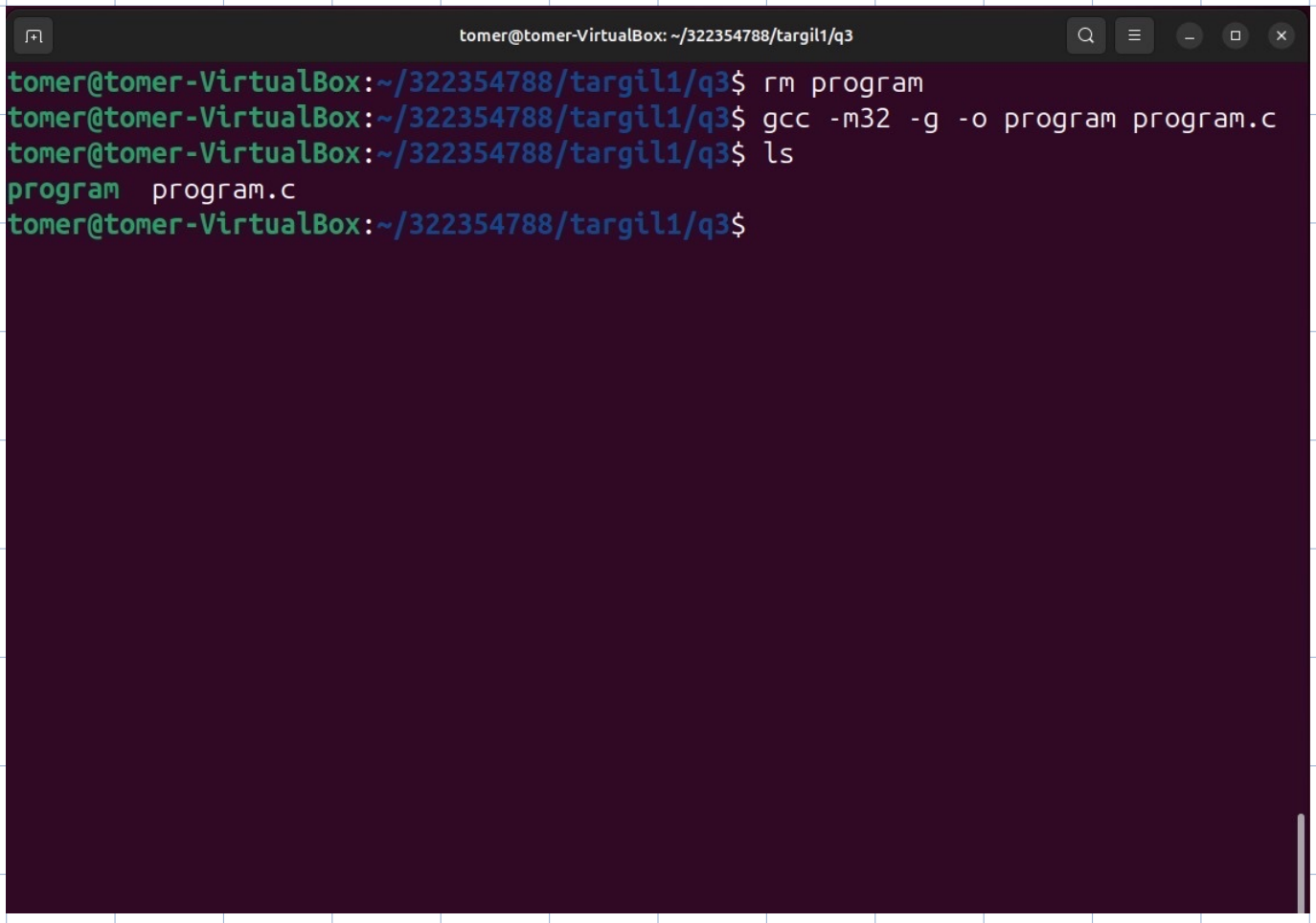
3

3.1


tomer@tomer-VirtualBox: ~/322354788/targil1/q3
tomer@tomer-VirtualBox:~/322354788/targil1/q3$ rm program
tomer@tomer-VirtualBox:~/322354788/targil1/q3$ gcc -m32 -g -o program program.c
tomer@tomer-VirtualBox:~/322354788/targil1/q3$ ls
program program.c
tomer@tomer-VirtualBox:~/322354788/targil1/q3$

