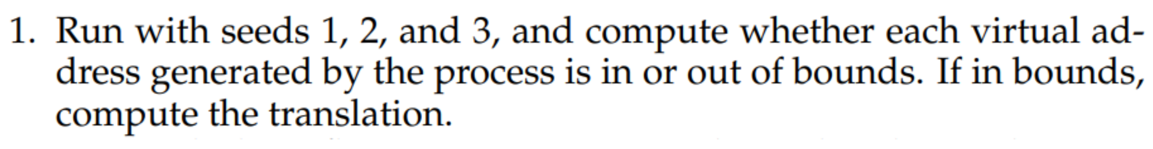
【OS】Day18(2)

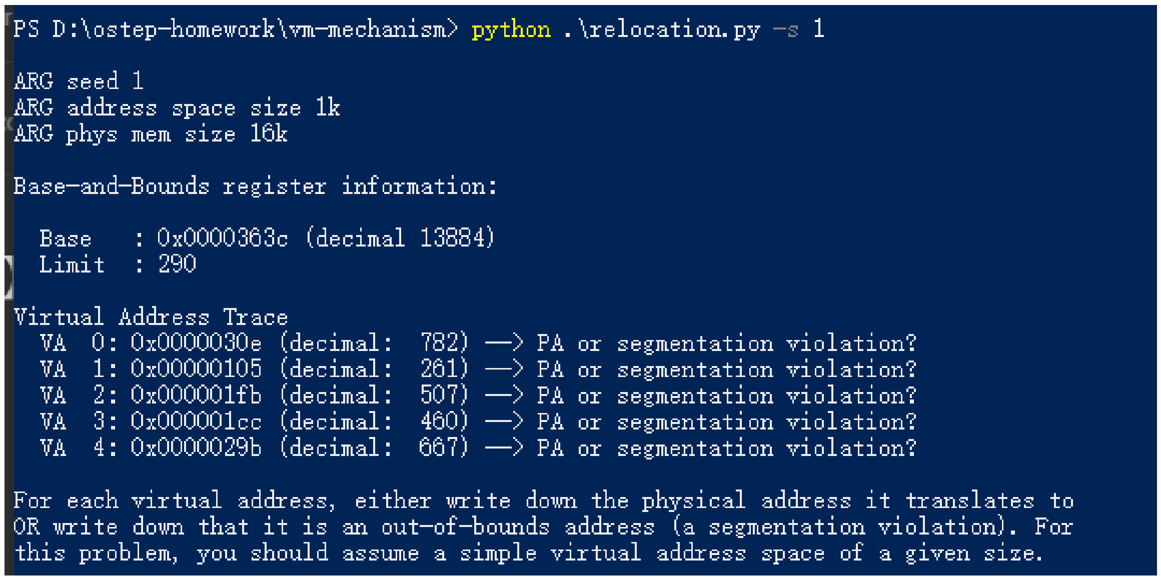
【Ch15】Address Translation Homework

Question 1



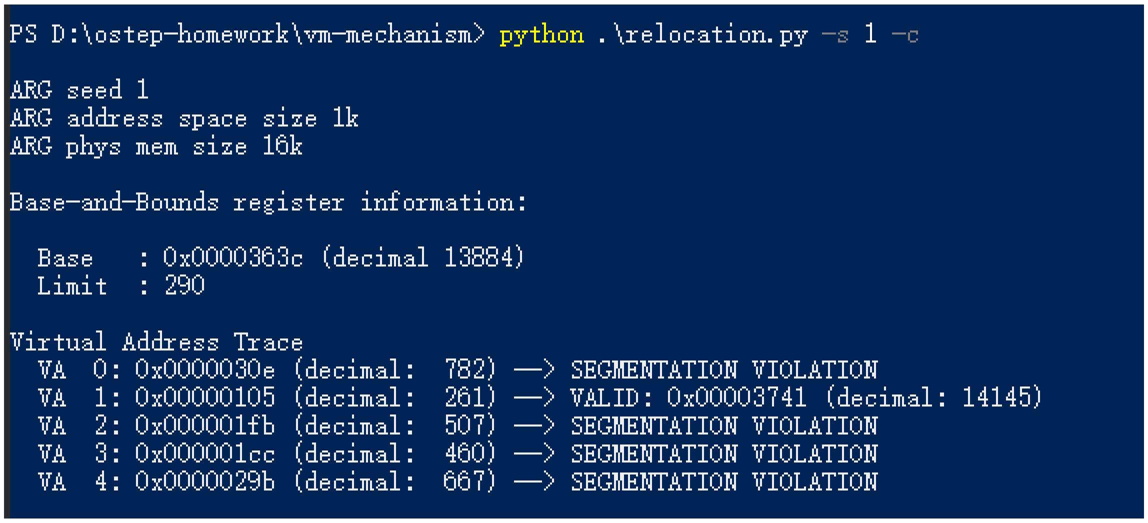
1. Run with seeds 1, 2, and 3, and compute whether each virtual address generated by the process is in or out of bounds. If in bounds, compute the translation.

