Name: Yu Qin

Results Screenshots

==========start of the write-up==========

Lab 1: Next permutation

Code:

```
13  void nextpermutation(vector<int> input){
14      int n = input.size();
15      int left = -1;
16      for (int i = n-1; i > 0; i--){
17          if (input[i-1] < input[i]){
18              left = i - 1;
19              break;
20          }
21      }
22
23      if (left == -1){
24          reverse(input.begin(), input.end());
25          cout <<"next permutation: ";
26          MyPrint(input);
27          return;
28      }
29
30      int right = 0;
31      for (int i = n-1; i > 0; i--){
32          if (input[i] > input[left]){
33              right = i;
34              break;
35          }
36      }
37      swap(input[left], input[right]);
38      reverse(input.begin()+left+1, input.end());
39      cout <<"next permutation: ";
40      MyPrint(input);
41  }
42  int main() {
43      //int a[] = {1,2,3};
44      //int a[] = {3,2,1};
45      //int a[] = {1,1,5};
46      int a[] = {5,6,3,2,2};
47      int len = sizeof(a)/sizeof(a[0]);
48      vector<int> input(a, a+len);
49      cout <<"this permutation: ";
50      MyPrint(input);
51      nextpermutation(input);
52      return 0;
53  }
```

test case 1, 2 , 3, 4:

```
$ valgrind ./main
==9152== Memcheck, a memory error detector
==9152== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==9152== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==9152== Command: ./main
==9152==
this permutation: 56322
next permutation: 62235
==9152==
==9152== HEAP SUMMARY:
==9152==     in use at exit: 72,704 bytes in 1 blocks
==9152==   total heap usage: 5 allocs, 4 frees, 72,784 bytes allocated
==9152==
==9152== LEAK SUMMARY:
==9152==    definitely lost: 0 bytes in 0 blocks
==9152==    indirectly lost: 0 bytes in 0 blocks
==9152==      possibly lost: 0 bytes in 0 blocks
==9152==    still reachable: 72,704 bytes in 1 blocks
==9152==         suppressed: 0 bytes in 0 blocks
==9152== Rerun with --leak-check=full to see details of leaked memory
==9152==
==9152== For counts of detected and suppressed errors, rerun with: -v
==9152== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
(base) yuq8@rip-riley 14:20:31 ~/253P/hw_lab/9/1_nextPermutation
$
```

2:

```
$ vi main.cpp
(base) yuq8@rip-riley 14:21:16 ~/253P/hw_lab/9/1_nextPermutation
$ make
------------compiling main.cpp to create executable program main----------------
g++  -ggdb   -std=c++11   main.cpp   -o   main
------------Congratulation to you! Successfully compile.
------------Run manually by :
------------./main
(base) yuq8@rip-riley 14:21:20 ~/253P/hw_lab/9/1_nextPermutation
$ valgrind ./main
==9230== Memcheck, a memory error detector
==9230== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==9230== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==9230== Command: ./main
==9230==
this permutation: 115
next permutation: 151
==9230==
==9230== HEAP SUMMARY:
==9230==     in use at exit: 72,704 bytes in 1 blocks
==9230==   total heap usage: 5 allocs, 4 frees, 72,752 bytes allocated
==9230==
==9230== LEAK SUMMARY:
==9230==    definitely lost: 0 bytes in 0 blocks
==9230==    indirectly lost: 0 bytes in 0 blocks
==9230==      possibly lost: 0 bytes in 0 blocks
==9230==    still reachable: 72,704 bytes in 1 blocks
==9230==         suppressed: 0 bytes in 0 blocks
==9230== Rerun with --leak-check=full to see details of leaked memory
==9230==
==9230== For counts of detected and suppressed errors, rerun with: -v
==9230== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
(base) yuq8@rip-riley 14:21:24 ~/253P/hw_lab/9/1_nextPermutation
$
```

