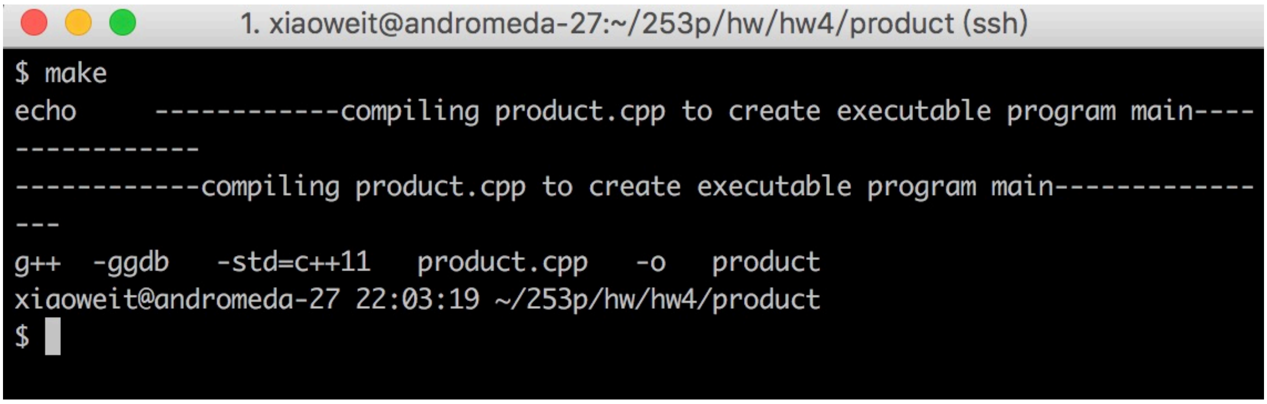
HW4 Report

Name: Xiaowei Tan

ID: 69203272

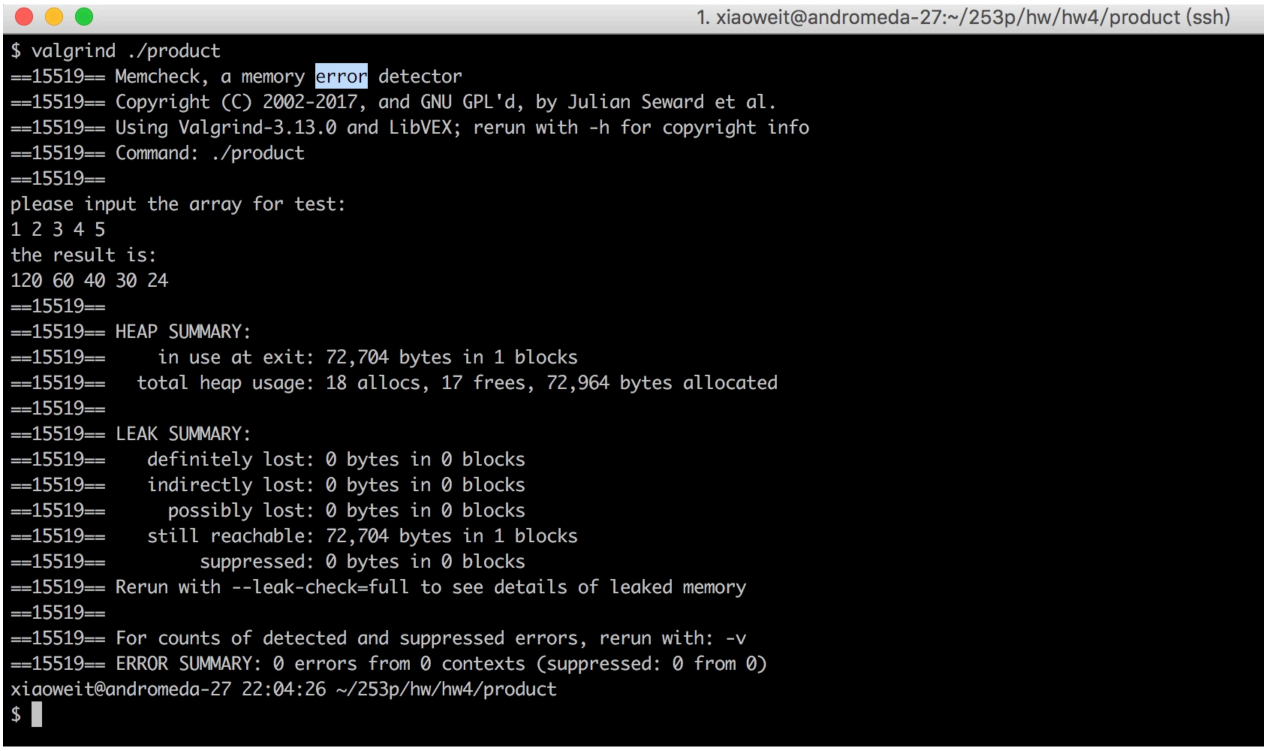
Screenshots for Lab4.1

Compile



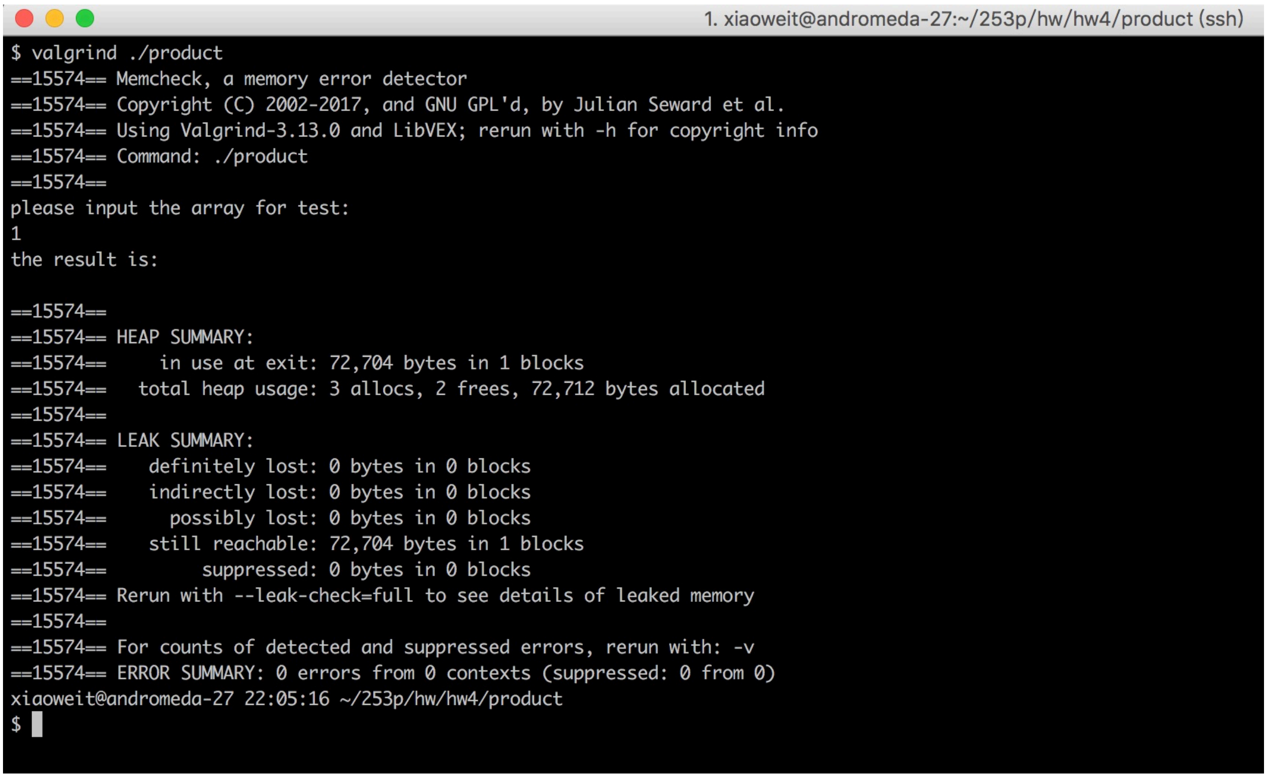
```
$ make
echo     ------------compiling product.cpp to create executable program main----
------------
------------compiling product.cpp to create executable program main-------------
---
g++  -ggdb   -std=c++11   product.cpp   -o   product
xiaoweit@andromeda-27 22:03:19 ~/253p/hw/hw4/product
$ 
```

Test1: 1 2 3 4 5



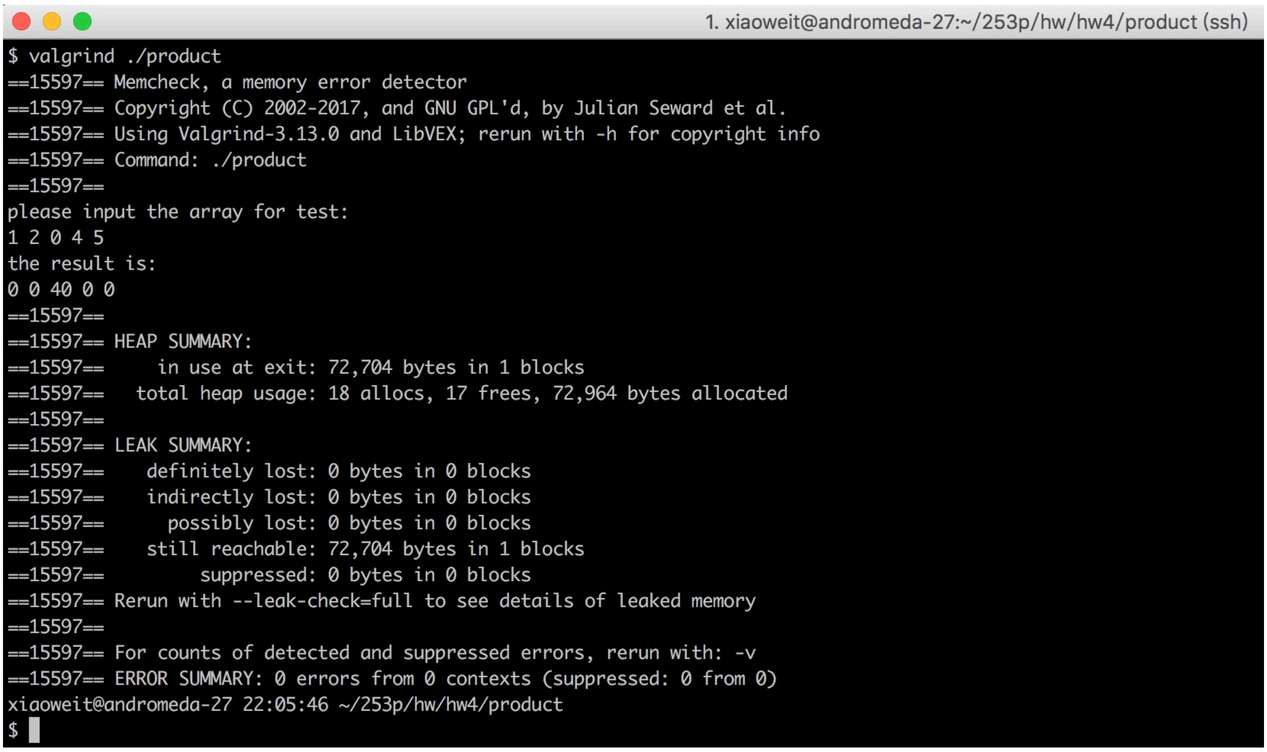
```
$ valgrind ./product
==15519== Memcheck, a memory error detector
==15519== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==15519== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==15519== Command: ./product
==15519==
please input the array for test:
1 2 3 4 5
the result is:
120 60 40 30 24
==15519==
==15519== HEAP SUMMARY:
==15519==     in use at exit: 72,704 bytes in 1 blocks
==15519==   total heap usage: 18 allocs, 17 frees, 72,964 bytes allocated
==15519==
==15519== LEAK SUMMARY:
==15519==    definitely lost: 0 bytes in 0 blocks
==15519==    indirectly lost: 0 bytes in 0 blocks
==15519==      possibly lost: 0 bytes in 0 blocks
==15519==    still reachable: 72,704 bytes in 1 blocks
==15519==         suppressed: 0 bytes in 0 blocks
==15519== Rerun with --leak-check=full to see details of leaked memory
==15519==
==15519== For counts of detected and suppressed errors, rerun with: -v
==15519== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
xiaoweit@andromeda-27 22:04:26 ~/253p/hw/hw4/product
$ 
```