3.2

```
tomer@tomer-VirtualBox: ~/322354788/targil1/q3
tomer@tomer-VirtualBox:~/322354788/targil1/q3$ rm program
tomer@tomer-VirtualBox:~/322354788/targil1/q3$ gcc -m32 -g -o program program.c
tomer@tomer-VirtualBox:~/322354788/targil1/q3$ ls
program  program.c
tomer@tomer-VirtualBox:~/322354788/targil1/q3$ gdb program
GNU gdb (Ubuntu 15.0.50.20240403-0ubuntu1) 15.0.50.20240403-git
Copyright (C) 2024 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 program...
(gdb) break func2
Breakpoint 1 at 0x119e: file program.c, line 4.
(gdb)
```

```
tomer@tomer-VirtualBox: ~/322354788/targil1/q3

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from program...
(gdb) break func2
Breakpoint 1 at 0x119e: file program.c, line 4.
(gdb) run
Starting program: /home/tomer/322354788/targil1/q3/program

This GDB supports auto-downloading debuginfo from the following URLs:
  <https://debuginfod.ubuntu.com>
Enable debuginfod for this session? (y or [n]) y
Debuginfod has been enabled.
To make this setting permanent, add 'set debuginfod enabled on' to .gdbinit.
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, func2 (a=269, b=6246) at program.c:4
4           c=a+b;
(gdb) where
#0  func2 (a=269, b=6246) at program.c:4
#1  0x565561ec in func1 (x=268, y=6247) at program.c:10
#2  0x5655621f in main () at program.c:16
(gdb)
```

3.3

<u>ebp</u> מציין את תחילת ה stack frame של הפונקציה הנוכחית, משמש לנו לעקוב אחרי כתובות של משתנים מקומיים ופרמטרים שעוברים לפונקציה.

```
tomer@tomer-VirtualBox: ~/322354788/targil1/q3
Type "apropos word" to search for commands related to "word"...
Reading symbols from program...
(gdb) break func2
Breakpoint 1 at 0x119e: file program.c, line 4.
(gdb) run
Starting program: /home/tomer/322354788/targil1/q3/program

This GDB supports auto-downloading debuginfo from the following URLs:
  <https://debuginfod.ubuntu.com>
Enable debuginfod for this session? (y or [n]) y
Debuginfod has been enabled.
To make this setting permanent, add 'set debuginfod enabled on' to .gdbinit.
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, func2 (a=269, b=6246) at program.c:4
4           c=a+b;
(gdb) where
#0  func2 (a=269, b=6246) at program.c:4
#1  0x565561ec in func1 (x=268, y=6247) at program.c:10
#2  0x5655621f in main () at program.c:16
(gdb) info registers ebp
ebp            0xffffcf38          0xffffcf38
(gdb) 
```

3.4

כתובת [ebp+8] = מספר

כתובת [ebp+12] = מספר

[ebp+8] = 8 + 0xffffcf38 = 0xffffcf40

[ebp+12] = 12 + 0xffffcf38 = 0xffffcf44

3.5

tomer@tomer-VirtualBox: ~/322354788/targil1/q3

```
  <https://debuginfod.ubuntu.com>
Enable debuginfod for this session? (y or [n]) y
Debuginfod has been enabled.
To make this setting permanent, add 'set debuginfod enabled on' to .gdbinit.
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, func2 (a=269, b=6246) at program.c:4
4           c=a+b;
(gdb) where
#0  func2 (a=269, b=6246) at program.c:4
#1  0x565561ec in func1 (x=268, y=6247) at program.c:10
#2  0x5655621f in main () at program.c:16
(gdb) info registers ebp
ebp            0xffffcf38          0xffffcf38
(gdb) x /wx $ebp+8
0xffffcf40:     0x0000010d
(gdb) x /wx $ebp+12
0xffffcf44:     0x00001866
(gdb) print &a
$1 = (int *) 0xffffcf40
(gdb) print &b
$2 = (int *) 0xffffcf44
(gdb)
```