3:

```
$ vi main.cpp
(base) yuq8@rip-riley 14:21:55 ~/253P/hw_lab/9/1_nextPermutation
$ make
------------compiling main.cpp to create executable program main----------------
g++  -ggdb   -std=c++11   main.cpp   -o   main
------------Congratulation to you! Successfully compile.
------------Run manually by :
------------./main
(base) yuq8@rip-riley 14:21:59 ~/253P/hw_lab/9/1_nextPermutation
$ valgrind ./main
==9273== Memcheck, a memory error detector
==9273== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==9273== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==9273== Command: ./main
==9273==
this permutation: 321
next permutation: 123
==9273==
==9273== HEAP SUMMARY:
==9273==     in use at exit: 72,704 bytes in 1 blocks
==9273==   total heap usage: 5 allocs, 4 frees, 72,752 bytes allocated
==9273==
==9273== LEAK SUMMARY:
==9273==    definitely lost: 0 bytes in 0 blocks
==9273==    indirectly lost: 0 bytes in 0 blocks
==9273==      possibly lost: 0 bytes in 0 blocks
==9273==    still reachable: 72,704 bytes in 1 blocks
==9273==         suppressed: 0 bytes in 0 blocks
==9273== Rerun with --leak-check=full to see details of leaked memory
==9273==
==9273== For counts of detected and suppressed errors, rerun with: -v
==9273== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
(base) yuq8@rip-riley 14:22:02 ~/253P/hw_lab/9/1_nextPermutation
$
```

4:

```
$ vi main.cpp
(base) yuq8@rip-riley 14:22:33 ~/253P/hw_lab/9/1_nextPermutation
$ make
------------compiling main.cpp to create executable program main----------------
g++  -ggdb   -std=c++11   main.cpp   -o   main
------------Congratulation to you! Successfully compile.
------------Run manually by :
------------./main
(base) yuq8@rip-riley 14:22:37 ~/253P/hw_lab/9/1_nextPermutation
$ valgrind ./main
==9318== Memcheck, a memory error detector
==9318== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==9318== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==9318== Command: ./main
==9318==
this permutation: 123
next permutation: 132
==9318==
==9318== HEAP SUMMARY:
==9318==     in use at exit: 72,704 bytes in 1 blocks
==9318==   total heap usage: 5 allocs, 4 frees, 72,752 bytes allocated
==9318==
==9318== LEAK SUMMARY:
==9318==    definitely lost: 0 bytes in 0 blocks
==9318==    indirectly lost: 0 bytes in 0 blocks
==9318==      possibly lost: 0 bytes in 0 blocks
==9318==    still reachable: 72,704 bytes in 1 blocks
==9318==         suppressed: 0 bytes in 0 blocks
==9318== Rerun with --leak-check=full to see details of leaked memory
==9318==
==9318== For counts of detected and suppressed errors, rerun with: -v
==9318== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
(base) yuq8@rip-riley 14:22:40 ~/253P/hw_lab/9/1_nextPermutation
$
```

Conclusion: right result, 0 error in memory leak.