Test2: single number 1



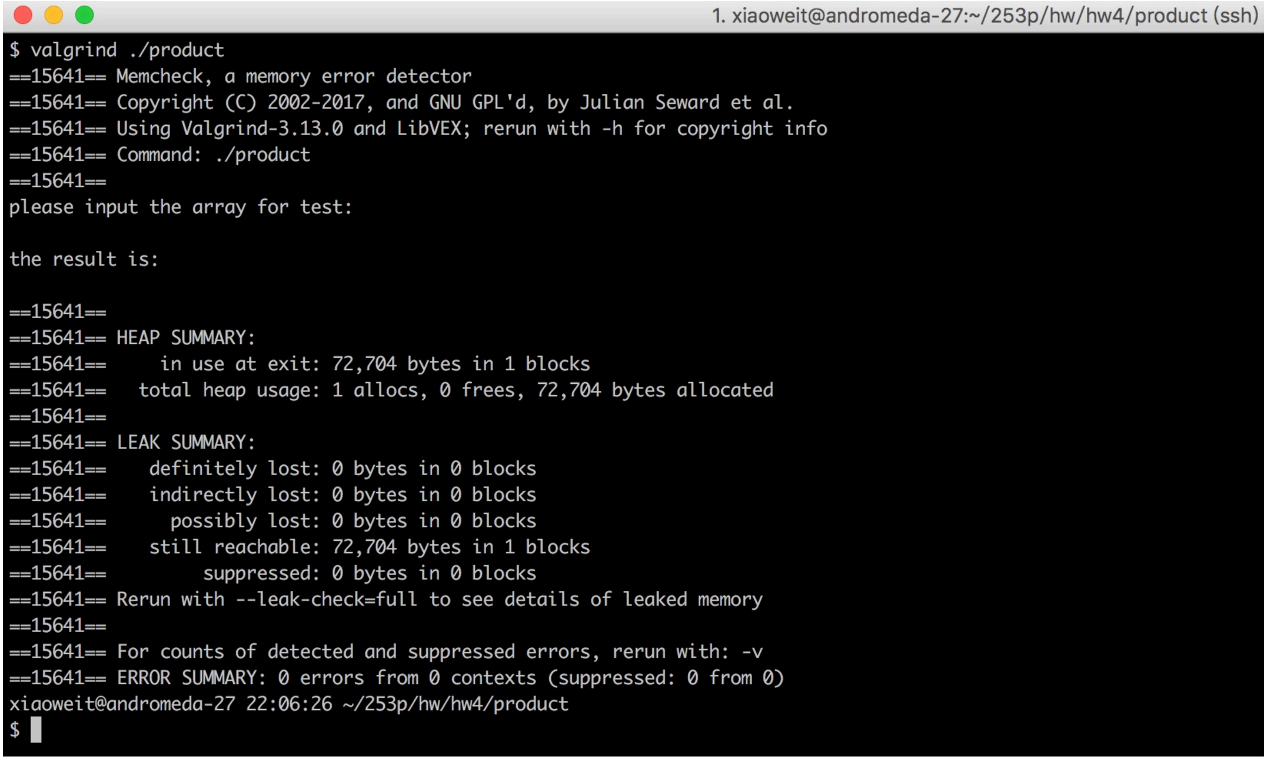
```
$ valgrind ./product
==15574== Memcheck, a memory error detector
==15574== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==15574== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==15574== Command: ./product
==15574==
please input the array for test:
1
the result is:

==15574==
==15574== HEAP SUMMARY:
==15574==     in use at exit: 72,704 bytes in 1 blocks
==15574==   total heap usage: 3 allocs, 2 frees, 72,712 bytes allocated
==15574==
==15574== LEAK SUMMARY:
==15574==    definitely lost: 0 bytes in 0 blocks
==15574==    indirectly lost: 0 bytes in 0 blocks
==15574==      possibly lost: 0 bytes in 0 blocks
==15574==    still reachable: 72,704 bytes in 1 blocks
==15574==         suppressed: 0 bytes in 0 blocks
==15574== Rerun with --leak-check=full to see details of leaked memory
==15574==
==15574== For counts of detected and suppressed errors, rerun with: -v
==15574== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
xiaoweit@andromeda-27 22:05:16 ~/253p/hw/hw4/product
$ 
```

Test3: 1 2 0 4 5



```
$ valgrind ./product
==15597== Memcheck, a memory error detector
==15597== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==15597== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==15597== Command: ./product
==15597==
please input the array for test:
1 2 0 4 5
the result is:
0 0 40 0 0
==15597==
==15597== HEAP SUMMARY:
==15597==     in use at exit: 72,704 bytes in 1 blocks
==15597==   total heap usage: 18 allocs, 17 frees, 72,964 bytes allocated
==15597==
==15597== LEAK SUMMARY:
==15597==    definitely lost: 0 bytes in 0 blocks
==15597==    indirectly lost: 0 bytes in 0 blocks
==15597==      possibly lost: 0 bytes in 0 blocks
==15597==    still reachable: 72,704 bytes in 1 blocks
==15597==         suppressed: 0 bytes in 0 blocks
==15597== Rerun with --leak-check=full to see details of leaked memory
==15597==
==15597== For counts of detected and suppressed errors, rerun with: -v
==15597== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
xiaoweit@andromeda-27 22:05:46 ~/253p/hw/hw4/product
$ 
```

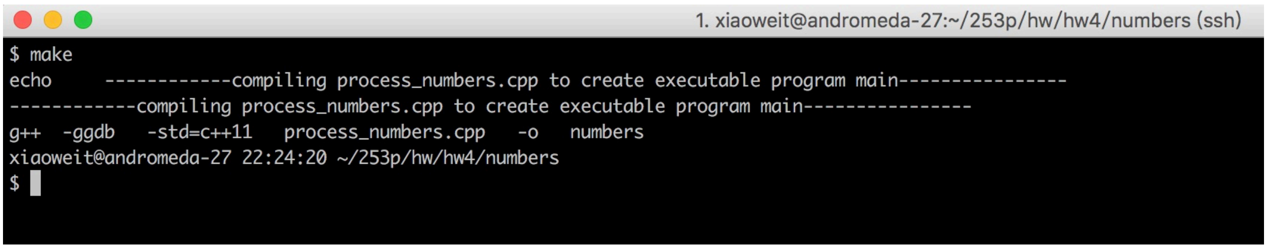
Test4: empty array



```
$ valgrind ./product
==15641== Memcheck, a memory error detector
==15641== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==15641== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==15641== Command: ./product
==15641==
please input the array for test:

the result is:

==15641==
==15641== HEAP SUMMARY:
==15641==     in use at exit: 72,704 bytes in 1 blocks
==15641==   total heap usage: 1 allocs, 0 frees, 72,704 bytes allocated
==15641==
==15641== LEAK SUMMARY:
==15641==    definitely lost: 0 bytes in 0 blocks
==15641==    indirectly lost: 0 bytes in 0 blocks
==15641==      possibly lost: 0 bytes in 0 blocks
==15641==    still reachable: 72,704 bytes in 1 blocks
==15641==         suppressed: 0 bytes in 0 blocks
==15641== Rerun with --leak-check=full to see details of leaked memory
==15641==
==15641== For counts of detected and suppressed errors, rerun with: -v
==15641== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
xiaoweit@andromeda-27 22:06:26 ~/253p/hw/hw4/product
$ 
```