-s 1:



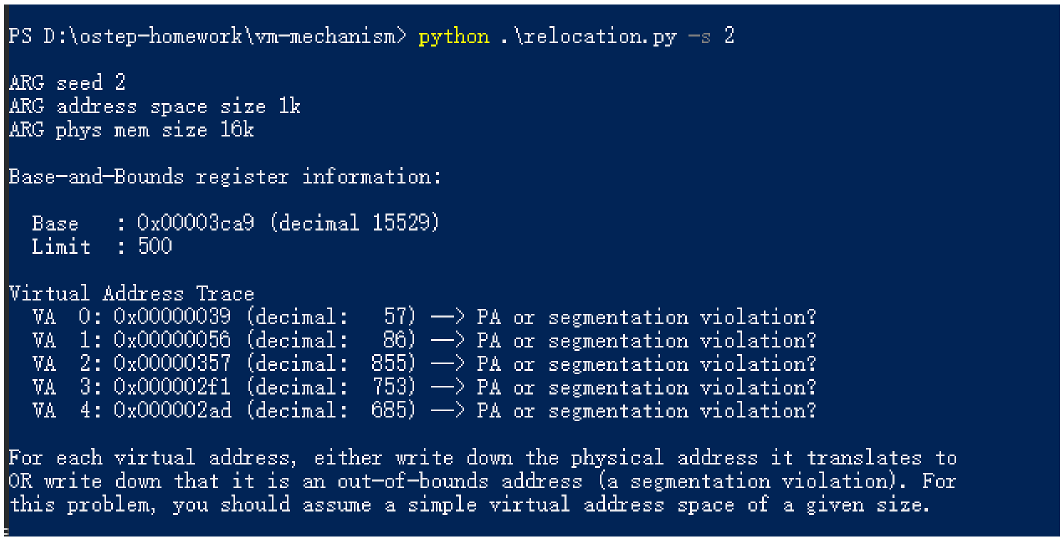
```
PS D:\ostep-homework\vm-mechanism> python .\relocation.py -s 1

ARG seed 1
ARG address space size 1k
ARG phys mem size 16k

Base-and-Bounds register information:

  Base   : 0x0000363c (decimal 13884)
  Limit  : 290

Virtual Address Trace
  VA  0: 0x0000030e (decimal:  782) --> PA or segmentation violation?
  VA  1: 0x00000105 (decimal:  261) --> PA or segmentation violation?
  VA  2: 0x000001fb (decimal:  507) --> PA or segmentation violation?
  VA  3: 0x000001cc (decimal:  460) --> PA or segmentation violation?
  VA  4: 0x0000029b (decimal:  667) --> PA or segmentation violation?

For each virtual address, either write down the physical address it translates to
OR write down that it is an out-of-bounds address (a segmentation violation). For
this problem, you should assume a simple virtual address space of a given size.
```

