

[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_08.md**

Fetching contributors...



42 lines (35 sloc) | 1.48 KB

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

Aşağıdaki test kodunda belirtilen yere (ayrı ayrı) şu işleri gerçekleştirecek kodları yazınız. Gereken yerlerde lamba ifadelerini kullanınız:

1. vector'de tutulan ilk öğeyi silin.
2. vector'de tutulan son öğeyi silin.
3. vector'de tutulan 15. öğeyi silin.
4. ilk ve son öğe haricindeki tüm öğeleri (varsa) silin.
5. değeri 5 olan ilk öğeyi (varsa) silin.
6. değeri 5 olan son öğeyi (varsa) silin.
7. değeri 5 olan tüm öğeleri (varsa) silin.
8. standart giriş akımından alınacak değerdeki tüm öğeleri (varsa) silin
9. standart giriş akımından alınacak indeksteki öğeyi silin
10. standart giriş akımından alınacak aralıktaki tüm öğeleri (varsa) silin. Örnek [3 - 7] aralığı.
11. tüm çift sayıları (varsa) ilin.
12. indeksi 3'e tam bölünen öğeleri silin.
13. ardışık aynı değerde olan öğelerin sayısını teke indirin.
14. vector'de aynı değerden birden fazla olmayacak şekilde silme işlemi yapın. Silme işleminden sonra her değer "unique" olmalı.

```
#include <vector>
#include <cstdlib>
#include <ctime>
#include <algorithm>
#include <iostream>
#include <iterator>

int main()
{
    using namespace std;
    vector<int> ivec(100);
    srand(static_cast<unsigned>(time(nullptr)));
    generate(ivec.begin(), ivec.end(), [] {return rand() % 20; });
    copy(ivec.begin(), ivec.end(), ostream_iterator<int>(cout, " "));
    //.....
    // kodunuz
    std::cout << "\n\n\n";
    copy(ivec.begin(), ivec.end(), ostream_iterator<int>(cout, " "));
}
```

© 2020 GitHub, Inc.

[Terms](#)

[Privacy](#)

[Security](#)

[Status](#)

[Help](#)

[Contact GitHub](#)

[Pricing](#)

[API](#)

[Training](#)

[Blog](#)

[About](#)