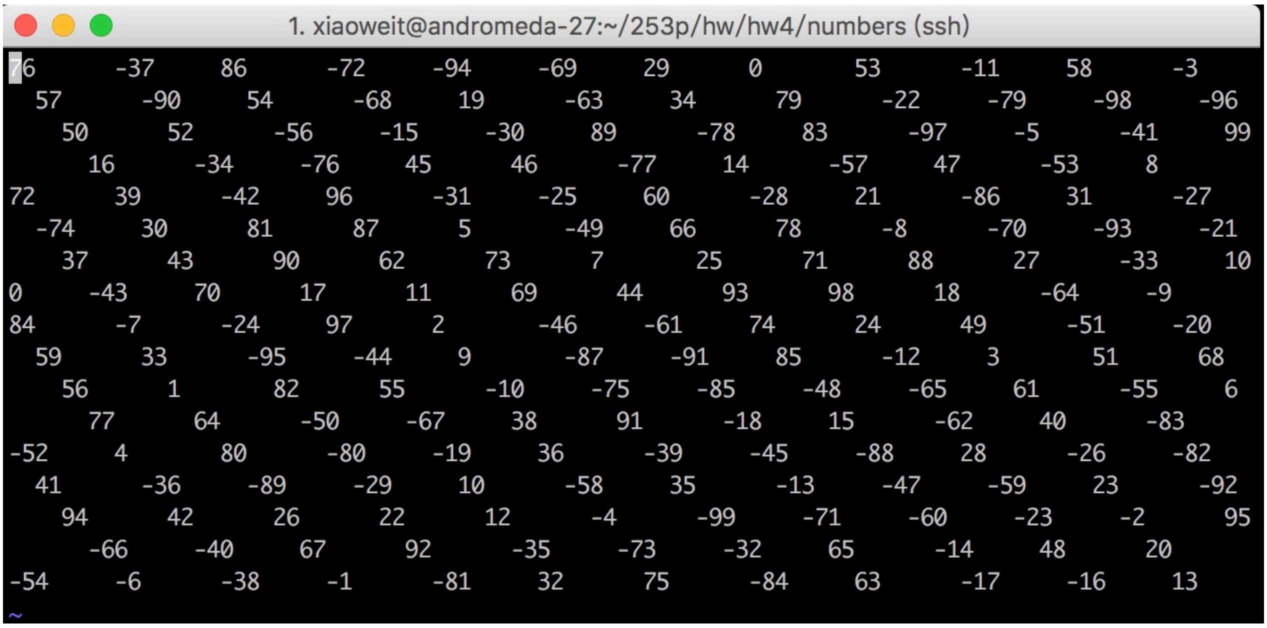
Screenshots for Hw4.1

Compile

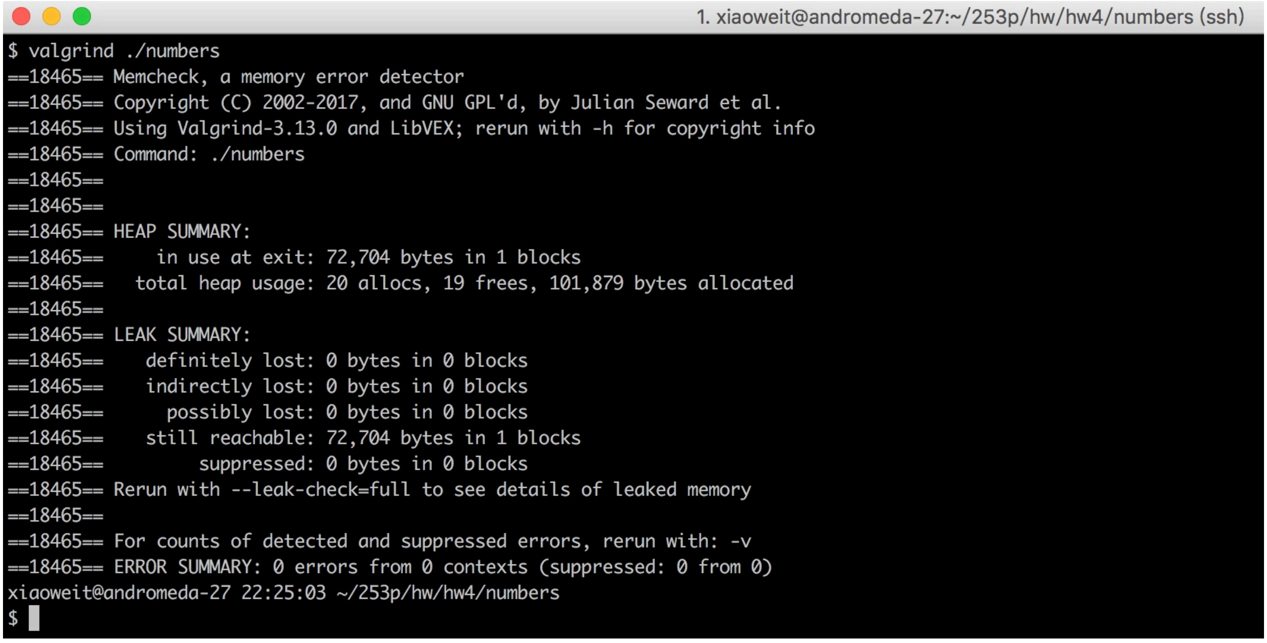


```
$ make
echo     ------------compiling process_numbers.cpp to create executable program main----------------
------------compiling process_numbers.cpp to create executable program main----------------
g++  -ggdb   -std=c++11   process_numbers.cpp   -o    numbers
xiaoweit@andromeda-27 22:24:20 ~/253p/hw/hw4/numbers
$ 
```

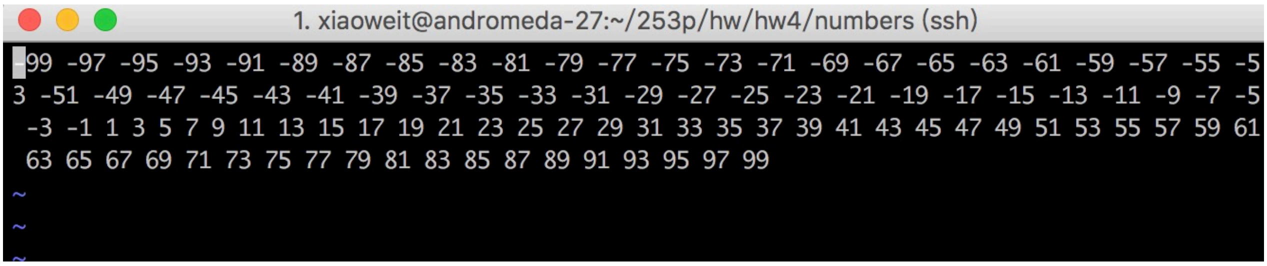
Test1: sample test



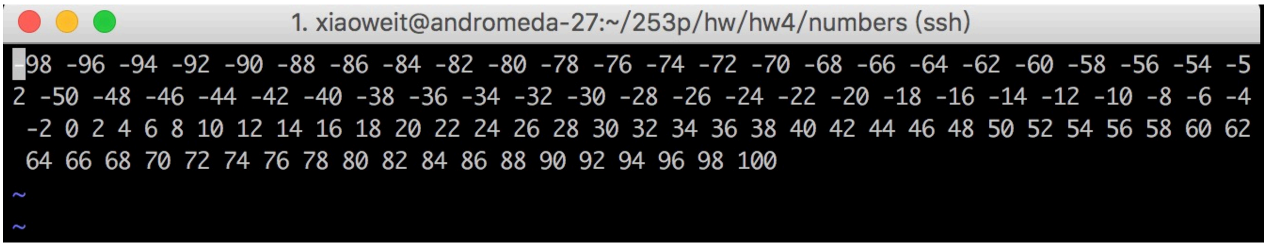
```
76      -37     86      -72     -94     -69     29      0       53      -11     58      -3
  57      -90     54      -68     19      -63     34      79      -22     -79     -98     -96
    50      52      -56     -15     -30     89      -78     83      -97     -5      -41     99
      16      -34     -76     45      46      -77     14      -57     47      -53     8
72      39      -42     96      -31     -25     60      -28     21      -86     31      -27
  -74     30      81      87      5       -49     66      78      -8      -70     -93     -21
    37      43      90      62      73      7       25      71      88      27      -33     10
0     -43     70      17      11      69      44      93      98      18      -64     -9
84      -7      -24     97      2       -46     -61     74      24      49      -51     -20
  59      33      -95     -44     9       -87     -91     85      -12     3       51      68
    56      1       82      55      -10     -75     -85     -48     -65     61      -55     6
      77      64      -50     -67     38      91      -18     15      -62     40      -83
-52     4       80      -80     -19     36      -39     -45     -88     28      -26     -82
  41      -36     -89     -29     10      -58     35      -13     -47     -59     23      -92
    94      42      26      22      12      -4      -99     -71     -60     -23     -2      95
      -66     -40     67      92      -35     -73     -32     65      -14     48      20
-54     -6      -38     -1      -81     32      75      -84     63      -17     -16     13
~
```



```
$ valgrind ./numbers
==18465== Memcheck, a memory error detector
==18465== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==18465== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==18465== Command: ./numbers
==18465==
==18465==
==18465== HEAP SUMMARY:
==18465==     in use at exit: 72,704 bytes in 1 blocks
==18465==   total heap usage: 20 allocs, 19 frees, 101,879 bytes allocated
==18465==
==18465== LEAK SUMMARY:
==18465==    definitely lost: 0 bytes in 0 blocks
==18465==    indirectly lost: 0 bytes in 0 blocks
==18465==      possibly lost: 0 bytes in 0 blocks
==18465==    still reachable: 72,704 bytes in 1 blocks
==18465==         suppressed: 0 bytes in 0 blocks
==18465== Rerun with --leak-check=full to see details of leaked memory
==18465==
==18465== For counts of detected and suppressed errors, rerun with: -v
==18465== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
xiaoweit@andromeda-27 22:25:03 ~/253p/hw/hw4/numbers
$
```

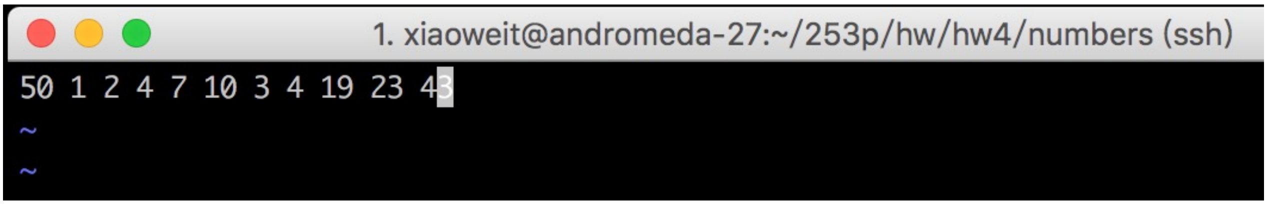


```
-99 -97 -95 -93 -91 -89 -87 -85 -83 -81 -79 -77 -75 -73 -71 -69 -67 -65 -63 -61 -59 -57 -55 -5
3 -51 -49 -47 -45 -43 -41 -39 -37 -35 -33 -31 -29 -27 -25 -23 -21 -19 -17 -15 -13 -11 -9 -7 -5
 -3 -1 1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55 57 59 61
 63 65 67 69 71 73 75 77 79 81 83 85 87 89 91 93 95 97 99
~
~
```