1. VA 0: Segmentation Violation(782 > 290)
2. VA 1: Valid(261 < 290) Address: 0x00003741
3. VA 2: Segmentation Violation

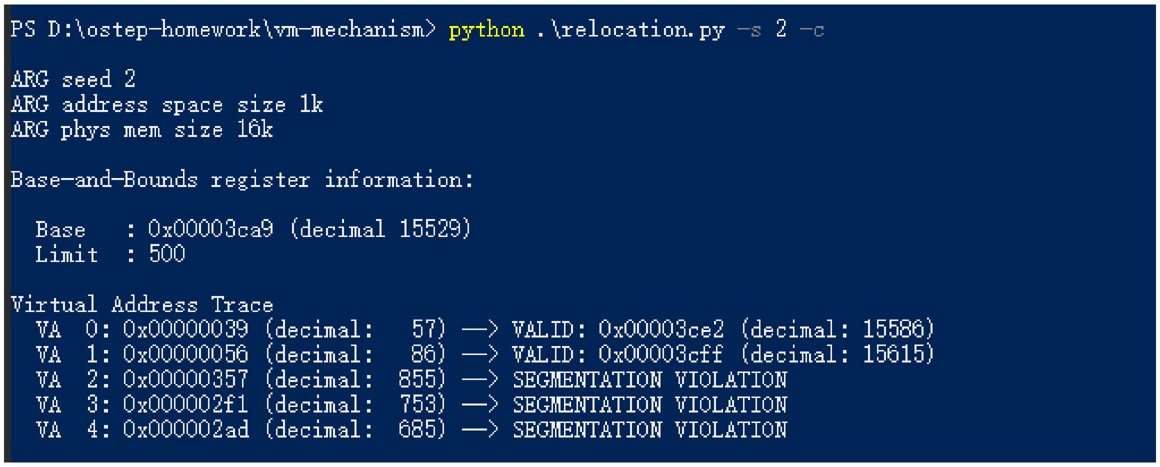


```
PS D:\ostep-homework\vm-mechanism> python .\relocation.py -s 1 -c

ARG seed 1
ARG address space size 1k
ARG phys mem size 16k

Base-and-Bounds register information:

  Base   : 0x0000363c (decimal 13884)
  Limit  : 290

Virtual Address Trace
  VA  0: 0x0000030e (decimal:  782) --> SEGMENTATION VIOLATION
  VA  1: 0x00000105 (decimal:  261) --> VALID: 0x00003741 (decimal: 14145)
  VA  2: 0x000001fb (decimal:  507) --> SEGMENTATION VIOLATION
  VA  3: 0x000001cc (decimal:  460) --> SEGMENTATION VIOLATION
  VA  4: 0x0000029b (decimal:  667) --> SEGMENTATION VIOLATION
```

-s 2:

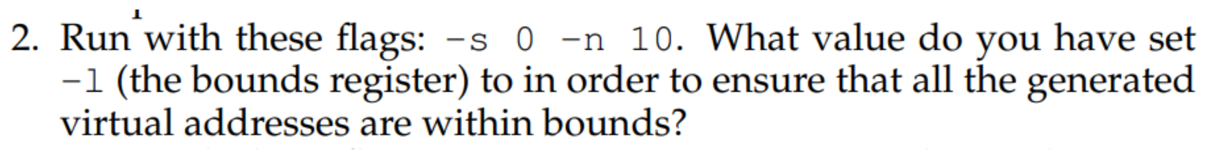


```
PS D:\ostep-homework\vm-mechanism> python .\relocation.py -s 2

ARG seed 2
ARG address space size 1k
ARG phys mem size 16k

Base-and-Bounds register information:

  Base   : 0x00003ca9 (decimal 15529)
  Limit  : 500

Virtual Address Trace
  VA  0: 0x00000039 (decimal:   57) --> PA or segmentation violation?
  VA  1: 0x00000056 (decimal:   86) --> PA or segmentation violation?
  VA  2: 0x00000357 (decimal:  855) --> PA or segmentation violation?
  VA  3: 0x000002f1 (decimal:  753) --> PA or segmentation violation?
  VA  4: 0x000002ad (decimal:  685) --> PA or segmentation violation?

For each virtual address, either write down the physical address it translates to
OR write down that it is an out-of-bounds address (a segmentation violation). For
this problem, you should assume a simple virtual address space of a given size.
```

