

[Sign up](#)[Code](#)[Issues](#) 0[Pull requests](#) 0[Projects](#) 0[Actions](#)[Wiki](#)[Security](#)[Pulse](#)

Branch: master

[Find file](#)[Copy path](#)[cpp_kursu_odevleri](#) / [containers](#) / [cnt_03.md](#)

Fetching contributors...



67 lines (52 sloc) | 1.63 KB

[Raw](#)[Blame](#)[History](#)

Elimizde küçükten büyüğe sıralı bir tamsayı *vector*'ü var. Bu *vector* içinde *[0 – 99]* aralığında bulunan *n* tane tamsayı tutuluyor. *n* değeri *0* dahil *100* dahil herhangi bir tamsayı olabilir. Her tamsayıdan en fazla bir tane bulunacağı garanti altında. Örnek:

```
vector<int> ivec{ 5, 23, 25, 30, 45, 47, 70, 98 };
```

Yapmanız gereken bu *vector*'den hareketle bir *string* oluşturmak. Oluşturduğunuz *string*'de bu *vector*'de bulunmayan tamsayılar aşağıdaki notasyon ile gösterilecek:

```
0-4, 6-22, 24, 26-29, 31-44, 46, 48-69, 71-97, 99
```

Birkaç örnek daha

```
{ } --> 0-99  
{ 0 } --> 1-99  
{ 3, 5 } --> 0-2, 4, 6-99
```

Bu *string*i oluşturacak bir işlev yazmalısınız:

```
#include <vector>  
#include <string>  
  
std::string get_missing_integers(const std::vector<int> &ivec);
```

- İşlevin parametresi tamsayıları tutan *vector*.
- İşlevin geri dönüş değeri istenen *string*.

Tanımladığınız işlevi aşağıdaki program ile sınavabilirsiniz:

```
#include <set>
#include <vector>
#include <ctime>
#include <algorithm>
#include <iterator>
#include <iostream>
#include <string>

using namespace std;

string get_missing_integers(const vector<int> &ivec);

vector<int> getRandomTestVector()
{
    set<int> iset;
    size_t n = rand() % 100;
    while (iset.size() < n) {
        iset.insert(rand() % 100);
    }
    return vector<int>(iset.begin(), iset.end());
}

int main()
{
    srand(static_cast<unsigned>(time(nullptr)));
    auto vec = getRandomTestVector();
    copy(vec.begin(), vec.end(), ostream_iterator<int>(cout, " "));
    cout << endl;
    cout << get_missing_integers(vec) << endl;

    return 0;
}
```