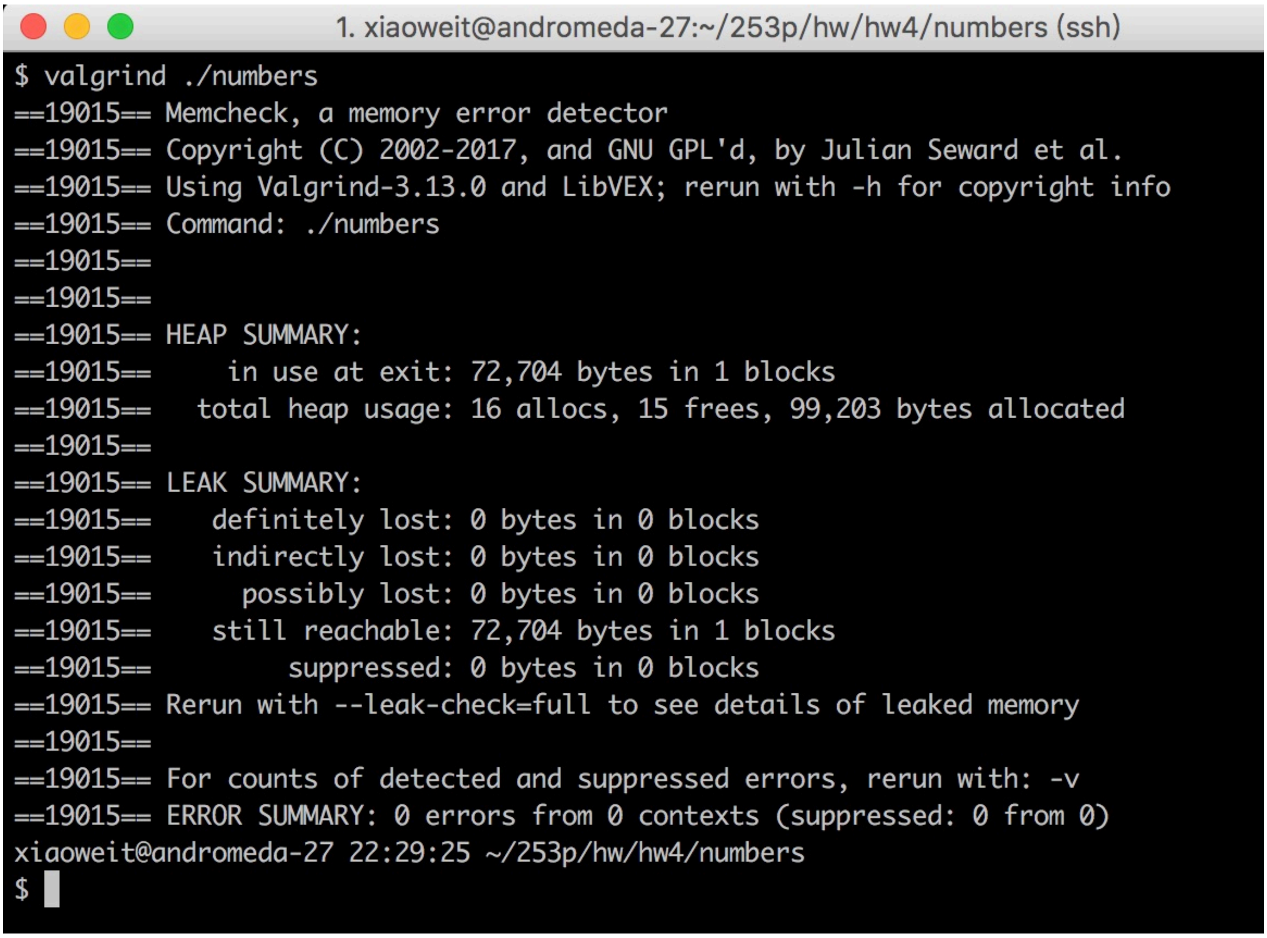


```
-98 -96 -94 -92 -90 -88 -86 -84 -82 -80 -78 -76 -74 -72 -70 -68 -66 -64 -62 -60 -58 -56 -54 -5
2 -50 -48 -46 -44 -42 -40 -38 -36 -34 -32 -30 -28 -26 -24 -22 -20 -18 -16 -14 -12 -10 -8 -6 -4
 -2 0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62
 64 66 68 70 72 74 76 78 80 82 84 86 88 90 92 94 96 98 100
~
~
```

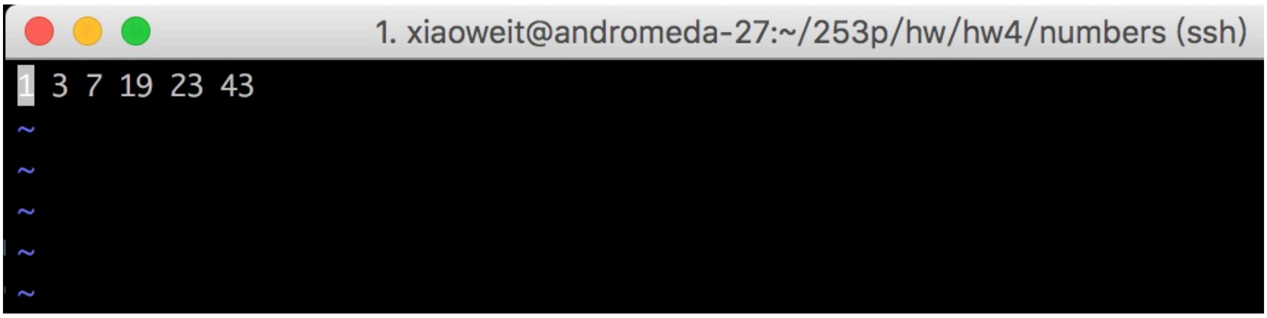
Test2: some common numbers



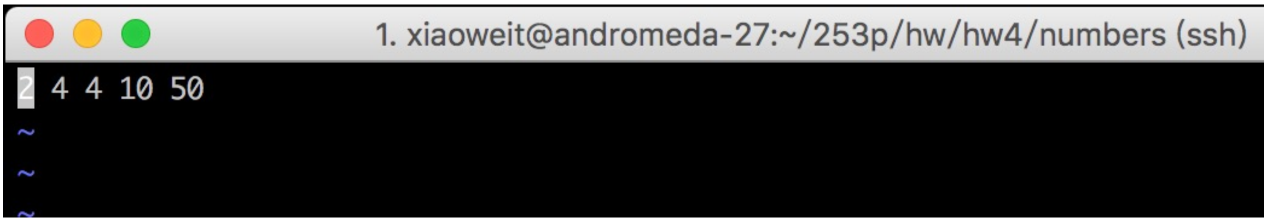
```
50 1 2 4 7 10 3 4 19 23 43
~
~
```



```
$ valgrind ./numbers
==19015== Memcheck, a memory error detector
==19015== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==19015== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==19015== Command: ./numbers
==19015==
==19015==
==19015== HEAP SUMMARY:
==19015==     in use at exit: 72,704 bytes in 1 blocks
==19015==   total heap usage: 16 allocs, 15 frees, 99,203 bytes allocated
==19015==
==19015== LEAK SUMMARY:
==19015==    definitely lost: 0 bytes in 0 blocks
==19015==    indirectly lost: 0 bytes in 0 blocks
==19015==      possibly lost: 0 bytes in 0 blocks
==19015==    still reachable: 72,704 bytes in 1 blocks
==19015==         suppressed: 0 bytes in 0 blocks
==19015== Rerun with --leak-check=full to see details of leaked memory
==19015==
==19015== For counts of detected and suppressed errors, rerun with: -v
==19015== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
xiaoweit@andromeda-27 22:29:25 ~/253p/hw/hw4/numbers
$ 
```



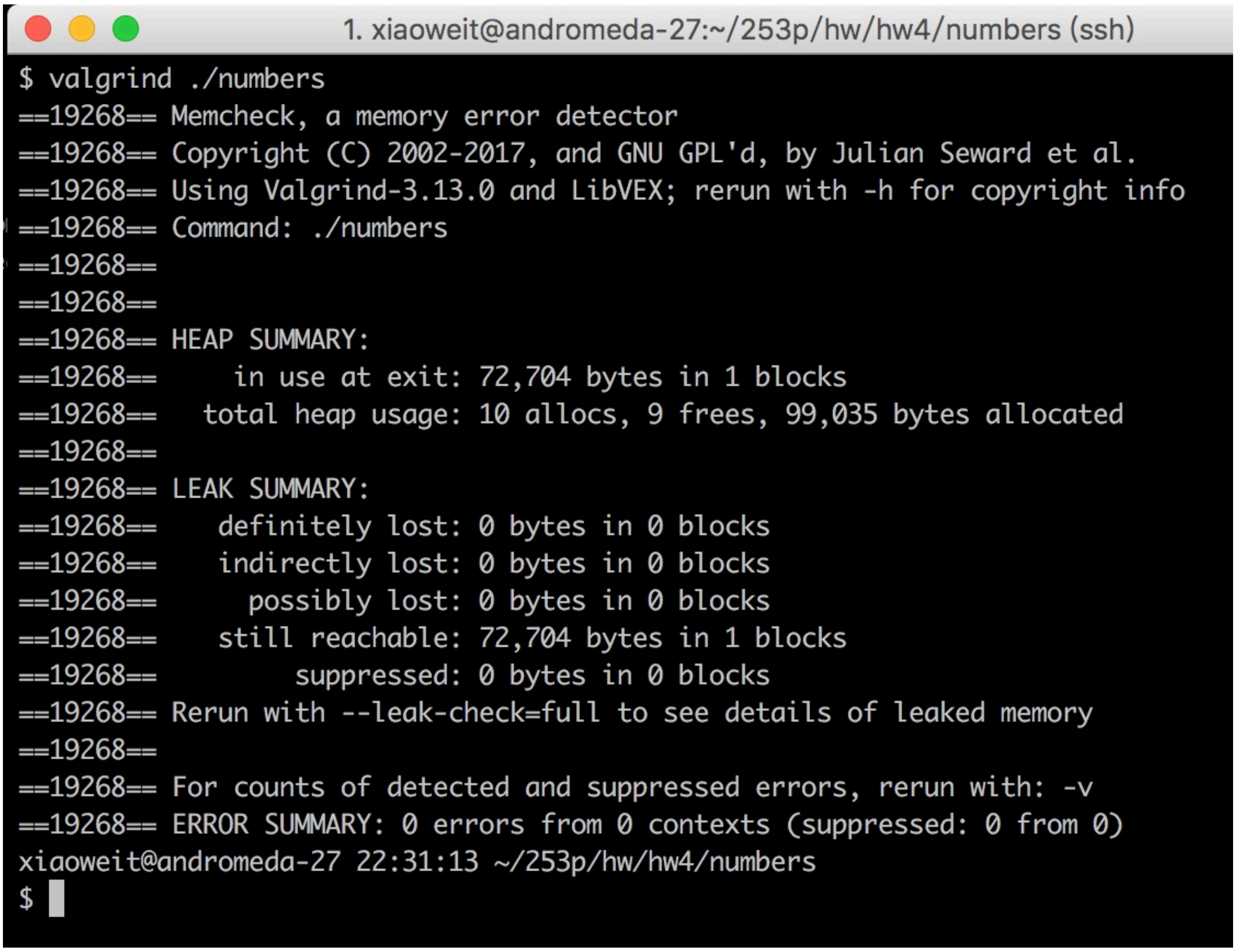
```
1 3 7 19 23 43
~
~
~
~
~
```



```
2 4 4 10 50
~
~
~
```

