.txt.gcov
./dijkstra < test_inputs/invalid_input_early_linebreak.txt || true
How many vertices do you want?
ERROR: Invalid input! a space was expected, received '
'
gcov dijkstra.c
File 'dijkstra.c'
Lines executed:85.19% of 54
Creating 'dijkstra.c.gcov'

Lines executed:85.19% of 54
mv dijkstra.c.gcov test_gcov_outputs/invalid_input_early_linebreak.txt.gcov

./dijkstra < test_inputs/invalid_input_vertices_number.txt || true
How many vertices do you want?
ERROR: Invalid input! a line-break was expected, received .
gcov dijkstra.c
File 'dijkstra.c'
Lines executed:88.89% of 54
Creating 'dijkstra.c.gcov'

Lines executed:88.89% of 54
mv dijkstra.c.gcov test_gcov_outputs/invalid_input_vertices_number.txt.gcov
./dijkstra < test_inputs/invalid_input_double_space.txt || true
How many vertices do you want?
ERROR: Invalid input! a space was expected, received ' '
gcov dijkstra.c
File 'dijkstra.c'
Lines executed:88.89% of 54
Creating 'dijkstra.c.gcov'

Lines executed:88.89% of 54
mv dijkstra.c.gcov test_gcov_outputs/invalid_input_double_space.txt.gcov
./dijkstra < test_inputs/invalid_input_negative_weights.txt || true
How many vertices do you want?
ERROR: Negative weights are not allowed!
gcov dijkstra.c
File 'dijkstra.c'
Lines executed:92.59% of 54
Creating 'dijkstra.c.gcov'

Lines executed:92.59% of 54
mv dijkstra.c.gcov test_gcov_outputs/invalid_input_negative_weights.txt.gcov
./dijkstra < test_inputs/invalid_input_non_integer.txt || true
How many vertices do you want?
ERROR: Invalid input! Please enter integers only!
gcov dijkstra.c
File 'dijkstra.c'
Lines executed:96.30% of 54
Creating 'dijkstra.c.gcov'

Lines executed:96.30% of 54
mv dijkstra.c.gcov test_gcov_outputs/invalid_input_non_integer.txt.gcov
./dijkstra < test_inputs/invalid_input_endline_space.txt || true
How many vertices do you want?
ERROR: Invalid input! a line-break was expected, received ''
Don't add a space at the end of the row.gcov dijkstra.c
File 'dijkstra.c'
Lines executed:100.00% of 54
Creating 'dijkstra.c.gcov'

Lines executed:100.00% of 54
mv dijkstra.c.gcov test_gcov_outputs/invalid_input_endline_space.txt.gcov
```

IDs: 53036281, 211791041

I will go over a few tests and analyse them here:

- Test #1 - Valid input (valid\_input.txt):

```
./dijkstra < test_inputs/valid_input.txt || true
How many vertices do you want?
Vertex          Distance from Source
0                  0
1                  2
2                  3
gcov dijkstra.c
File 'dijkstra.c'
Lines executed:81.48% of 54
Creating 'dijkstra.c.gcov'

Lines executed:81.48% of 54
mv dijkstra.c.gcov test_gcov_outputs/valid_input.txt.gcov
```

We see 81.48% of lines were executed with only 1 run, of valid input.

By analysing the generate .gcov file we can see which lines were not executed:  
(Each line starts with the number of times it was executed. ##### = 0 times)