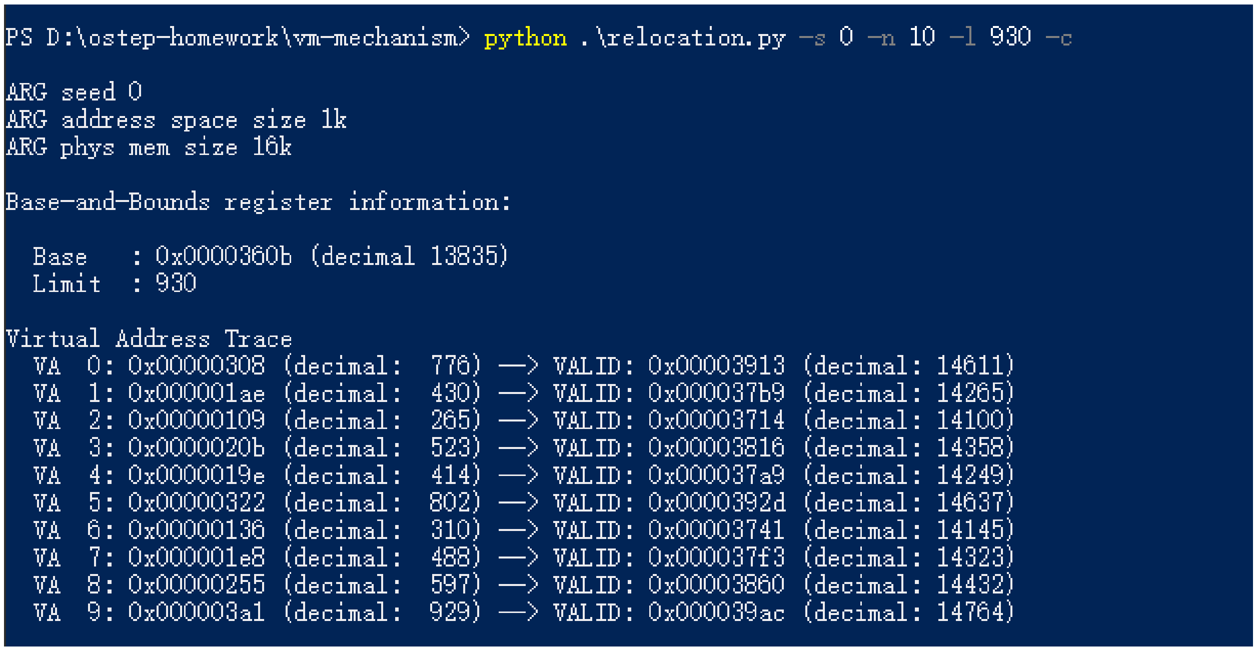


```
PS D:\ostep-homework\vm-mechanism> python .\relocation.py -s 2 -c

ARG seed 2
ARG address space size 1k
ARG phys mem size 16k

Base-and-Bounds register information:

  Base   : 0x00003ca9 (decimal 15529)
  Limit  : 500

Virtual Address Trace
  VA  0: 0x00000039 (decimal:   57) --> VALID: 0x00003ce2 (decimal: 15586)
  VA  1: 0x00000056 (decimal:   86) --> VALID: 0x00003cff (decimal: 15615)
  VA  2: 0x00000357 (decimal:  855) --> SEGMENTATION VIOLATION
  VA  3: 0x000002f1 (decimal:  753) --> SEGMENTATION VIOLATION
  VA  4: 0x000002ad (decimal:  685) --> SEGMENTATION VIOLATION
```

Question 2



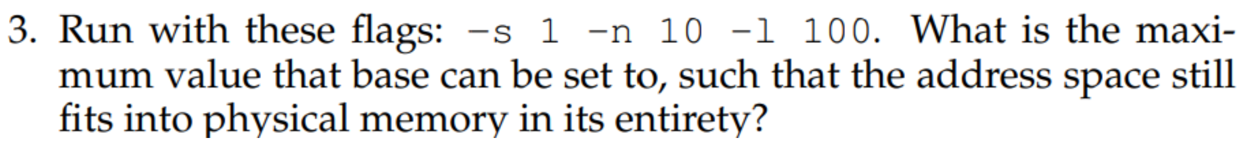
2. Run with these flags: `-s 0 -n 10`. What value do you have set `-l` (the bounds register) to in order to ensure that all the generated virtual addresses are within bounds?



