```
// $Id: vecitor.cpp,v 1.15 2015-03-31 18:00:22-07 - - $

//
// NAME
//    vecitor - example of iterator over a vector and an array.
//
// SYNOPSIS
//    vecitor
//
// DESCRIPTION
//    Shows the idea of an iterator.  Iterate over an array and a
//    vector, both using subscripts and using iteration.
//

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

using namespace std;

const int array[] = {3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5};
constexpr size_t asize = sizeof array / sizeof *array;

vector<int> vec (&array[0], &array[asize]);

void print_array_sub () {
   cout << "print_array_sub:";
   for (size_t ai = 0; ai < asize; ++ai) cout << " " << array[ai];
   cout << endl;
}

void print_array_itor () {
   cout << "print_array_itor:";
   for (const int* aip = array; aip != &array[asize]; ++aip) {
      cout << " " << *aip;
   }
   cout << endl;
}

void print_vec_sub () {
   cout << "print_vec_sub:";
   vector<int>::size_type index = 0;
   for (; index < vec.size(); ++index) cout << " " << vec[index];
   cout << endl;
}

void print_vec_itor () {
   cout << "print_vec_itor:";
   auto itor = vec.begin();
   for (; itor != vec.end(); ++itor) cout << " " << *itor;
   cout << endl;
}

```

```

int main () {
   print_array_sub ();
   print_array_itor ();
   print_vec_sub ();
   print_vec_itor ();
   return EXIT_SUCCESS;
}

//TEST// ./vecitor >vecitor.lis 2>&1
//TEST// mkpspdf vecitor.ps vecitor.cpp* vecitor.lis

```

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: starting vecitor.cpp
vecitor.cpp:
     $Id: vecitor.cpp,v 1.15 2015-03-31 18:00:22-07 - - $
g++ -g -O0 -Wall -Wextra -rdynamic -std=gnu++11 vecitor.cpp -o vecitor -lglut -lGLU -lGL -lX11 -lrt -lm
rm -f vecitor.o
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: finished vecitor.cpp
```

```
print_array_sub: 3 1 4 1 5 9 2 6 5 3 5
print_array_itor: 3 1 4 1 5 9 2 6 5 3 5
print_vec_sub: 3 1 4 1 5 9 2 6 5 3 5
print_vec_itor: 3 1 4 1 5 9 2 6 5 3 5
```