```
    1: 100:     int** graph = (int **) malloc (V * sizeof(int));
    1: 101:         if ((c = getchar()) != '\n'){
    1: 102:             if ((c = getchar()) != '\n'){
    1: 103:                 if ((c = getchar()) != '\n'){
    1: 104:                     fprintf(stderr, "ERROR: Invalid input! a line-break was expected, received %c.\n", c);
    1: 105:                     exit(EXIT_FAILURE);
    1: 106:                 }
    1: 107:
    1: 108:                 for (int i = 0; i < V; i++){
    1: 109:                     graph[i] = (int *)malloc(V * sizeof(int));
    1: 110:                         for (int j = 0; j < V; j++){
    1: 111:                             // skip 1 space or print error if less or more than 1 space was given (only done
    1: 112:                             if (j != 0 && (c = getchar()) != ' ') {
    1: 113:                                 fprintf(stderr, "ERROR: Invalid input! a space was expected, received '%c'.
    1: 114:                                 exit(EXIT_FAILURE);
    1: 115:                             }
    1: 116:
    1: 117:                             if (scanf("%d", &num) == 1) {
    1: 118:                                 if (num >= 0){
    1: 119:                                     graph[i][j] = num;
    1: 120:                                 } else {
    1: 121:                                     fprintf(stderr, "ERROR: Negative weights are not allowed!\n");
    1: 122:                                     exit(EXIT_FAILURE);
    1: 123:                                 }
    1: 124:                             }
    1: 125:                         }
    1: 126:                     else {
    1: 127:                         fprintf(stderr, "ERROR: Invalid input! Please enter integers only!\n");
    1: 128:                         exit(EXIT_FAILURE);
    1: 129:                     }
    1: 130:                 }
    1: 131:                 if ((c = getchar()) != '\n'){
    1: 132:                     fprintf(stderr, "ERROR: Invalid input! a line-break was expected, received '%c'.
    1: 133:                     exit(EXIT_FAILURE);
    1: 134:                 }
    1: 135:             }
    1: 136:
    1: 137:             // Function call
    1: 138:             dijkstra(graph, 0, V);
    1: 139:
    1: 140:             return 0;
    1: 141: }
    1: 142:#endif
```

- Test #2 - Invalid input (invalid\_input\_early\_linebreak.txt):

```
./dijkstra < test_inputs/invalid_input_early_linebreak.txt || true
How many vertices do you want?
ERROR: Invalid input! a space was expected, received ' '
gcov dijkstra.c
File 'dijkstra.c'
Lines executed:85.19% of 54
Creating 'dijkstra.c.gcov'

Lines executed:85.19% of 54
mv dijkstra.c.gcov test_gcov_outputs/invalid_input_early_linebreak.txt.gcov
```

We see the percent of executes lines raised to 85.19%, with 2 test runs  
(as seen on the top of .gcov file: “Runs:2”).

```
[ ]: 0:Source:dijkstra.c
[-]: 0:Graph:dijkstra.gcno
[-]: 0:Data:dijkstra.gcda
[-]: 0:Runs:2
```

IDs: 53036281, 211791041

By analysing the generated .gcov file we can see which lines were not executed.

- Test #7 -

```
./dijkstra < test_inputs/invalid_input_endline_space.txt || true
How many vertices do you want?
ERROR: Invalid input! a line-break was expected, received ''
Don't add a space at the end of the row.gcov dijkstra.c
File 'dijkstra.c'
Lines executed:100.00% of 54
Creating 'dijkstra.c.gcov'

Lines executed:100.00% of 54
mv dijkstra.c.gcov test_gcov_outputs/invalid_input_endline_space.txt.gcov
```

We see that after 7 test runs:

```
[ ]: 0:Source:dijkstra.c
[ ]: -: 0:Graph:dijkstra.gcno
[ ]: -: 0:Data:dijkstra.gcda
[ ]: -: 0:Runs:7
```

If we analyse the .gcov file we will see that 100% of the lines were executed, indicating a well written test file that checks all edge-cases.