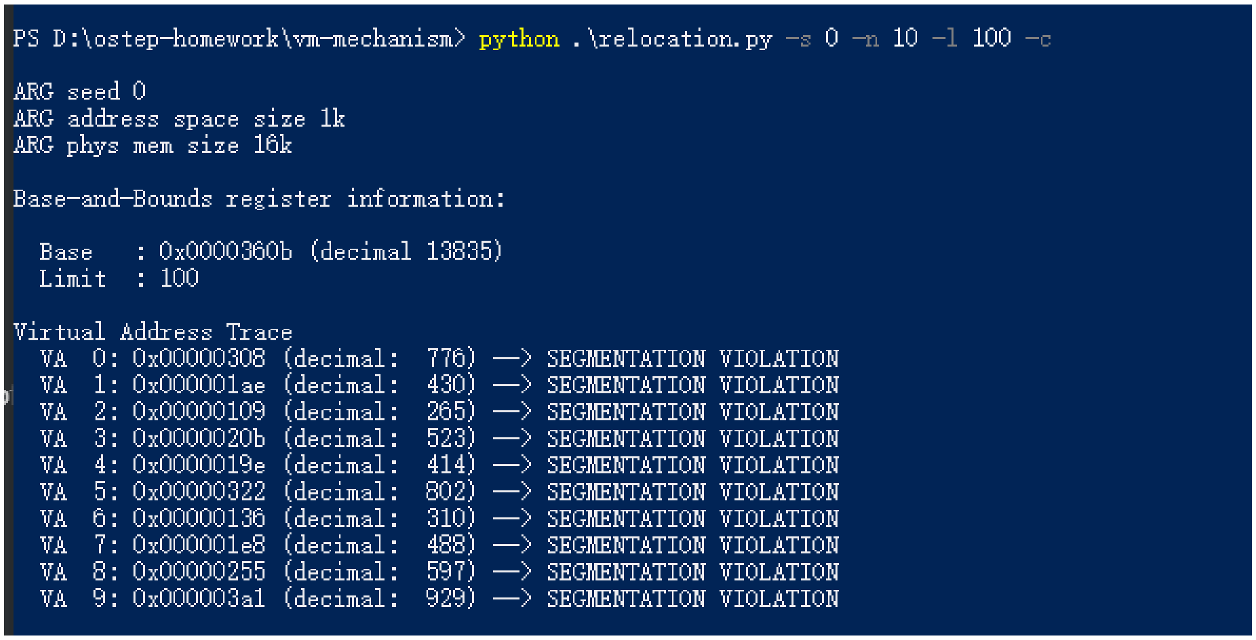
```
PS D:\ostep-homework\vm-mechanism> python .\relocation.py -s 0 -n 10 -l 930 -c

ARG seed 0
ARG address space size 1k
ARG phys mem size 16k

Base-and-Bounds register information:

  Base   : 0x0000360b (decimal 13835)
  Limit  : 930

Virtual Address Trace
  VA  0: 0x00000308 (decimal:  776) --> VALID: 0x00003913 (decimal: 14611)
  VA  1: 0x000001ae (decimal:  430) --> VALID: 0x000037b9 (decimal: 14265)
  VA  2: 0x00000109 (decimal:  265) --> VALID: 0x00003714 (decimal: 14100)
  VA  3: 0x0000020b (decimal:  523) --> VALID: 0x00003816 (decimal: 14358)
  VA  4: 0x0000019e (decimal:  414) --> VALID: 0x000037a9 (decimal: 14249)
  VA  5: 0x00000322 (decimal:  802) --> VALID: 0x0000392d (decimal: 14637)
  VA  6: 0x00000136 (decimal:  310) --> VALID: 0x00003741 (decimal: 14145)
  VA  7: 0x000001e8 (decimal:  488) --> VALID: 0x000037f3 (decimal: 14323)
  VA  8: 0x00000255 (decimal:  597) --> VALID: 0x00003860 (decimal: 14432)
  VA  9: 0x000003a1 (decimal:  929) --> VALID: 0x000039ac (decimal: 14764)
```

The value of the bounds register has to be greater than the greatest address request.

Question 3



3. Run with these flags: `-s 1 -n 10 -l 100`. What is the maximum value that base can be set to, such that the address space still fits into physical memory in its entirety?