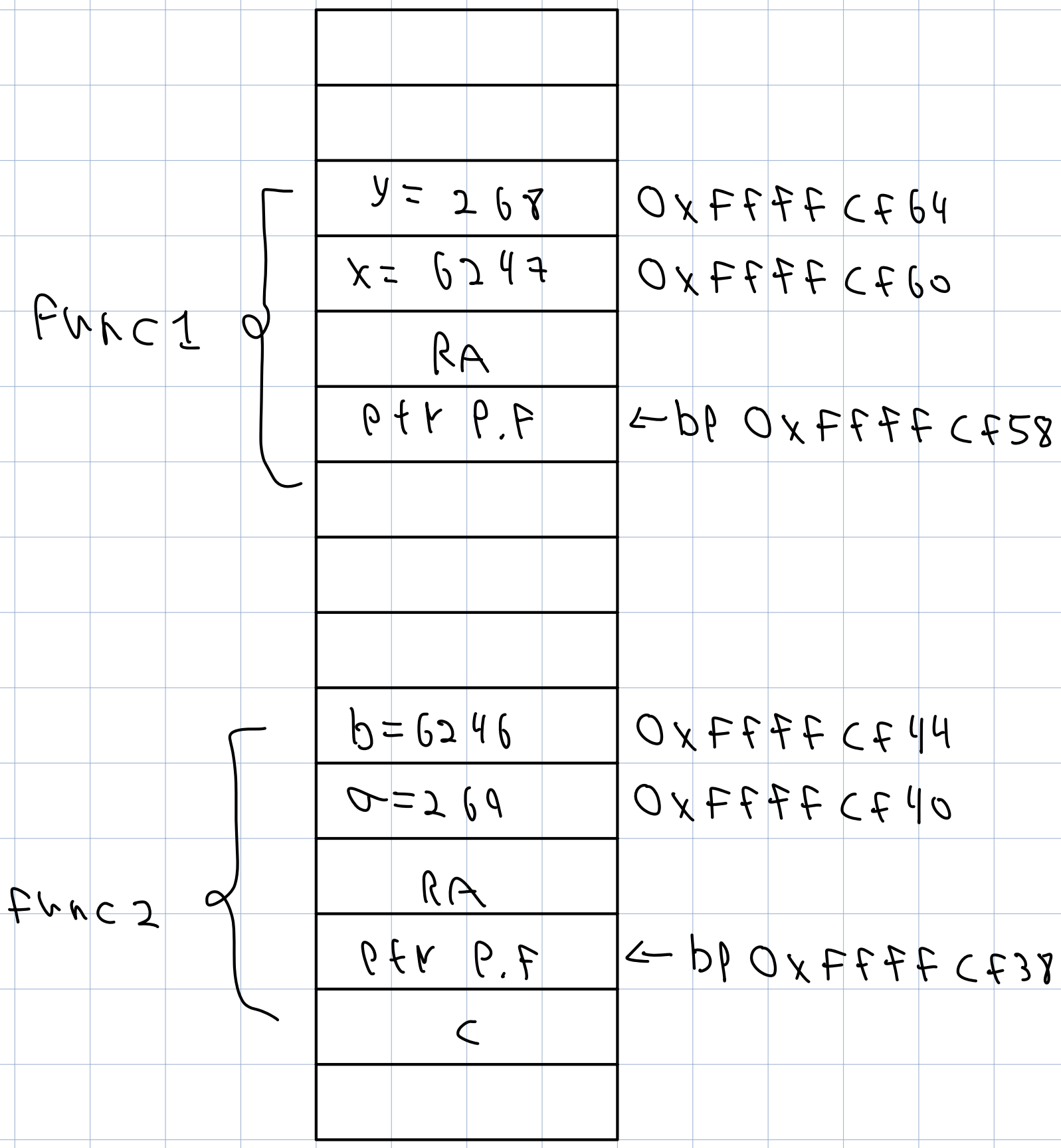
3.6

```
tomer@tomer-VirtualBox: ~/322354788/targil1/q3
Type "apropos word" to search for commands related to "word"...
Reading symbols from program...
(gdb) break func2
Breakpoint 1 at 0x119e: file program.c, line 4.
(gdb) run
Starting program: /home/tomer/322354788/targil1/q3/program

This GDB supports auto-downloading debuginfo from the following URLs:
  <https://debuginfod.ubuntu.com>
Enable debuginfod for this session? (y or [n]) y
Debuginfod has been enabled.
To make this setting permanent, add 'set debuginfod enabled on' to .gdbinit.
Downloading separate debug info for /lib/ld-linux.so.2
Downloading separate debug info for /lib32/libc.so.6
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, func2 (a=269, b=6246) at program.c:4
4           c=a+b;
(gdb) x /d $ebp+8
0xffffcf40:     269
(gdb) x /d $ebp+12
0xffffcf44:     6246
(gdb)
```

3.7

```
tomer@tomer-VirtualBox: ~/322354788/targil1/q3
  <https://debuginfod.ubuntu.com>
Enable debuginfod for this session? (y or [n]) y
Debuginfod has been enabled.
To make this setting permanent, add 'set debuginfod enabled on' to .gdbinit.
Downloading separate debug info for /lib/ld-linux.so.2
Downloading separate debug info for /lib32/libc.so.6
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, func2 (a=269, b=6246) at program.c:4
4           c=a+b;
(gdb) info registers ebp
ebp            0xffffcf38          0xffffcf38
(gdb) print *(void**) $ebp
$1 = (void *) 0xffffcf58
(gdb) print *((int*)$ebp+2)
$2 = 269
(gdb) print *((int*)$ebp+3)
$3 = 6246
(gdb) backtrace
#0  func2 (a=269, b=6246) at program.c:4
#1  0x565561ec in func1 (x=268, y=6247) at program.c:10
#2  0x5655621f in main () at program.c:16
(gdb)
```

3.8


y = 268
0xffffcf64
x = 6247
0xffffcf60
RA
ptr P.F
← bp 0xffffcf58
b = 6246
0xffffcf44
a = 269
0xffffcf40
RA
ptr P.F
← bp 0xffffcf38
c
func1
func2