Test3: empty file

```
~
~
~
~
~
~
~
~
~
```

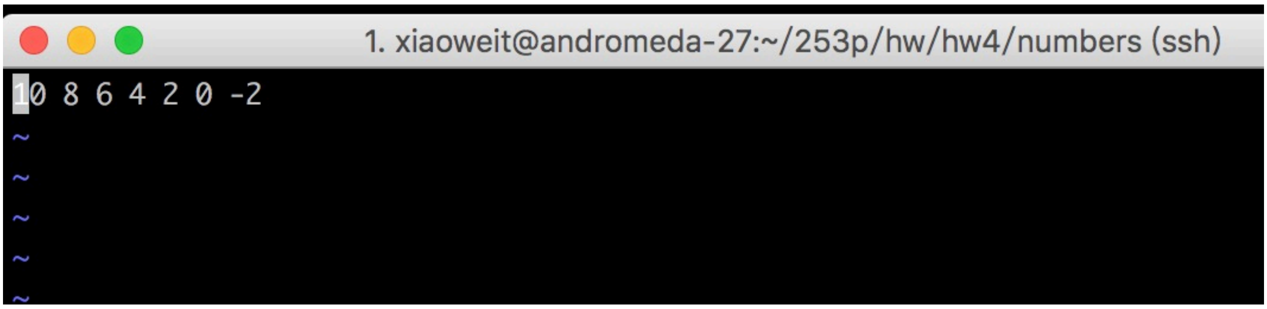


```
$ valgrind ./numbers
==19268== Memcheck, a memory error detector
==19268== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==19268== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==19268== Command: ./numbers
==19268==
==19268==
==19268== HEAP SUMMARY:
==19268==     in use at exit: 72,704 bytes in 1 blocks
==19268==   total heap usage: 10 allocs, 9 frees, 99,035 bytes allocated
==19268==
==19268== LEAK SUMMARY:
==19268==    definitely lost: 0 bytes in 0 blocks
==19268==    indirectly lost: 0 bytes in 0 blocks
==19268==      possibly lost: 0 bytes in 0 blocks
==19268==    still reachable: 72,704 bytes in 1 blocks
==19268==         suppressed: 0 bytes in 0 blocks
==19268== Rerun with --leak-check=full to see details of leaked memory
==19268==
==19268== For counts of detected and suppressed errors, rerun with: -v
==19268== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
xiaoweit@andromeda-27 22:31:13 ~/253p/hw/hw4/numbers
$
```

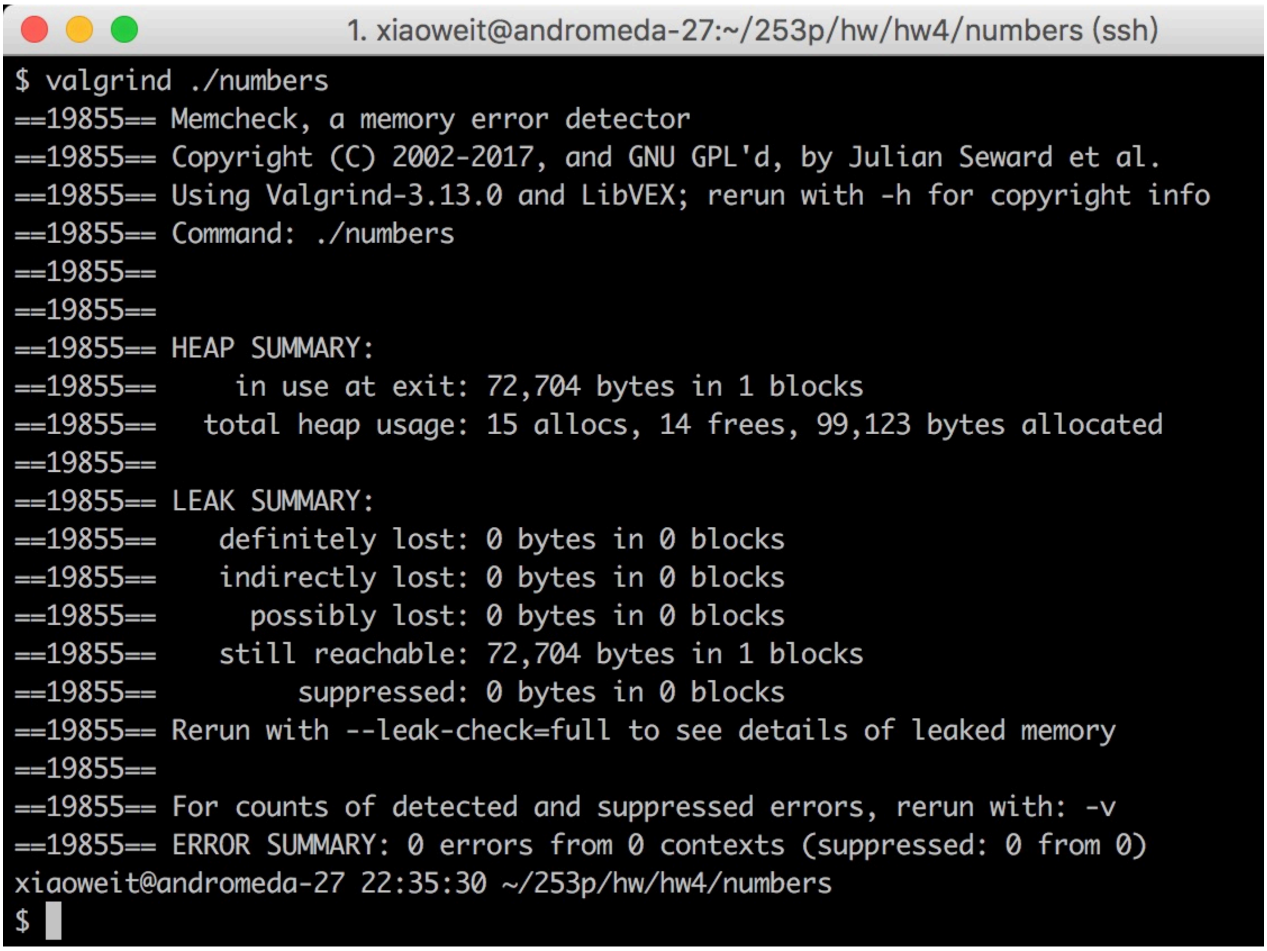
```
~
~
~
~
~
~
~
```

```

~
~
~
~
~
~
~
~
```

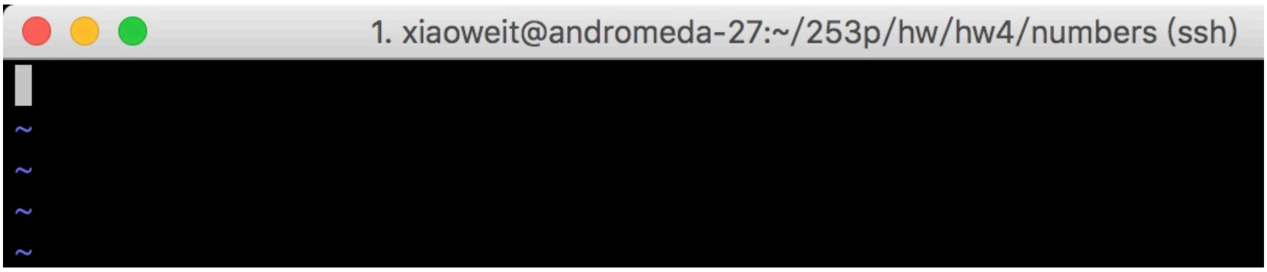
Test4: all even numbers with a negative number inside and all in decending order



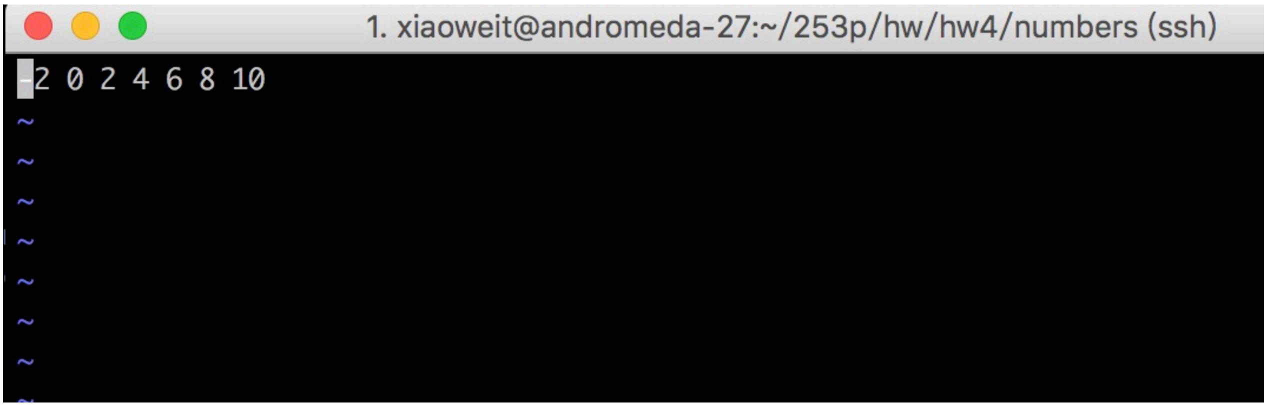
```
10 8 6 4 2 0 -2
~
~
~
~
~
```



```
$ valgrind ./numbers
==19855== Memcheck, a memory error detector
==19855== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==19855== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==19855== Command: ./numbers
==19855==
==19855==
==19855== HEAP SUMMARY:
==19855==     in use at exit: 72,704 bytes in 1 blocks
==19855==   total heap usage: 15 allocs, 14 frees, 99,123 bytes allocated
==19855==
==19855== LEAK SUMMARY:
==19855==    definitely lost: 0 bytes in 0 blocks
==19855==    indirectly lost: 0 bytes in 0 blocks
==19855==      possibly lost: 0 bytes in 0 blocks
==19855==    still reachable: 72,704 bytes in 1 blocks
==19855==         suppressed: 0 bytes in 0 blocks
==19855== Rerun with --leak-check=full to see details of leaked memory
==19855==
==19855== For counts of detected and suppressed errors, rerun with: -v
==19855== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
xiaoweit@andromeda-27 22:35:30 ~/253p/hw/hw4/numbers
$
```