Lab 2: spiral Matrix

Test cases 1, 2, 3, 4 :

```
$ valgrind ./main
==9617== Memcheck, a memory error detector
==9617== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==9617== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==9617== Command: ./main
==9617==
input row, col: 7, 7
Formed Matrix:
1  2  3  4  5  6  7
24  25  26  27  28  29  8
23  40  41  42  43  30  9
22  39  48  49  44  31  10
21  38  47  46  45  32  11
20  37  36  35  34  33  12
19  18  17  16  15  14  13
==9617==
==9617== HEAP SUMMARY:
==9617==     in use at exit: 72,704 bytes in 1 blocks
==9617==   total heap usage: 18 allocs, 17 frees, 73,460 bytes allocated
==9617==
==9617== LEAK SUMMARY:
==9617==    definitely lost: 0 bytes in 0 blocks
==9617==    indirectly lost: 0 bytes in 0 blocks
==9617==      possibly lost: 0 bytes in 0 blocks
==9617==    still reachable: 72,704 bytes in 1 blocks
==9617==         suppressed: 0 bytes in 0 blocks
==9617== Rerun with --leak-check=full to see details of leaked memory
==9617==
==9617== For counts of detected and suppressed errors, rerun with: -v
==9617== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
(base) yuq8@rip-riley 14:25:45 ~/253P/hw_lab/9/2_spiralMatrix
$
```

```
$ make
------------compiling main.cpp to create executable program main----------------
g++  -ggdb   -std=c++11   main.cpp   -o   main
------------Congratulation to you! Successfully compile.
------------Run manually by :
------------./main
(base) yuq8@rip-riley 14:26:38 ~/253P/hw_lab/9/2_spiralMatrix
$ valgrind ./main
==9701== Memcheck, a memory error detector
==9701== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==9701== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==9701== Command: ./main
==9701==
input row, col: 7, 4
Formed Matrix:
1  2  3  4
18  19  20  5
17  28  21  6
16  27  22  7
15  26  23  8
14  25  24  9
13  12  11  10
==9701==
==9701== HEAP SUMMARY:
==9701==     in use at exit: 72,704 bytes in 1 blocks
==9701==   total heap usage: 18 allocs, 17 frees, 73,280 bytes allocated
==9701==
==9701== LEAK SUMMARY:
==9701==    definitely lost: 0 bytes in 0 blocks
==9701==    indirectly lost: 0 bytes in 0 blocks
==9701==      possibly lost: 0 bytes in 0 blocks
==9701==    still reachable: 72,704 bytes in 1 blocks
==9701==         suppressed: 0 bytes in 0 blocks
==9701== Rerun with --leak-check=full to see details of leaked memory
==9701==
==9701== For counts of detected and suppressed errors, rerun with: -v
==9701== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
(base) yuq8@rip-riley 14:26:42 ~/253P/hw_lab/9/2_spiralMatrix
```

```
$ vi main.cpp
(base) yuq8@rip-riley 14:28:03 ~/253P/hw_lab/9/2_spiralMatrix
$ make
------------compiling main.cpp to create executable program main----------------
g++  -ggdb   -std=c++11   main.cpp   -o   main
------------Congratulation to you! Successfully compile.
------------Run manually by :
------------./main
(base) yuq8@rip-riley 14:28:08 ~/253P/hw_lab/9/2_spiralMatrix
$ valgrind ./main
==9949== Memcheck, a memory error detector
==9949== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==9949== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==9949== Command: ./main
==9949==
input row, col: 4, 6
Formed Matrix:
1  2  3  4  5  6
16  17  18  19  20  7
15  24  23  22  21  8
14  13  12  11  10  9
==9949==
==9949== HEAP SUMMARY:
==9949==     in use at exit: 72,704 bytes in 1 blocks
==9949==   total heap usage: 12 allocs, 11 frees, 73,112 bytes allocated
==9949==
==9949== LEAK SUMMARY:
==9949==    definitely lost: 0 bytes in 0 blocks
==9949==    indirectly lost: 0 bytes in 0 blocks
==9949==      possibly lost: 0 bytes in 0 blocks
==9949==    still reachable: 72,704 bytes in 1 blocks
==9949==         suppressed: 0 bytes in 0 blocks
==9949== Rerun with --leak-check=full to see details of leaked memory
==9949==
==9949== For counts of detected and suppressed errors, rerun with: -v
==9949== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
(base) yuq8@rip-riley 14:28:11 ~/253P/hw_lab/9/2_spiralMatrix
$
```

```
$ make
------------compiling main.cpp to create executable program main----------------
g++  -ggdb   -std=c++11   main.cpp   -o   main
------------Congratulation to you! Successfully compile.
------------Run manually by :
------------./main
(base) yuq8@rip-riley 14:29:57 ~/253P/hw_lab/9/2_spiralMatrix
$ valgrind ./main
==10093== Memcheck, a memory error detector
==10093== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==10093== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==10093== Command: ./main
==10093==
input row, col: 6, 4
Formed Matrix:
1  2  3  4
16  17  18  5
15  24  19  6
14  23  20  7
13  22  21  8
12  11  10  9
==10093==
==10093== HEAP SUMMARY:
==10093==     in use at exit: 72,704 bytes in 1 blocks
==10093==   total heap usage: 16 allocs, 15 frees, 73,200 bytes allocated
==10093==
==10093== LEAK SUMMARY:
==10093==    definitely lost: 0 bytes in 0 blocks
==10093==    indirectly lost: 0 bytes in 0 blocks
==10093==      possibly lost: 0 bytes in 0 blocks
==10093==    still reachable: 72,704 bytes in 1 blocks
==10093==         suppressed: 0 bytes in 0 blocks
==10093== Rerun with --leak-check=full to see details of leaked memory
==10093==
==10093== For counts of detected and suppressed errors, rerun with: -v
==10093== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
(base) yuq8@rip-riley 14:30:00 ~/253P/hw_lab/9/2_spiralMatrix
$
```

Conclusion: right result, 0 error in memory leak.

Little Notes:

1) At this lab, when encountered with different length in width and heighth, there are a little bit difference in tackling.