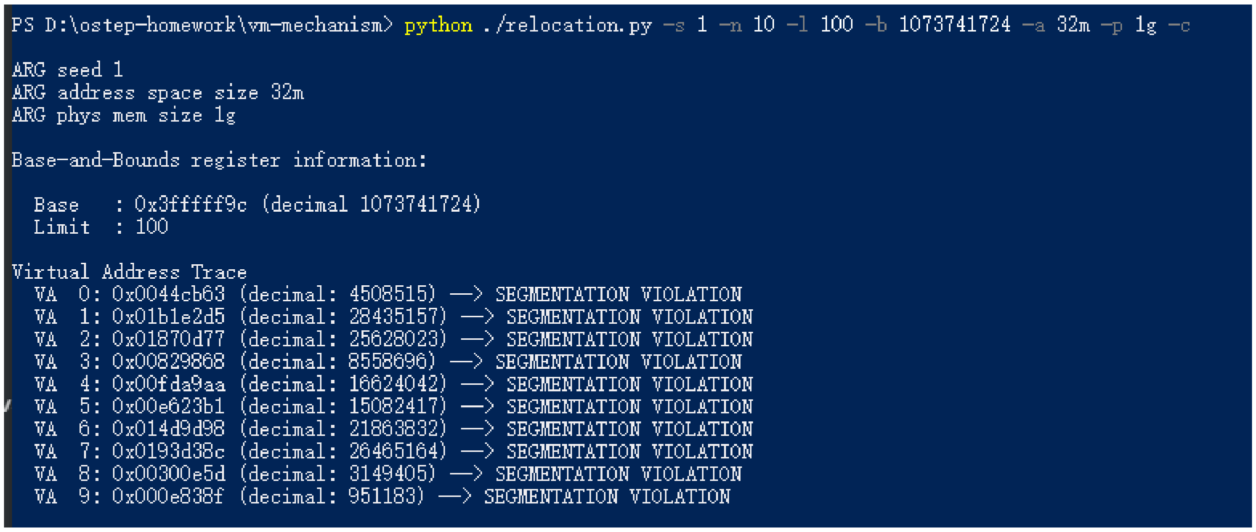


```
PS D:\ostep-homework\vm-mechanism> python .\relocation.py -s 0 -n 10 -l 100 -c

ARG seed 0
ARG address space size 1k
ARG phys mem size 16k

Base-and-Bounds register information:

  Base   : 0x0000360b (decimal 13835)
  Limit  : 100

Virtual Address Trace
  VA  0: 0x00000308 (decimal:  776) --> SEGMENTATION VIOLATION
  VA  1: 0x000001ae (decimal:  430) --> SEGMENTATION VIOLATION
  VA  2: 0x00000109 (decimal:  265) --> SEGMENTATION VIOLATION
  VA  3: 0x0000020b (decimal:  523) --> SEGMENTATION VIOLATION
  VA  4: 0x0000019e (decimal:  414) --> SEGMENTATION VIOLATION
  VA  5: 0x00000322 (decimal:  802) --> SEGMENTATION VIOLATION
  VA  6: 0x00000136 (decimal:  310) --> SEGMENTATION VIOLATION
  VA  7: 0x000001e8 (decimal:  488) --> SEGMENTATION VIOLATION
  VA  8: 0x00000255 (decimal:  597) --> SEGMENTATION VIOLATION
  VA  9: 0x000003a1 (decimal:  929) --> SEGMENTATION VIOLATION
```

Maximum value: 0x0000366F

Question 4

4. Run some of the same problems above, but with larger address spaces (-a) and physical memories (-p).



```
PS D:\ostep-homework\vm-mechanism> python ./relocation.py -s 1 -n 10 -l 100 -b 1073741724 -a 32m -p 1g -c

ARG seed 1
ARG address space size 32m
ARG phys mem size 1g

Base-and-Bounds register information:

  Base   : 0x3fffff9c (decimal 1073741724)
  Limit  : 100

Virtual Address Trace
  VA  0: 0x0044cb63 (decimal: 4508515) --> SEGMENTATION VIOLATION
  VA  1: 0x01b1e2d5 (decimal: 28435157) --> SEGMENTATION VIOLATION
  VA  2: 0x01870d77 (decimal: 25628023) --> SEGMENTATION VIOLATION
  VA  3: 0x00829868 (decimal: 8558696) --> SEGMENTATION VIOLATION
  VA  4: 0x00fda9aa (decimal: 16624042) --> SEGMENTATION VIOLATION
  VA  5: 0x00e623b1 (decimal: 15082417) --> SEGMENTATION VIOLATION
  VA  6: 0x014d9d98 (decimal: 21863832) --> SEGMENTATION VIOLATION
  VA  7: 0x0193d38c (decimal: 26465164) --> SEGMENTATION VIOLATION
  VA  8: 0x00300e5d (decimal: 3149405) --> SEGMENTATION VIOLATION
  VA  9: 0x000e838f (decimal: 951183) --> SEGMENTATION VIOLATION
```