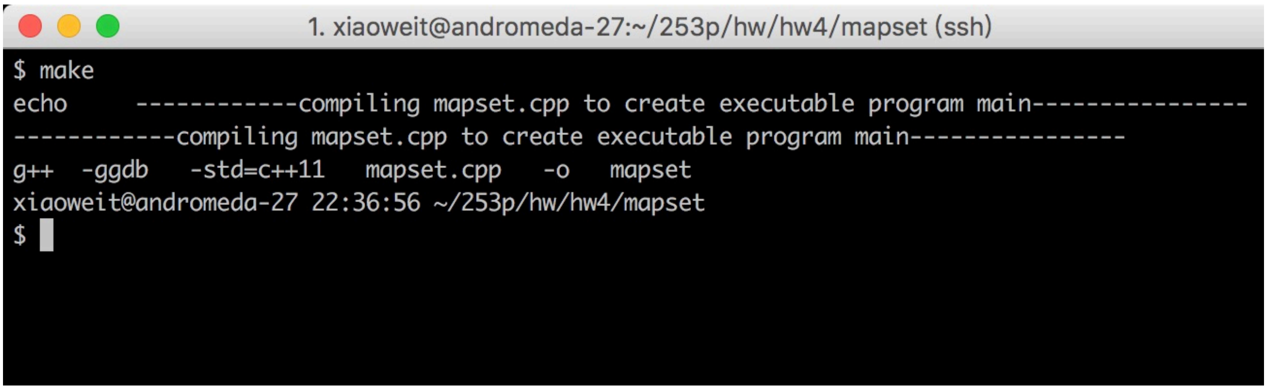
```
~
~
~
~
```



```
-2 0 2 4 6 8 10
~
~
~
~
~
~
~
```

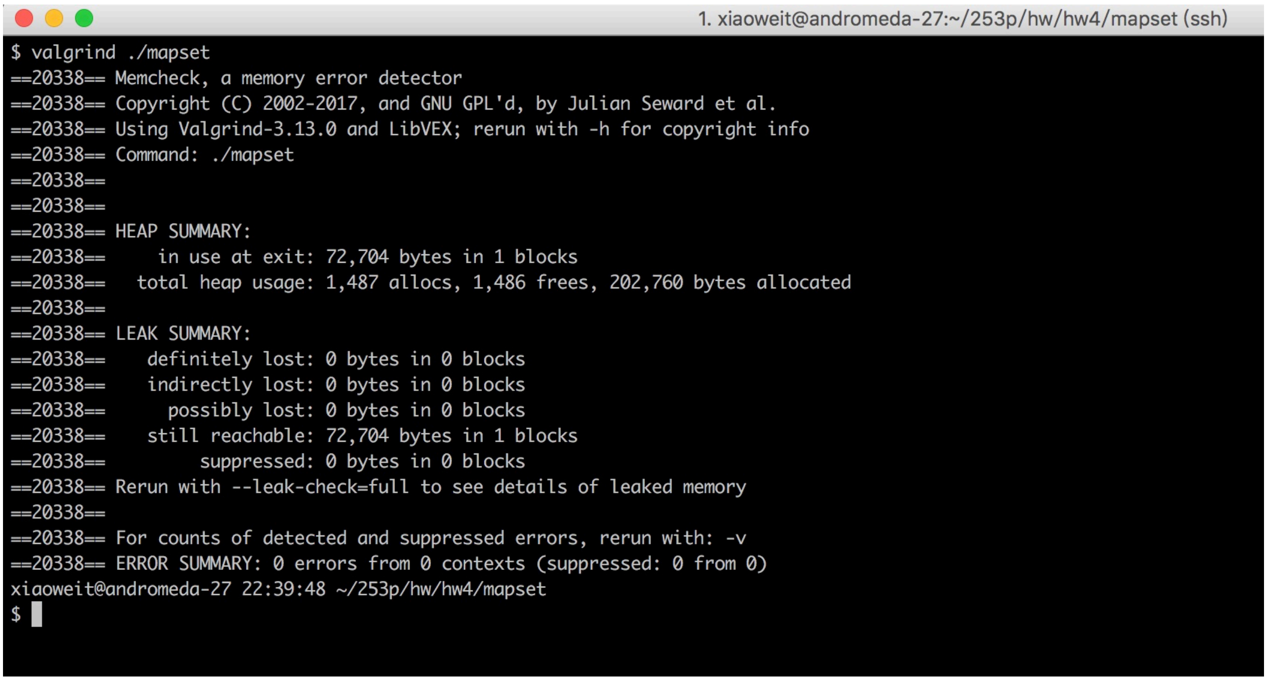
Screenshots for Hw4.2

Compile

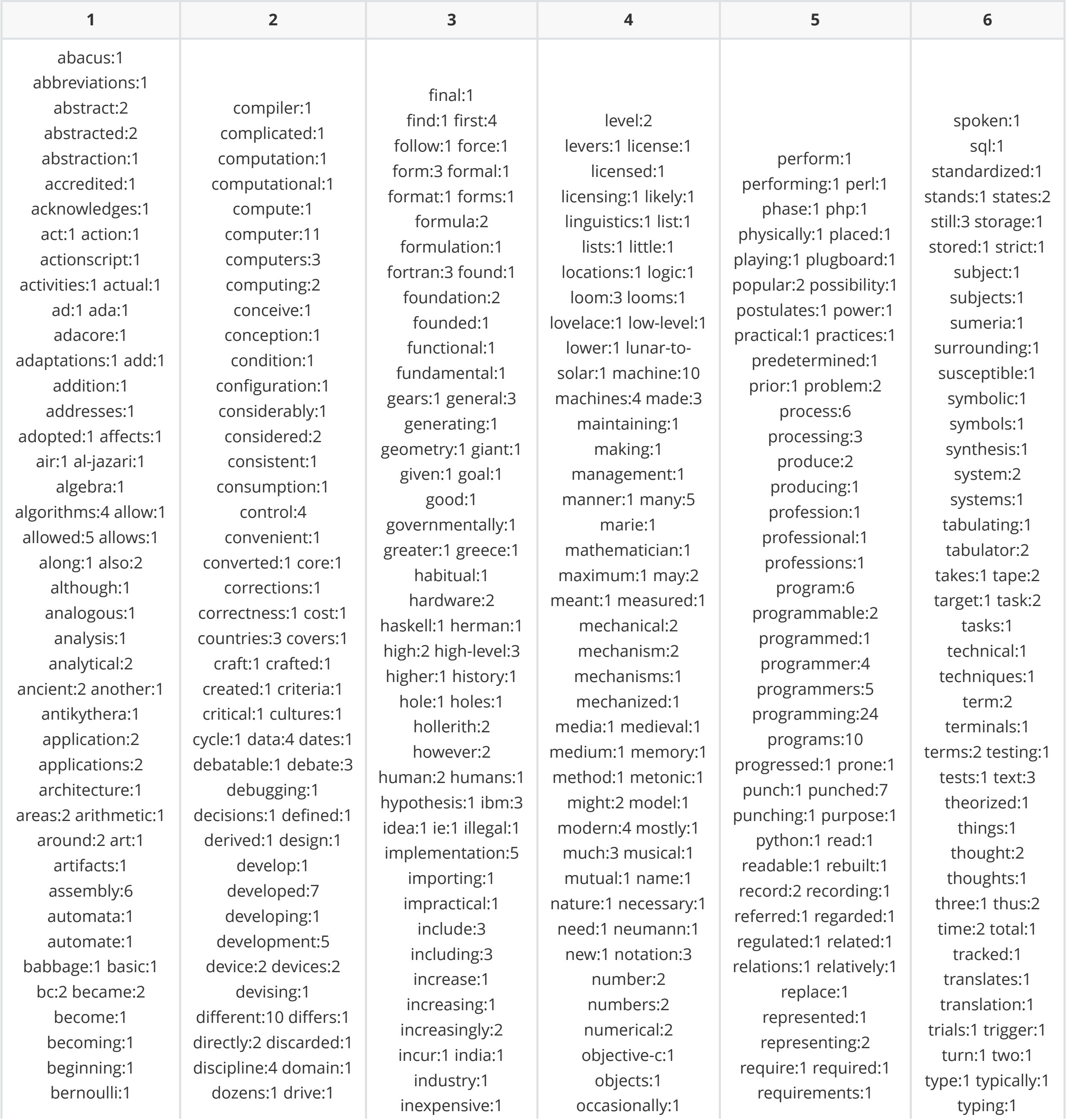


```
$ make
echo     ------------compiling mapset.cpp to create executable program main---------------
------------compiling mapset.cpp to create executable program main---------------
g++  -ggdb   -std=c++11   mapset.cpp   -o   mapset
xiaoweit@andromeda-27 22:36:56 ~/253p/hw/hw4/mapset
$ 
```

Test1: sample test



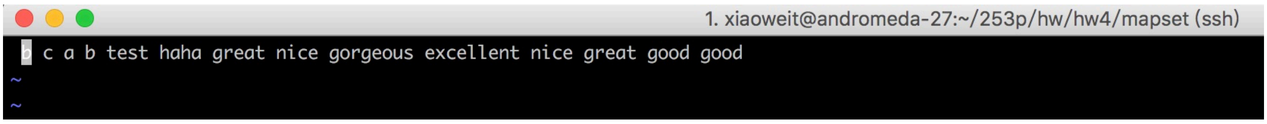
```
$ valgrind ./mapset
==20338== Memcheck, a memory error detector
==20338== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==20338== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==20338== Command: ./mapset
==20338==
==20338==
==20338== HEAP SUMMARY:
==20338==     in use at exit: 72,704 bytes in 1 blocks
==20338==   total heap usage: 1,487 allocs, 1,486 frees, 202,760 bytes allocated
==20338==
==20338== LEAK SUMMARY:
==20338==    definitely lost: 0 bytes in 0 blocks
==20338==    indirectly lost: 0 bytes in 0 blocks
==20338==      possibly lost: 0 bytes in 0 blocks
==20338==    still reachable: 72,704 bytes in 1 blocks
==20338==         suppressed: 0 bytes in 0 blocks
==20338== Rerun with --leak-check=full to see details of leaked memory
==20338==
==20338== For counts of detected and suppressed errors, rerun with: -v
==20338== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
xiaoweit@andromeda-27 22:39:48 ~/253p/hw/hw4/mapset
$ 
```