Lab4: sort Color

test cases 1,2, 3:

```
$ cd 4_sortColor/
(base) yuq8@rip-riley 14:30:30 ~/253P/hw_lab/9/4_sortColor
$ make
------------compiling main.cpp to create executable program main----------------
g++  -ggdb   -std=c++11   main.cpp   -o   main
------------Congratulation to you! Successfully compile.
------------Run manually by :
------------./main
(base) yuq8@rip-riley 14:30:36 ~/253P/hw_lab/9/4_sortColor
$ valgrind ./main
==10474== Memcheck, a memory error detector
==10474== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==10474== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==10474== Command: ./main
==10474==
Input Color:
B  R  R  B  R  G  R  G  G
Segregated Color:
R  R  R  R  G  G  G  B  B
==10474==
==10474== HEAP SUMMARY:
==10474==     in use at exit: 72,704 bytes in 1 blocks
==10474==   total heap usage: 1 allocs, 0 frees, 72,704 bytes allocated
==10474==
==10474== LEAK SUMMARY:
==10474==    definitely lost: 0 bytes in 0 blocks
==10474==    indirectly lost: 0 bytes in 0 blocks
==10474==      possibly lost: 0 bytes in 0 blocks
==10474==    still reachable: 72,704 bytes in 1 blocks
==10474==         suppressed: 0 bytes in 0 blocks
==10474== Rerun with --leak-check=full to see details of leaked memory
==10474==
==10474== For counts of detected and suppressed errors, rerun with: -v
==10474== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
(base) yuq8@rip-riley 14:31:12 ~/253P/hw_lab/9/4_sortColor
$
```

```
$ vi main.cpp
(base) yuq8@rip-riley 14:31:56 ~/253P/hw_lab/9/4_sortColor
$ make
------------compiling main.cpp to create executable program main----------------
g++  -ggdb   -std=c++11   main.cpp   -o   main
------------Congratulation to you! Successfully compile.
------------Run manually by :
------------./main
(base) yuq8@rip-riley 14:32:02 ~/253P/hw_lab/9/4_sortColor
$ valgrind ./main
==10537== Memcheck, a memory error detector
==10537== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==10537== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==10537== Command: ./main
==10537==
Input Color:
G  B  R  R  B  R  G
Segregated Color:
R  R  R  G  G  B  B
==10537==
==10537== HEAP SUMMARY:
==10537==     in use at exit: 72,704 bytes in 1 blocks
==10537==   total heap usage: 1 allocs, 0 frees, 72,704 bytes allocated
==10537==
==10537== LEAK SUMMARY:
==10537==    definitely lost: 0 bytes in 0 blocks
==10537==    indirectly lost: 0 bytes in 0 blocks
==10537==      possibly lost: 0 bytes in 0 blocks
==10537==    still reachable: 72,704 bytes in 1 blocks
==10537==         suppressed: 0 bytes in 0 blocks
==10537== Rerun with --leak-check=full to see details of leaked memory
==10537==
==10537== For counts of detected and suppressed errors, rerun with: -v
==10537== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
(base) yuq8@rip-riley 14:32:05 ~/253P/hw_lab/9/4_sortColor
$
```

```
$ vi main.cpp
(base) yuq8@rip-riley 14:33:37 ~/253P/hw_lab/9/4_sortColor
$ make
------------compiling main.cpp to create executable program main---------------
g++  -ggdb   -std=c++11   main.cpp   -o   main
------------Congratulation to you! Successfully compile.
------------Run manually by :
------------./main
(base) yuq8@rip-riley 14:33:40 ~/253P/hw_lab/9/4_sortColor
$ valgrind ./main
==10677== Memcheck, a memory error detector
==10677== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==10677== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==10677== Command: ./main
==10677==
Input Color:
G  B
Segregated Color:
G  B
==10677==
==10677== HEAP SUMMARY:
==10677==     in use at exit: 72,704 bytes in 1 blocks
==10677==   total heap usage: 1 allocs, 0 frees, 72,704 bytes allocated
==10677==
==10677== LEAK SUMMARY:
==10677==    definitely lost: 0 bytes in 0 blocks
==10677==    indirectly lost: 0 bytes in 0 blocks
==10677==      possibly lost: 0 bytes in 0 blocks
==10677==    still reachable: 72,704 bytes in 1 blocks
==10677==         suppressed: 0 bytes in 0 blocks
==10677== Rerun with --leak-check=full to see details of leaked memory
==10677==
==10677== For counts of detected and suppressed errors, rerun with: -v
==10677== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
(base) yuq8@rip-riley 14:33:44 ~/253P/hw_lab/9/4_sortColor
$
```

