שאלה 5

כאשר הרצנו valgrind קיבלנו את השגיאות הבאות:

```
matan@matan:~/IdeaProjects/OS_4/Q_5$ make valgrind_memcheck
valgrind --leak-check=full ./hello
==22543== Memcheck, a memory error detector
==22543== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==22543== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==22543== Command: ./hello
==22543==
==22543== Conditional jump or move depends on uninitialised value(s)
==22543==    at 0x109234: main (in /home/matan/IdeaProjects/OS_4/Q_5/hello)
==22543==
You entered: ./hello
==22543==
==22543== HEAP SUMMARY:
==22543==     in use at exit: 9 bytes in 1 blocks
==22543==   total heap usage: 2 allocs, 1 frees, 1,033 bytes allocated
==22543==
==22543== 9 bytes in 1 blocks are definitely lost in loss record 1 of 1
==22543==    at 0x4848899: malloc (in /usr/libexec/valgrind/vgpreload_memcheck-amd64-linux.so)
==22543==    by 0x10922A: main (in /home/matan/IdeaProjects/OS_4/Q_5/hello)
==22543==
==22543== LEAK SUMMARY:
==22543==    definitely lost: 9 bytes in 1 blocks
==22543==    indirectly lost: 0 bytes in 0 blocks
==22543==      possibly lost: 0 bytes in 0 blocks
==22543==    still reachable: 0 bytes in 0 blocks
==22543==         suppressed: 0 bytes in 0 blocks
==22543==
==22543== Use --track-origins=yes to see where uninitialised values come from
==22543== For lists of detected and suppressed errors, rerun with: -s
==22543== ERROR SUMMARY: 2 errors from 2 contexts (suppressed: 0 from 0)
matan@matan:~/IdeaProjects/OS_4/Q_5$
```

במלבן האדום זו הפקודה שהרצנו.

במלבן הירוק רואים שבוודאות איבדנו 9 בתים, נרצה יותר מידע ולכן נריץ עם הדגל g-

```
matan@matan:~/IdeaProjects/OS_4/Q_5$ make valgrind_memcheck
valgrind --leak-check=full ./hello
==22957== Memcheck, a memory error detector
==22957== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==22957== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==22957== Command: ./hello
==22957==
==22957== Conditional jump or move depends on uninitialised value(s)
==22957==    at 0x109234: main (hello.c:16)
==22957==
You entered: ./hello
==22957==
==22957== HEAP SUMMARY:
==22957==     in use at exit: 9 bytes in 1 blocks
==22957==   total heap usage: 2 allocs, 1 frees, 1,033 bytes allocated
==22957==
==22957== 9 bytes in 1 blocks are definitely lost in loss record 1 of 1
==22957==    at 0x4848899: malloc (in /usr/libexec/valgrind/vgpreload_memcheck-amd64-linux.so)
==22957==    by 0x10922A: main (hello.c:14)
==22957==
==22957== LEAK SUMMARY:
==22957==    definitely lost: 9 bytes in 1 blocks
==22957==    indirectly lost: 0 bytes in 0 blocks
==22957==      possibly lost: 0 bytes in 0 blocks
==22957==    still reachable: 0 bytes in 0 blocks
==22957==         suppressed: 0 bytes in 0 blocks
==22957==
==22957== Use --track-origins=yes to see where uninitialised values come from
==22957== For lists of detected and suppressed errors, rerun with: -s
==22957== ERROR SUMMARY: 2 errors from 2 contexts (suppressed: 0 from 0)
```

הפעם שמים לב שהבעיה נמצאת בקובץ hello.c בשורה 14:

```
14        string = malloc(length + 1);
```

יש הקצאה לזיכרון אבל אין שחרור שלו.

שאלה 6

כאשר הרצנו את הפקודה

```
valgrind --vgdb=yes --vgdb-error=0 ./hello
```

קיבלנו את רצף ההוראות הבא:

```
==23422== Memcheck, a memory error detector
==23422== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==23422== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==23422== Command: ./hello
==23422==
==23422== (action at startup) vgdb me ...
==23422==
==23422== TO DEBUG THIS PROCESS USING GDB: start GDB like this
==23422==   /path/to/gdb ./hello
==23422== and then give GDB the following command
==23422==   target remote | /usr/bin/vgdb --pid=23422
==23422== --pid is optional if only one valgrind process is running
==23422==
```

נפתח טרמינל חדש ונבצע את ההוראות, נפתח gdb:

```
matan@matan:~/IdeaProjects/OS_4/Q_5$ gdb ./hello
```

נקשר את ה- gdb ל- valgrind:

```
(gdb) target remote | /usr/bin/vgdb
Remote debugging using | /usr/bin/vgdb
relaying data between gdb and process 23422
warning: remote target does not support file transfer, attempting to access files
from local filesystem.
Reading symbols from /lib64/ld-linux-x86-64.so.2...
Reading symbols from /usr/lib/debug/.build-id/41/86944c50f8a32b47d74931e3f512b8118
13b64.debug...
0x0000000004020290 in _start () from /lib64/ld-linux-x86-64.so.2
(gdb)
```

נתחיל את הדיבוג:

```
(gdb) c
Continuing.

Program received signal SIGTRAP, Trace/breakpoint trap.
0x0000000000109234 in main (argc=1, argv=0x1ffefff8d8) at hello.c:16
16              if (string_so_far != (char *)0)
```

נשים לב שנעצרנו בשורה 16 ע"י השוואה, נרצה להדפיס את הערך של המשתנה 'string_so_far'

```
(gdb) print string_so_far
$1 = 0x0
```

השגיאה אומרת שקיבלנו הוראת שגיאה או שנתקלנו ב- brake point.

```
(gdb) c
Continuing.
[Inferior 1 (Remote target) exited normally]
(gdb)
```

נמשיך להריץ ונסיים את התוכנית, בטרמינל של ה- valgrind נקבל את הפלט הבא בסוף התוכנית:

```
You entered: ./hello
==23422==
==23422== HEAP SUMMARY:
==23422==     in use at exit: 9 bytes in 1 blocks
==23422==   total heap usage: 2 allocs, 1 frees, 1,033 bytes allocated
==23422==
==23422== LEAK SUMMARY:
==23422==    definitely lost: 9 bytes in 1 blocks
==23422==    indirectly lost: 0 bytes in 0 blocks
==23422==      possibly lost: 0 bytes in 0 blocks
==23422==    still reachable: 0 bytes in 0 blocks
==23422==         suppressed: 0 bytes in 0 blocks
==23422== Rerun with --leak-check=full to see details of leaked memory
==23422==
==23422== Use --track-origins=yes to see where uninitialised values come from
==23422== For lists of detected and suppressed errors, rerun with: -s
==23422== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
```

יש שגיאה אחת והיא זליגת הזיכרון.