1	2	3	4	5	6
abacus:1 abbreviations:1 abstract:2 abstracted:2 abstraction:1 accredited:1 acknowledges:1 act:1 action:1 actionscript:1 activities:1 actual:1 ad:1 ada:1 adacore:1 adaptations:1 add:1 addition:1 addresses:1 adopted:1 affects:1 air:1 al-jazari:1 algebra:1 algorithms:4 allow:1 allowed:5 allows:1 along:1 also:2 although:1 analogous:1 analysis:1 analytical:2 ancient:2 another:1 antikythera:1 application:2 applications:2 architecture:1 areas:2 arithmetic:1 around:2 art:1 artifacts:1 assembly:6 automata:1 automate:1 babbage:1 basic:1 bc:2 became:2 become:1 becoming:1 beginning:1 bernoulli:1	compiler:1 complicated:1 computation:1 computational:1 compute:1 computer:11 computers:3 computing:2 conceive:1 conception:1 condition:1 configuration:1 considerably:1 considered:2 consistent:1 consumption:1 control:4 convenient:1 converted:1 core:1 corrections:1 correctness:1 cost:1 countries:3 covers:1 craft:1 crafted:1 created:1 criteria:1 critical:1 cultures:1 cycle:1 data:4 dates:1 debatable:1 debate:3 debugging:1 decisions:1 defined:1 derived:1 design:1 develop:1 developed:7 developing:1 development:5 device:2 devices:2 devising:1 different:10 differs:1 directly:2 discarded:1 discipline:4 domain:1 dozens:1 drive:1	final:1 find:1 first:4 follow:1 force:1 form:3 formal:1 format:1 forms:1 formula:2 formulation:1 fortran:3 found:1 foundation:2 founded:1 functional:1 fundamental:1 gears:1 general:3 generating:1 geometry:1 giant:1 given:1 goal:1 good:1 governmentally:1 greater:1 greece:1 habitual:1 hardware:2 haskell:1 herman:1 high:2 high-level:3 higher:1 history:1 hole:1 holes:1 hollerith:2 however:2 human:2 humans:1 hypothesis:1 ibm:3 idea:1 ie:1 illegal:1 implementation:5 importing:1 impractical:1 include:3 including:3 increase:1 increasing:1 increasingly:2 incur:1 india:1 industry:1 inexpensive:1	level:2 levers:1 license:1 licensed:1 licensing:1 likely:1 linguistics:1 list:1 lists:1 little:1 locations:1 logic:1 loom:3 looms:1 lovelace:1 low-level:1 lower:1 lunar-to-solar:1 machine:10 machines:4 made:3 maintaining:1 making:1 management:1 manner:1 many:5 marie:1 mathematician:1 maximum:1 may:2 meant:1 measured:1 mechanical:2 mechanism:2 mechanisms:1 mechanized:1 media:1 medieval:1 medium:1 memory:1 method:1 metonic:1 might:2 model:1 modern:4 mostly:1 much:3 musical:1 mutual:1 name:1 nature:1 necessary:1 need:1 neumann:1 new:1 notation:3 number:2 numbers:2 numerical:2 objective-c:1 objects:1 occasionally:1	perform:1 performing:1 perl:1 phase:1 php:1 physically:1 placed:1 playing:1 plugboard:1 popular:2 possibility:1 postulates:1 power:1 practical:1 practices:1 predetermined:1 prior:1 problem:2 process:6 processing:3 produce:2 producing:1 profession:1 professional:1 professions:1 program:6 programmable:2 programmed:1 programmer:4 programmers:5 programming:24 programs:10 progressed:1 prone:1 punch:1 punched:7 punching:1 purpose:1 python:1 read:1 readable:1 rebuilt:1 record:2 recording:1 referred:1 regarded:1 regulated:1 related:1 relations:1 relatively:1 replace:1 represented:1 representing:2 require:1 required:1 requirements:1	spoken:1 sql:1 standardized:1 stands:1 states:2 still:3 storage:1 stored:1 strict:1 subject:1 subjects:1 sumeria:1 surrounding:1 susceptible:1 symbolic:1 symbols:1 synthesis:1 system:2 systems:1 tabulating:1 tabulator:2 takes:1 tape:2 target:1 task:2 tasks:1 technical:1 techniques:1 term:2 terminals:1 terms:2 testing:1 tests:1 text:3 theorized:1 things:1 thought:2 thoughts:1 three:1 thus:2 time:2 total:1 tracked:1 translates:1 translation:1 trials:1 trigger:1 turn:1 two:1 type:1 typically:1 typing:1

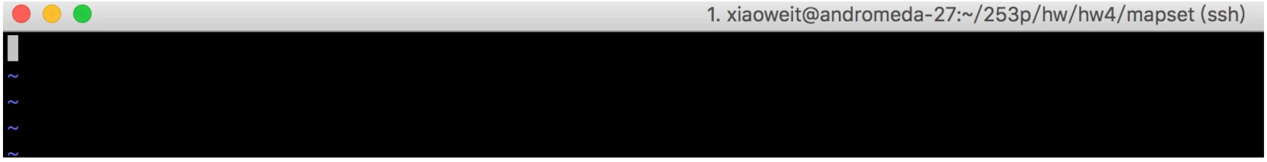
besides:1 beyond:2 binary:1 build:1 building:1 built:1 c:3 calculate:1 calculating:1 calculation:1 calculations:2 calculator:1 calculators:1 calculus:1 calendars:1 call:1 called:1 cams:1 card:3 cards:8 career:2 cases:1 certification:1 challenges:1 changes:1 charles:1 china:1 circa:1 clearance:1 cloth:1 cobol:1 code:6 coding:2 cognitive:1 combines:1 commercial:1 commonly:1 community:1 company:1	drum:2 drummer:1 due:1 early:1 easier:1 easily:1 easy:1 economic:1 editors:1 efficient:2 efficiently:1 eg:4 elementary:1 elements:1 employed:1 engine:5 engineer:1 engineering:4 engineers:1 enough:1 entered:1 entering:3 entirely:1 entities:1 environments:1 era:2 error:2 europe:1 even:1 every:1 evolvable:2 executable:1 exhaustion:1 existed:1 expertise:1 expressed:1 extent:2 fact:1 faster:1	influences:1 initial:1 input:1 instead:1 institution:1 instruction:2 instructions:7 instruments:2 intent:1 invented:5 invention:1 inventions:1 involves:1 jacquard:3 japan:1 java:1 javascript:1 jobs:1 joseph:1 just:1 keypunch:1 knowledge:1 known:2 kurdish:1 labor:1 language:16 languages:11 larger:1 late:2 later:3 leads:1 leaps:1 learn:1 led:1 less:3 let:1	often:4 olympiads:1 one:4 oneself:1 ongoing:2 operated:1 operating:1 operation:2 operations:3 opportunities:1 opposed:1 order:1 organize:1 original:2 output:2 outsourcing:1 overhead:1 painstakingly:1 panel:1 panels:1 paper:3 part:1 particular:2 particularly:2 parts:1 pass:1 past:1 pasteboard:1 pattern:2 patterns:3 pegs:1 percussion:1 perfectly:1	requires:1 reserved:1 resources:1 rhythms:1 ruby:1 s:3 sapirwhorf:1 science:2 scientist:1 security:1 see:1 seemed:1 self-governed:1 sequence:3 sequentially:1 series:1 services:1 sets:2 settled:1 several:1 shortened:1 sizes:1 small:1 smalltalk:1 software:9 solution:1 solving:1 sometimes:1 sorter:1 source:5 speaker:1 speakers:1 special:1 specialized:1 specific:2 specified:1 specify:2 specifying:1 speed:2	underlying:1 understanding:1 unit:2 united:2 upon:1 use:5 used:3 uses:2 using:5 usually:4 utilizing:1 various:2 vary:1 verification:1 visual:1 vital:1 von:1 wage:1 way:1 weaves:1 weaving:1 well:1 whether:1 whole:1 whose:2 widely-used:1 will:1 within:1 without:2 wooden:1 world:3 worlds:1 write:1 writing:3 written:1 wrote:1 x:3 y:1 yield:1

Test2: no stop words



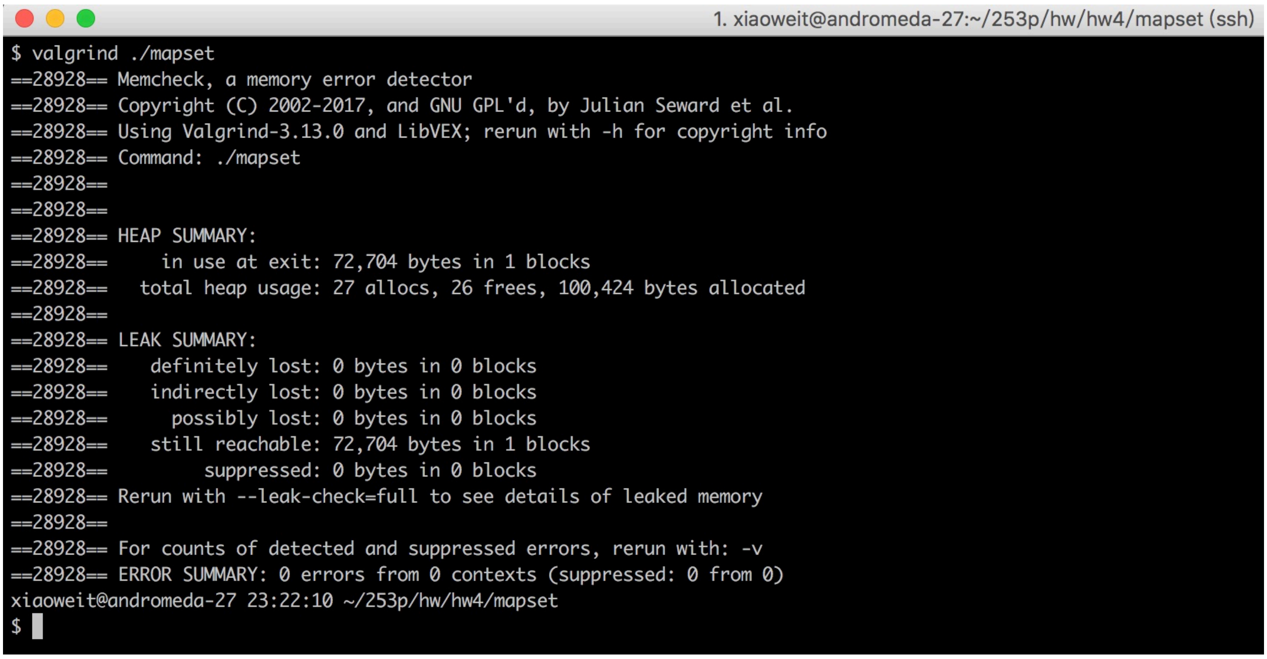
1. xiaoweit@andromeda-27:~/253p/hw/hw4/mapset (ssh)

```
b c a b test haha great nice gorgeous excellent nice great good good
~
~
```