```
--: 89:
--: 90:#ifndef TEST
7: 91:int main()
--: 92:{
--: 93:     /* Let us create the example graph discussed above */
7: 94:     int V = 0;           // Number of verties in the graph
--: 95:     char c;
--: 96:     int num;
--: 97:
7: 98:     printf("How many vertices do you want?\n");
7: 99:     scanf("%d", &V);
--: 100:
7: 101:    int** graph = (int **) malloc (V * sizeof(int *));
--: 102:
7: 103:    if ((c = getchar()) != '\n'){
1: 104:        fprintf(stderr, "ERROR: Invalid input! a line-break was expected, received %c.\n", c);
1: 105:        exit(EXIT_FAILURE);
--: 106:    }
--: 107:
9: 108:    for (int i = 0; i < V; i++){
8: 109:        graph[i] = (int *)malloc(V * sizeof(int));
29: 110:        for (int j = 0; j < V; j++){
--: 111:            // skip 1 space or print error if less or more than 1 space was given (only done af
25: 112:            if (j != 0 && (c = getchar()) != ' ') {
2: 113:                fprintf(stderr, "ERROR: Invalid input! a space was expected, received '%c'\n");
2: 114:                exit(EXIT_FAILURE);
--: 115:            }
--: 116:
23: 117:            if (scanf("%d", &num) == 1) {
22: 118:                if (num >= 0){
21: 119:                    graph[i][j] = num;
--: 120:                } else {
1: 121:                    fprintf(stderr, "ERROR: Negative weights are not allowed!\n");
1: 122:                    exit(EXIT_FAILURE);
--: 123:                }
--: 124:            }
--: 125:        }
--: 126:    }
1: 127:    else {
1: 128:        fprintf(stderr, "ERROR: Invalid input! Please enter integers only!\n");
1: 129:        exit(EXIT_FAILURE);
--: 130:    }
4: 131:    if ((c = getchar()) != '\n'){
1: 132:        fprintf(stderr, "ERROR: Invalid input! a line-break was expected, received '%c'\nDo
1: 133:        exit(EXIT_FAILURE);
--: 134:    }
--: 135: }
--: 136:
--: 137: // Function call
1: 138: dijkstra(graph, 0, V);
--: 139:
1: 140: return 0;
--: 141:}
[ ]: 142:#endif
```

IDs: 53036281, 211791041

## Question 5

Simple run (without -pg flag):

```
 sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/q5$ make
gcc -Wall -g -c max_sub_array_sum.c
gcc -Wall -g -o max_sub_array_sum max_sub_array_sum.o
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/q5$ ./max_sub_array_sum 1 100
Running O(n):
Time taken for O(n): 0.000001 seconds
Running O(n^2):
Time taken for O(n^2): 0.000011 seconds
Running O(n^3):
Time taken for O(n^3): 0.000320 seconds
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/q5$ ./max_sub_array_sum 1 1000
Running O(n):
Time taken for O(n): 0.000010 seconds
Running O(n^2):
Time taken for O(n^2): 0.001771 seconds
Running O(n^3):
Time taken for O(n^3): 0.326951 seconds
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/q5$ ./max_sub_array_sum 1 10000
Running O(n):
Time taken for O(n): 0.000083 seconds
Running O(n^2):
Time taken for O(n^2): 0.114126 seconds
Running O(n^3):
Time taken for O(n^3): 377.891273 seconds
```

Run to analyse profiling data (with -pg flag):

1. Input size: 100

```
 sam@sam-Vir
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/q5$ make
gcc -Wall -g -pg -c max_sub_array_sum.c
gcc -Wall -g -pg -o max_sub_array_sum max_sub_array_sum.o
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/q5$ ./max_sub_array_sum 1 100
Running O(n):
Time taken for O(n): 0.000000 seconds
Running O(n^2):
Time taken for O(n^2): 0.000000 seconds
Running O(n^3):
Time taken for O(n^3): 0.000000 seconds
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/q5$ gprof ma
makefile      max_sub_array_sum    max_sub_array_sum.c  max_sub_array_sum.o
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/q5$ gprof max_sub_array_sum
Flat profile:
```

Each sample counts as 0.01 seconds.  
no time accumulated

%	cumulative	self	calls	self	total	name
time	seconds	seconds		Ts/call	Ts/call	
0.00	0.00	0.00	1	0.00	0.00	max_sub_array_sum_n
0.00	0.00	0.00	1	0.00	0.00	max_sub_array_sum_n2
0.00	0.00	0.00	1	0.00	0.00	max_sub_array_sum_n3

