```
// $Id: leaks.cpp,v 1.4 2014-04-10 14:17:42-07 - - $

#include <cstdlib>
#include <iostream>
#include <string>
#include <vector>

using namespace std;

int main (int argc, char** argv) {
   vector<string*> *vecstr = new vector<string*>();
   cout << "vector<string*> = " << vecstr << endl;
   for (int itor = 0; itor < argc; ++itor) {
      string str = argv[itor];
      vecstr->push_back (new string(str));
   }
   for (auto itor = vecstr->cbegin(); itor != vecstr->cend(); ++itor) {
      cout << "string* = " << *itor << " -> " << **itor << endl;
   }
   cerr << "We are done." << endl;
   return EXIT_SUCCESS;
}

/*
//TEST// valgrind --leak-check=full --show-reachable=yes \
//TEST//       --log-file=leaks.out.grind \
//TEST//       leaks foo bar baz qux >leaks.out 2>&1
//TEST// mkpspdf leaks.ps leaks.cpp* leaks.out*
*/

```

```
* @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: starting leaks.cpp
* leaks.cpp: $Id: leaks.cpp,v 1.4 2014-04-10 14:17:42-07 - - $
* g++ -g -O0 -Wall -Wextra -std=gnu++11 leaks.cpp -o leaks -lm
* rm -f leaks.o
* @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: finished leaks.cpp
```

```
vector<string*> = 0x4c2b040
string* = 0x4c2b100 -> leaks
string* = 0x4c2b200 -> foo
string* = 0x4c2b300 -> bar
string* = 0x4c2b410 -> baz
string* = 0x4c2b4c0 -> qux
We are done.
```

```
==25687== Memcheck, a memory error detector
==25687== Copyright (C) 2002-2012, and GNU GPL'd, by Julian Seward et al.
==25687== Using Valgrind-3.8.1 and LibVEX; rerun with -h for copyright info
==25687== Command: leaks foo bar baz qux
==25687== Parent PID: 25686
==25687== 
==25687== 
==25687== HEAP SUMMARY:
==25687==     in use at exit: 270 bytes in 12 blocks
==25687==   total heap usage: 15 allocs, 3 frees, 326 bytes allocated
==25687== 
==25687== 40 bytes in 5 blocks are indirectly lost in loss record 1 of 4
==25687==    at 0x4A075FC: operator new(unsigned long) (in /opt/rh/devtoolset-2/root/usr/lib64/valgrind/vgpreload_memcheck-amd64-linux.so)
==25687==    by 0x400E8D: main (leaks.cpp:15)
==25687== 
==25687== 64 bytes in 1 blocks are indirectly lost in loss record 2 of 4
==25687==    at 0x4A075FC: operator new(unsigned long) (in /opt/rh/devtoolset-2/root/usr/lib64/valgrind/vgpreload_memcheck-amd64-linux.so)
==25687==    by 0x40181D: __gnu_cxx::new_allocator<std::string*>::allocate(unsigned long, void const*) (new_allocator.h:104)
==25687==    by 0x401668: std::_Vector_base<std::string*, std::allocator<std::string*> >::_M_allocate(unsigned long) (in /afs/cats.ucsc.edu/courses/cmps109-wm/Examples/wk02b-classes-exns/leaks)
==25687==    by 0x40134F: void std::vector<std::string*, std::allocator<std::string*> >::_M_emplace_back_aux<std::string*>(std::string*&&) (vector.tcc:404)
==25687==    by 0x40123A: void std::vector<std::string*, std::allocator<std::string*> >::emplace_back<std::string*>(std::string*&&) (vector.tcc:101)
==25687==    by 0x4010C1: std::vector<std::string*, std::allocator<std::string*> >::push_back(std::string*&&) (stl_vector.h:920)
==25687==    by 0x400EB6: main (leaks.cpp:15)
==25687== 
==25687== 142 bytes in 5 blocks are indirectly lost in loss record 3 of 4
==25687==    at 0x4A075FC: operator new(unsigned long) (in /opt/rh/devtoolset-2/root/usr/lib64/valgrind/vgpreload_memcheck-amd64-linux.so)
==25687==    by 0x38C129C3C8: std::string::_Rep::_S_create(unsigned long, unsigned long, std::allocator<char> const&) (in /usr/lib64/libstdc++.so.6.0.13)
==25687==    by 0x38C129CDE4: ??? (in /usr/lib64/libstdc++.so.6.0.13)
==25687==    by 0x38C129CF32: std::basic_string<char, std::char_traits<char>, std::allocator<char> >::basic_string(char const*, std::allocator<char> const&) (in /usr/lib64/libstdc++.so.6.0.13)
==25687==    by 0x400E77: main (leaks.cpp:14)
==25687== 
==25687== 270 (24 direct, 246 indirect) bytes in 1 blocks are definitely lost in loss record 4 of 4
==25687==    at 0x4A075FC: operator new(unsigned long) (in /opt/rh/devtoolset-2/root/usr/lib64/valgrind/vgpreload_memcheck-amd64-linux.so)
==25687==    by 0x400DFB: main (leaks.cpp:11)
==25687== 
==25687== LEAK SUMMARY:
==25687==    definitely lost: 24 bytes in 1 blocks
==25687==    indirectly lost: 246 bytes in 11 blocks
==25687==      possibly lost: 0 bytes in 0 blocks
```

```
==25687==    still reachable: 0 bytes in 0 blocks
==25687==         suppressed: 0 bytes in 0 blocks
==25687==
==25687== For counts of detected and suppressed errors, rerun with: -v
==25687== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 6 from 6)
```