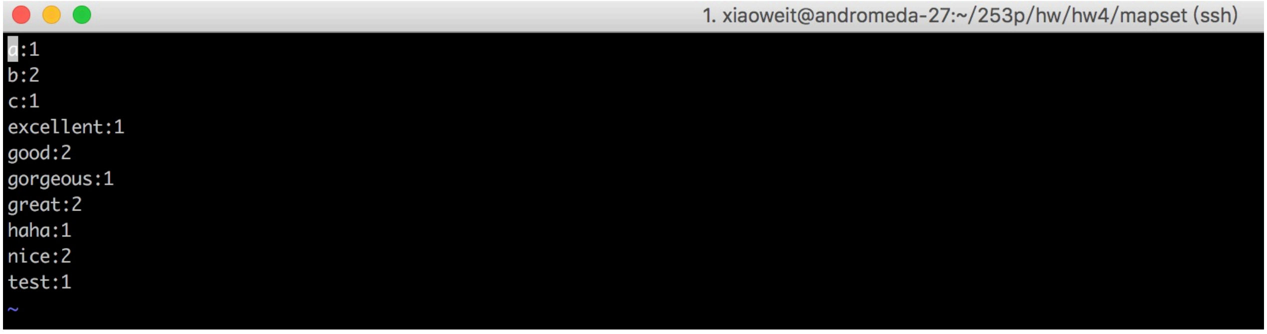
1. xiaoweit@andromeda-27:~/253p/hw/hw4/mapset (ssh)

```

~
~
~
~
```



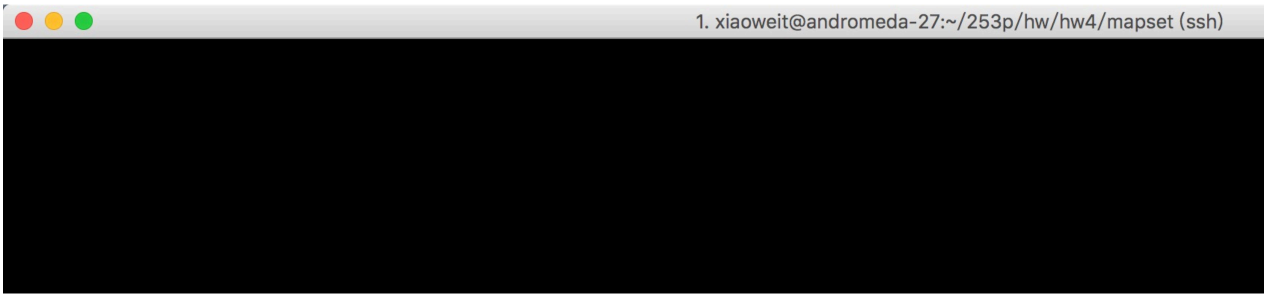
1. xiaoweit@andromeda-27:~/253p/hw/hw4/mapset (ssh)

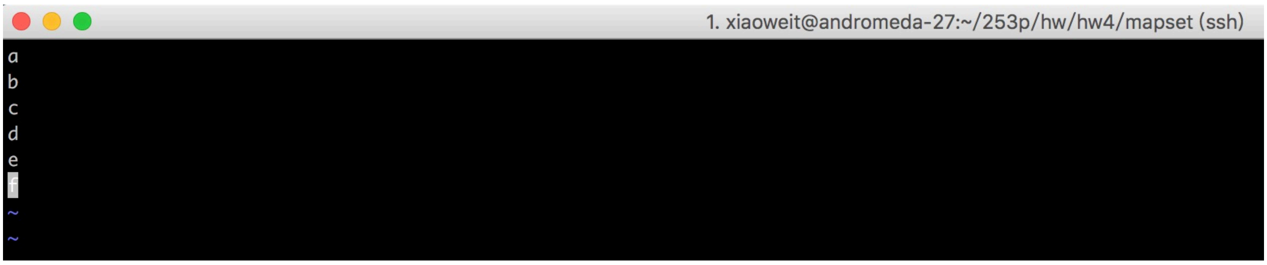
```
$ valgrind ./mapset
==28928== Memcheck, a memory error detector
==28928== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==28928== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==28928== Command: ./mapset
==28928==
==28928==
==28928== HEAP SUMMARY:
==28928==     in use at exit: 72,704 bytes in 1 blocks
==28928==   total heap usage: 27 allocs, 26 frees, 100,424 bytes allocated
==28928==
==28928== LEAK SUMMARY:
==28928==    definitely lost: 0 bytes in 0 blocks
==28928==    indirectly lost: 0 bytes in 0 blocks
==28928==      possibly lost: 0 bytes in 0 blocks
==28928==    still reachable: 72,704 bytes in 1 blocks
==28928==         suppressed: 0 bytes in 0 blocks
==28928== Rerun with --leak-check=full to see details of leaked memory
==28928==
==28928== For counts of detected and suppressed errors, rerun with: -v
==28928== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
xiaoweit@andromeda-27 23:22:10 ~/253p/hw/hw4/mapset
$
```



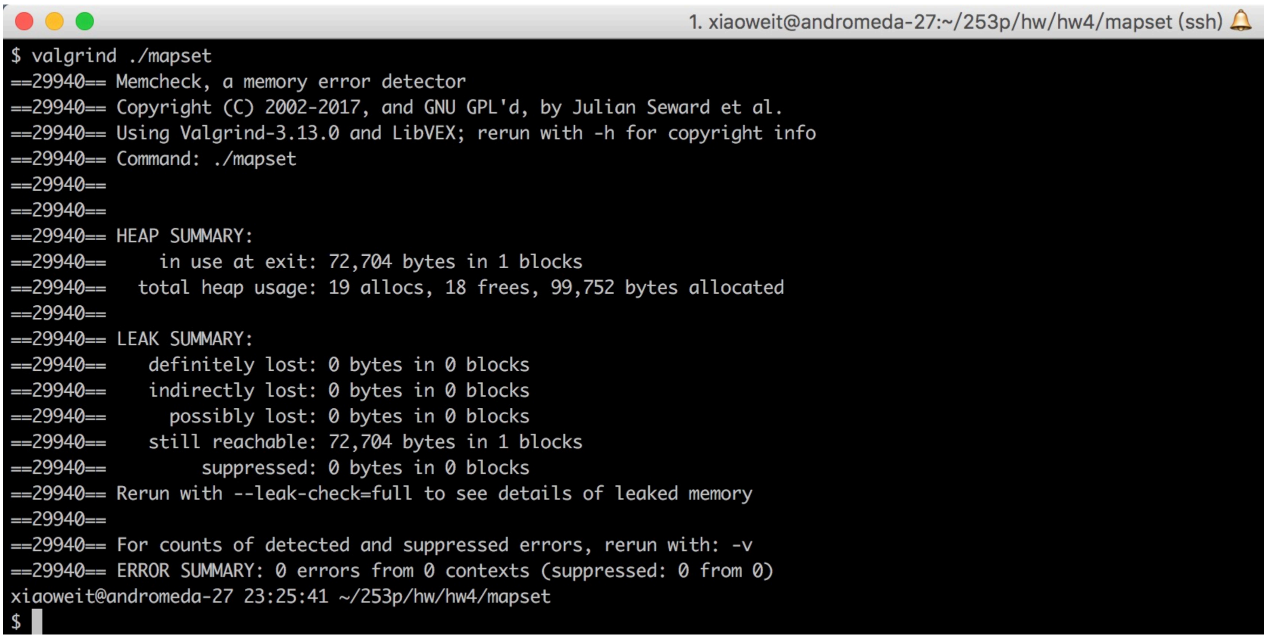
```
a:1
b:2
c:1
excellent:1
good:2
gorgeous:1
great:2
haha:1
nice:2
test:1
~
```

Test3: empty document

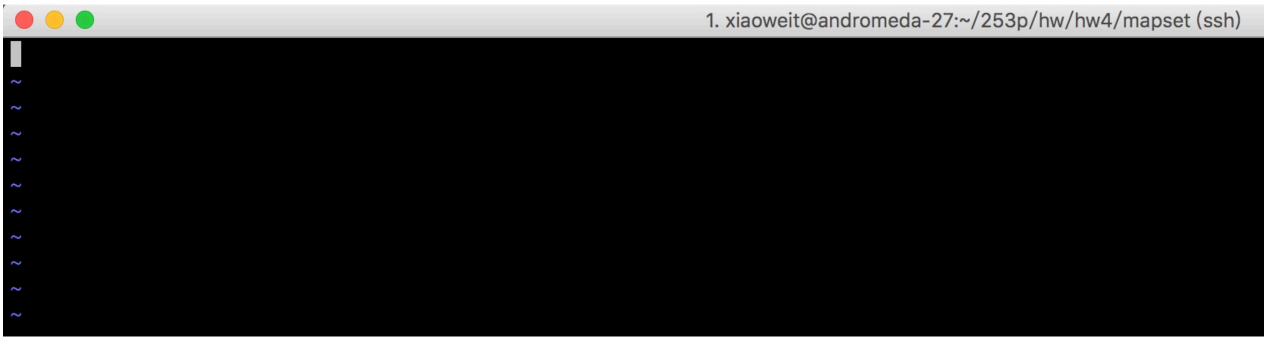




```
a
b
c
d
e
f
~
~
```



```
$ valgrind ./mapset
==29940== Memcheck, a memory error detector
==29940== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==29940== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==29940== Command: ./mapset
==29940==
==29940==
==29940== HEAP SUMMARY:
==29940==     in use at exit: 72,704 bytes in 1 blocks
==29940==   total heap usage: 19 allocs, 18 frees, 99,752 bytes allocated
==29940==
==29940== LEAK SUMMARY:
==29940==    definitely lost: 0 bytes in 0 blocks
==29940==    indirectly lost: 0 bytes in 0 blocks
==29940==      possibly lost: 0 bytes in 0 blocks
==29940==    still reachable: 72,704 bytes in 1 blocks
==29940==         suppressed: 0 bytes in 0 blocks
==29940== Rerun with --leak-check=full to see details of leaked memory
==29940==
==29940== For counts of detected and suppressed errors, rerun with: -v
==29940== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
xiaoweit@andromeda-27 23:25:41 ~/253p/hw/hw4/mapset
$ 
```



```

~
~
~
~
~
~
~
~
~
~
```