```
 Call graph (explanation follows)

granularity: each sample hit covers 4 byte(s) no time propagated

index % time    self  children   called    name
          0.00  0.00     0.00    1/1        main [9]
[1]      0.0    0.00    0.00      1    max_sub_array_sum_n [1]
-----
          0.00  0.00     0.00    1/1        main [9]
[2]      0.0    0.00    0.00      1    max_sub_array_sum_n2 [2]
-----
          0.00  0.00     0.00    1/1        main [9]
[3]      0.0    0.00    0.00      1    max_sub_array_sum_n3 [3]
```

IDs: 53036281, 211791041

2. Input size: 1,000

```
 sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/q5$ make
gcc -Wall -g -pg           -c max_sub_array_sum.c
gcc -Wall -g -pg           -o max_sub_array_sum max_sub_array_sum.o
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/q5$ ./max_sub_array_sum 1 1000
Running O(n):
Time taken for O(n): 0.000000 seconds
Running O(n^2):
Time taken for O(n^2): 0.000000 seconds
Running O(n^3):
Time taken for O(n^3): 0.331311 seconds
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/q5$ gprof max_sub_array_sum
Flat profile:

Each sample counts as 0.01 seconds.
      % cumulative   self          self      total
      time  seconds  seconds  calls  ms/call  ms/call  name
100.00      0.32    0.32       1    320.00   320.00  max_sub_array_sum_n3
  0.00      0.32    0.00       1     0.00     0.00  max_sub_array_sum_n
  0.00      0.32    0.00       1     0.00     0.00  max_sub_array_sum_n2

Call graph (explanation follows)

granularity: each sample hit covers 4 byte(s) for 3.12% of 0.32 seconds

index % time   self  children  called      name
          0.32   0.00    1/1          main [2]
[1]   100.0   0.32   0.00        1  max_sub_array_sum_n3 [1]
-----
                                     <spontaneous>
[2]   100.0   0.00   0.32          main [2]
          0.32   0.00    1/1  max_sub_array_sum_n3 [1]
          0.00   0.00    1/1  max_sub_array_sum_n [3]
          0.00   0.00    1/1  max_sub_array_sum_n2 [4]
-----
          0.00   0.00    1/1          main [2]
[3]     0.0   0.00   0.00        1  max_sub_array_sum_n [3]
-----
          0.00   0.00    1/1          main [2]
[4]     0.0   0.00   0.00        1  max_sub_array_sum_n2 [4]
```

IDs: 53036281, 211791041

3. Input size: 10,000

```
 sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/q5$ make
gcc -Wall -g -pg           -c max_sub_array_sum.c
gcc -Wall -g -pg           -o max_sub_array_sum max_sub_array_sum.o
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/q5$ ./max_sub_array_sum 1 10000
Running O(n):
Time taken for O(n): 0.000000 seconds
Running O(n^2):
Time taken for O(n^2): 0.101958 seconds
Running O(n^3):
Time taken for O(n^3): 337.822562 seconds
sam@sam-VirtualBox:~/Desktop/git/Operation-Systems-Ex1/q5$ gprof max_sub_array_sum
Flat profile:
```

Each sample counts as 0.01 seconds.

% cumulative	self time	seconds	calls	self s/call	total s/call	name
99.97	337.85	337.85	1	337.85	337.85	max_sub_array_sum_n3
0.03	337.95	0.10	1	0.10	0.10	max_sub_array_sum_n2
0.00	337.95	0.00	1	0.00	0.00	max_sub_array_sum_n

Call graph (explanation follows)

granularity: each sample hit covers 4 byte(s) for 0.00% of 337.95 seconds

index	% time	self	children	called	name
[1]	100.0	0.00	337.95		<spontaneous>
		337.85	0.00	1/1	main [1]
		0.10	0.00	1/1	max_sub_array_sum_n3 [2]
		0.00	0.00	1/1	max_sub_array_sum_n2 [3]
					max_sub_array_sum_n [4]
[2]	100.0	337.85	0.00	1	main [1]
					max_sub_array_sum_n3 [2]
[3]	0.0	0.10	0.00	1	main [1]
					max_sub_array_sum_n2 [3]
[4]	0.0	0.00	0.00	1	main [1]
					max_sub_array_sum_n [4]

IDs: 53036281, 211791041

## Question 6

Here are some example runs:

```
sam@sam-VirtualBox:~/Desktop/git/Operating-Systems/Operating-Systems-Ex1/q6$ make
gcc -Wall -g add2PB.c -o add2PB
gcc -Wall -g findPhone.c -o findPhone
sam@sam-VirtualBox:~/Desktop/git/Operating-Systems/Operating-Systems-Ex1/q6$ more phonebook.txt
Nezer Zaidenberg,054-5531415
Shani Ben Meir,054-3146629
Noya Shalom,054-2109862
Omri Cohen,054-1869245
Maya Levi,054-8783920
Yair Avraham,054-5023718
Ella Amar,054-9508136
Ron Ben David,054-6179354
Maayan Goldstein,054-1195723
Tal Bachar,054-8019463
Adam Biton,054-2451863
Ayelet Ovadla,054-1795083
Noam Yosef,054-6739185
Yael Friedman,054-7823059
Eitan Cohen,054-6412085
Roni Levi,054-3749602
Barak Mizrahi,054-5906172
Shira Levi,054-8563017
Nadav Amar,054-2309865
Avigail Cohen,054-8916237
Ido Ben Shimon,054-4058912
Tamar Avraham,054-7269138
Hadar Mizrahi,054-5183072
Maor Shalom,054-3692840
Or Shabtay,054-7940532
Roni Ovadia,054-2875019
Yamit Cohen,054-6150824
Nir Biton,054-9362074
```

Adding 2 valid inputs using ./add2PB:

```
sam@sam-VirtualBox:~/Desktop/git/Operating-Systems/Operating-Systems-Ex1/q6$ ./add2PB Samuel Lazareanu,054-1234567
sam@sam-VirtualBox:~/Desktop/git/Operating-Systems/Operating-Systems-Ex1/q6$ ./add2PB Lior Trachtman,054-1234568
sam@sam-VirtualBox:~/Desktop/git/Operating-Systems/Operating-Systems-Ex1/q6$ more phonebook.txt
Nezer Zaidenberg,054-5531415
Shani Ben Meir,054-3146629
Noya Shalom,054-2109862
Omri Cohen,054-1869245
Maya Levi,054-8783920
Yair Avraham,054-5023718
Ella Amar,054-9508136
Ron Ben David,054-6179354
Maayan Goldstein,054-1195723
Tal Bachar,054-8019463
Adam Biton,054-2451863
Ayelet Ovadla,054-1795083
Noam Yosef,054-6739185
Yael Friedman,054-7823059
Eitan Cohen,054-6412085
Roni Levi,054-3749602
Barak Mizrahi,054-5906172
Shira Levi,054-8563017
Nadav Amar,054-2309865
Avigail Cohen,054-8916237
Ido Ben Shimon,054-4058912
Tamar Avraham,054-7269138
Hadar Mizrahi,054-5183072
Maor Shalom,054-3692840
Or Shabtay,054-7940532
Roni Ovadia,054-2875019
Yamit Cohen,054-6150824
Nir Biton,054-9362074
Samuel Lazareanu,054-1234567
Lior Trachtman,054-1234568
```

Searching for them using ./findPhone:

IDs: 53036281, 211791041