Lab 6: rotatedSearch

test cases 1,2, 3:

```
$ make
------------compiling main.cpp to create executable program main----------------
g++  -ggdb   -std=c++11   main.cpp   -o   main
------------Congratulation to you! Successfully compile.
------------Run manually by :
------------./main
(base) yuq8@rip-riley 14:34:22 ~/253P/hw_lab/9/6_rotatedSearch
$ valgrind ./main
==10794== Memcheck, a memory error detector
==10794== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==10794== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==10794== Command: ./main
==10794==
Input array:
1
Target: 1
Return the index in the array: 0
==10794==
==10794== HEAP SUMMARY:
==10794==     in use at exit: 72,704 bytes in 1 blocks
==10794==   total heap usage: 1 allocs, 0 frees, 72,704 bytes allocated
==10794==
==10794== LEAK SUMMARY:
==10794==    definitely lost: 0 bytes in 0 blocks
==10794==    indirectly lost: 0 bytes in 0 blocks
==10794==      possibly lost: 0 bytes in 0 blocks
==10794==    still reachable: 72,704 bytes in 1 blocks
==10794==         suppressed: 0 bytes in 0 blocks
==10794== Rerun with --leak-check=full to see details of leaked memory
==10794==
==10794== For counts of detected and suppressed errors, rerun with: -v
==10794== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
(base) yuq8@rip-riley 14:34:35 ~/253P/hw_lab/9/6_rotatedSearch
$
```

```
$ vi main.cpp
(base) yuq8@rip-riley 14:35:26 ~/253P/hw_lab/9/6_rotatedSearch
$ make
------------compiling main.cpp to create executable program main----------------
g++  -ggdb   -std=c++11   main.cpp   -o   main
------------Congratulation to you! Successfully compile.
------------Run manually by :
------------./main
(base) yuq8@rip-riley 14:35:30 ~/253P/hw_lab/9/6_rotatedSearch
$ valgrind ./main
==10945== Memcheck, a memory error detector
==10945== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==10945== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==10945== Command: ./main
==10945==
Input array:
13  18  25  64  1  2  8  10
Target: 8
Return the index in the array: 6
==10945==
==10945== HEAP SUMMARY:
==10945==     in use at exit: 72,704 bytes in 1 blocks
==10945==   total heap usage: 1 allocs, 0 frees, 72,704 bytes allocated
==10945==
==10945== LEAK SUMMARY:
==10945==    definitely lost: 0 bytes in 0 blocks
==10945==    indirectly lost: 0 bytes in 0 blocks
==10945==      possibly lost: 0 bytes in 0 blocks
==10945==    still reachable: 72,704 bytes in 1 blocks
==10945==         suppressed: 0 bytes in 0 blocks
==10945== Rerun with --leak-check=full to see details of leaked memory
==10945==
==10945== For counts of detected and suppressed errors, rerun with: -v
==10945== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
(base) yuq8@rip-riley 14:35:33 ~/253P/hw_lab/9/6_rotatedSearch
$
```

```
$ vi main.cpp
(base) yuq8@rip-riley 14:36:13 ~/253P/hw_lab/9/6_rotatedSearch
$ make
------------compiling main.cpp to create executable program main----------------
g++  -ggdb   -std=c++11   main.cpp   -o   main
------------Congratulation to you! Successfully compile.
------------Run manually by :
------------./main
(base) yuq8@rip-riley 14:36:17 ~/253P/hw_lab/9/6_rotatedSearch
$ valgrind ./main
==11150== Memcheck, a memory error detector
==11150== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==11150== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==11150== Command: ./main
==11150==
Input array:
13  18  25  2  8  10
Target: 8
Return the index in the array: 4
==11150==
==11150== HEAP SUMMARY:
==11150==     in use at exit: 72,704 bytes in 1 blocks
==11150==   total heap usage: 1 allocs, 0 frees, 72,704 bytes allocated
==11150==
==11150== LEAK SUMMARY:
==11150==    definitely lost: 0 bytes in 0 blocks
==11150==    indirectly lost: 0 bytes in 0 blocks
==11150==      possibly lost: 0 bytes in 0 blocks
==11150==    still reachable: 72,704 bytes in 1 blocks
==11150==         suppressed: 0 bytes in 0 blocks
==11150== Rerun with --leak-check=full to see details of leaked memory
==11150==
==11150== For counts of detected and suppressed errors, rerun with: -v
==11150== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
(base) yuq8@rip-riley 14:36:19 ~/253P/hw_lab/9/6_rotatedSearch
$
```