**How to find duplicates in a vector ?**

In this article we will discuss how to find duplicate elements in vector and their repetition count.

Suppose we have a vector of strings i.e.

|  |  |
| --- | --- |
| 1  2  3 | // Vector of strings  std::vector<std::string> vecOfStings{ "at" , "hello", "hi", "there", "where", "now", "is", \  "that" , "hi" , "where", "at", "no", "yes", "at"}; |

Let’s find duplicate elements from this list and their duplication count. For example in above vector duplicate strings and their duplication count is as follows,

|  |  |
| --- | --- |
| 1  2  3 | at :: 3  hi :: 2  where :: 2 |

Let’s see how to do that,

**Finding duplicates in a vector**

Steps are :

* Create a map of <string , int> type to store the frequency count of each string in vector.
* Iterate over all the elements in vector try to insert it in map as key with value as 1.

If string already exists in map then increment its value by 1.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | // Create a map to store the frequency of each element in vector  std::map<std::string, int> countMap;    // Iterate over the vector and store the frequency of each element in map  for (auto & elem : vecOfStings)  {  auto result = countMap.insert(std::pair<std::string, int>(elem, 1));  if (result.second == false)  result.first->second++;  } |

Now iterate over the map and print items whose value is greater than 1 i.e.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | // Iterate over the map  for (auto & elem : countMap)  {  // If frequency count is greater than 1 then its a duplicate element  if (elem.second > 1)  {  std::cout << elem.first << " :: " << elem.second << std::endl;  }  } |

It will print duplicate elements in vector and their duplication count i.e.

|  |  |
| --- | --- |
| 1  2  3 | at :: 3  hi :: 2  where :: 2 |

**Generic function to find duplicates in vector :**

Create a Generic function to get the duplicate elements and their duplication count i.e.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | /\*  \* Generic function to find duplicates elements in vector.  \* It adds the duplicate elements and their duplication count in given map countMap  \*/  template <typename T>  void findDuplicates(std::vector<T> & vecOfElements, std::map<T, int> & countMap)  {  // Iterate over the vector and store the frequency of each element in map  for (auto & elem : vecOfElements)  {  auto result = countMap.insert(std::pair<std::string, int>(elem, 1));  if (result.second == false)  result.first->second++;  }    // Remove the elements from Map which has 1 frequency count  for (auto it = countMap.begin() ; it != countMap.end() ;)  {  if (it->second == 1)  it = countMap.erase(it);  else  it++;    }  } |

Let’s use this generic function to find duplicate elements in vector i.e.

|  |  |
| --- | --- |
| 1  2  3  4 | std::map<std::string, int> duplicateElements;    // Get the duplicate elements in vector  findDuplicates(vecOfStings, duplicateElements); |

Now Iterate over this map to print the duplicate elements with count i.e.

|  |  |
| --- | --- |
| 1  2 | for (auto & elem : duplicateElements)  std::cout << elem.first << " :: " << elem.second << std::endl; |

|  |  |
| --- | --- |
| 1  2  3 | at :: 3  hi :: 2  where :: 2 |

**Complete example is as follows,**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89 | #include <iostream>  #include <vector>  #include <map>  #include <algorithm>  #include <functional>  #include <string>    // Print the contents of vector  template <typename T>  void print(T & vecOfElements, std::string delimeter = " , ")  {  for(auto elem : vecOfElements)  std::cout<<elem<<delimeter;  std::cout << std::endl;  }    /\*  \* Generic function to find duplicates elements in vector.  \* It adds the duplicate elements and their duplication count in given map countMap  \*/  template <typename T>  void findDuplicates(std::vector<T> & vecOfElements, std::map<T, int> & countMap)  {  // Iterate over the vector and store the frequency of each element in map  for (auto & elem : vecOfElements)  {  auto result = countMap.insert(std::pair<std::string, int>(elem, 1));  if (result.second == false)  result.first->second++;  }    // Remove the elements from Map which has 1 frequency count  for (auto it = countMap.begin() ; it != countMap.end() ;)  {  if (it->second == 1)  it = countMap.erase(it);  else  it++;    }  }      int main()  {  // Vector of strings  std::vector<std::string> vecOfStings{ "at" , "hello", "hi", "there", "where", "now", "is", \  "that" , "hi" , "where", "at", "no", "yes", "at"};    print(vecOfStings);    // Create a map to store the frequency of each element in vector  std::map<std::string, int> countMap;    // Iterate over the vector and store the frequency of each element in map  for (auto & elem : vecOfStings)  {  auto result = countMap.insert(std::pair<std::string, int>(elem, 1));  if (result.second == false)  result.first->second++;  }    std::cout << "Duplicate elements and their duplication count " << std::endl;    // Iterate over the map  for (auto & elem : countMap)  {  // If frequency count is greater than 1 then its a duplicate element  if (elem.second > 1)  {  std::cout << elem.first << " :: " << elem.second << std::endl;  }  }    /\*  \* Finding duplicates in vector using generic function  \*/    std::map<std::string, int> duplicateElements;    // Get the duplicate elements in vector  findDuplicates(vecOfStings, duplicateElements);    std::cout << "Duplicate elements and their duplication count " << std::endl;  for (auto & elem : duplicateElements)  std::cout << elem.first << " :: " << elem.second << std::endl;    return 0;  } |

**Output:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | at , hello , hi , there , where , now , is , that , hi , where , at , no , yes , at ,  Duplicate elements and their duplication count  at :: 3  hi :: 2  where :: 2  Duplicate elements and their duplication count  at :: 3  hi :: 2  where :: 2 |

To compile the example use following command,

***g++ –std=c++11 example.cpp***