```
Lior Trachtmann,054-1234568
sam@sam-VirtualBox:~/Desktop/git/Operating-Systems/Operating-Systems-Ex1/q6$ ./findPhone Lior
Lior Trachtmann,054-1234568
sam@sam-VirtualBox:~/Desktop/git/Operating-Systems/Operating-Systems-Ex1/q6$ ./findPhone Samuel
Samuel Lazareanu,054-1234567
sam@sam-VirtualBox:~/Desktop/git/Operating-Systems/Operating-Systems-Ex1/q6$ ./findPhone Or
Maor Shalom,054-3692840
Or Shabtay,054-7940532
Lior Trachtmann,054-1234568
```

(Note: we made the grep inside findPhone case-insensitive so “Or” print Maor as well, this can be disabled by removing the ‘-i’ flag from the exec command)

We wrote a few simple tests for the input (having to add 1 comma, and a hyphen in the phone number):

```
sam@sam-VirtualBox:~/Desktop/git/Operating-Systems/Operating-Systems-Ex1/q6$ ./add2PB Elon Musk 111-234987
Error! commas found: 0, commas expected: 1.
Usage: ./add2PB <name>,<phone_number>
sam@sam-VirtualBox:~/Desktop/git/Operating-Systems/Operating-Systems-Ex1/q6$ ./add2PB Elon,Musk,111-234987
Error! commas found: 2, commas expected: 1.
Usage: ./add2PB <name>,<phone_number>
sam@sam-VirtualBox:~/Desktop/git/Operating-Systems/Operating-Systems-Ex1/q6$ ./add2PB 111-234987
Error! commas found: 0, commas expected: 1.
Usage: ./add2PB <name>,<phone_number>
sam@sam-VirtualBox:~/Desktop/git/Operating-Systems/Operating-Systems-Ex1/q6$ ./add2PB
Usage: ./add2PB <name>,<phone_number>
sam@sam-VirtualBox:~/Desktop/git/Operating-Systems/Operating-Systems-Ex1/q6$ ./add2PB Elon Musk,1110000000
Error! Phone number should be written with at least one hyphen '-'.
Example: ./add2PB Full Name,054-1234567
```

Valid:

```
sam@sam-VirtualBox:~/Desktop/git/Operating-Systems/Operating-Systems-Ex1/q6$ ./add2PB Elon Musk,111-0000000
sam@sam-VirtualBox:~/Desktop/git/Operating-Systems/Operating-Systems-Ex1/q6$ ./findPhone Elon
Elon Musk,111-0000000
sam@sam-VirtualBox:~/Desktop/git/Operating-Systems/Operating-Systems-Ex1/q6$ more phonebook.txt
Nezer Zaidenberg,054-5531415
Shani Ben Meir,054-3146629
Noya Shalom,054-2109862
Omri Cohen,054-1869245
Maya Levi,054-8783920
Yair Avraham,054-5023718
Ella Amar,054-9508136
Ron Ben David,054-6179354
Maayan Goldstein,054-1195723
Tal Bachar,054-8019463
Adam Biton,054-2451863
Ayelet Ovadia,054-1795083
Noam Yosef,054-6739185
Yael Friedman,054-7823059
Eitan Cohen,054-6412085
Roni Levit,054-3749602
Barak Mizrahi,054-5906172
Shira Levi,054-8563017
Nadav Amar,054-2309865
Avigail Cohen,054-8916237
Ido Ben Shimon,054-4058912
Tamar Avraham,054-7269138
Hadar Mizrahi,054-5183072
Maor Shalom,054-3692840
Or Shabtay,054-7940532
Roni Ovadia,054-2875019
Yamit Cohen,054-6150824
Nir Biton,054-9362074
Samuel Lazareanu,054-1234567
Lior Trachtmann,054-1234568
Elon Musk,111-0000000
```