

* istream iterator;

```
int main()
{
    using namespace std;

    istringstream iss{ "23 12 30 57 41 20 51 42 16 25 44" };

    auto iter = istream_iterator<int>(iss);
    auto val = *iter;

    std::cout << "val = " << val << "\n";
    ++iter;
    val = *iter;
    std::cout << "val = " << val << "\n";
```

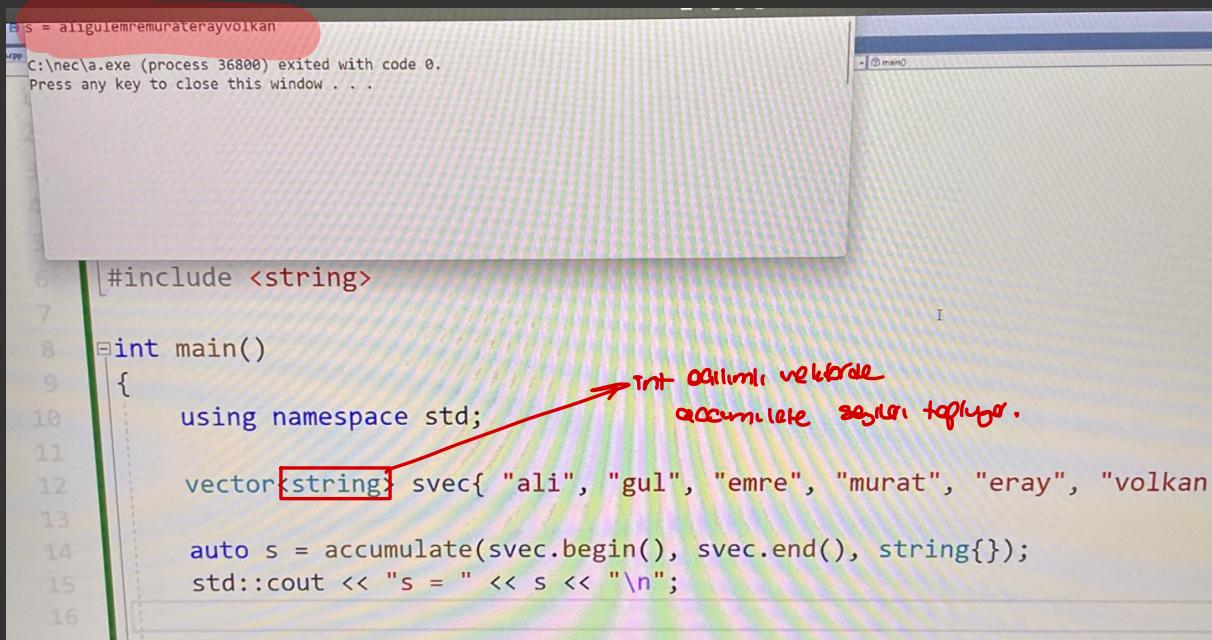
→ Bu iteratör kullanırsak, forlu algoritmalar kullanılır.

```
#include <iostream>
#include <string>
#include <vector>

int main()
{
    using namespace std;

    vector<string> svec{ "ali", "gul", "emre", "murat", "eray", "volkan" };

    auto s = accumulate(svec.begin(), svec.end(), string{});
    std::cout << "s = " << s << "\n";
```



C:\> S = ali gul emre murat eray volkan
C:\nec\al.exe (process 36800) exited with code 0.
Press any key to close this window . . .

```
int main()
{
    using namespace std;

    istringstream iss{ "23 12 30 57 41 20 51 42 16 25 44" };

    cout << accumulate(istream_iterator<int>(iss), istream_iterator<int>{}, 0) << '\n'
```

sentinel olarak kullanmak {3}

tempate

bu arguman yerine {} yazarak
değiştir int etsen, yine aynı sonucu elde edebilir

```

8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
}

```

int main()
{
 using namespace std;

 istringstream iss{ "23 12 30 57 41 20 51 42 16 25 44" };

 cout << accumulate(istream_iterator<int>{iss}, {}, 0) << '\n';
 iss.clear();]→ her istemden sonra, clear! Content iss'in sondayız, eof state
 iss.seekg(0);
 iss >> hex;]→ hexten degerler.
 cout << accumulate(istream_iterator<int>{iss}, {}, 0) << '\n';
 iss.clear();
 iss.seekg(0);
 iss >> oct;
 cout << accumulate(istream_iterator<int>{iss}, {}, 0) << '\n';
}



Aynı seconde string de tutulurdu!

```

int main()

using namespace std;

istringstream iss{ "23 12 30 57 41 20 51 42 16 25 44" };

auto str = accumulate(istream_iterator<string>{iss}, {}, ""s"); user defined literal  
bu default initialized  
edilmiş string {} ile  
aynı!

reverse(str.begin(), str.end());

cout << "[" << str << "]\\n"; ]→ Once initialized, sonraki değerlerde, [uu s2b1... ]
}

```

* Bir Dosyadan Vektöre Kopyalama Örneği:

```

int main()
{
    using namespace std;

    auto ifs = open_text_file("primes10000.txt");
    vector<int> ivec;

    ivec.reserve(10'000); ]→ size, bitmek her vektor için  
reserve kullan! ]→ Mükemmel  
sakur.
    int ival;

    while (ifs >> ival)
        ivec.push_back(ival);

    std::cout << "ivec.size() = " << ivec.size() << "\\n";
}

```

→ Fırat bu işlemi while döngüsü aracılıkla
yapabiliyor.

→ Bu istringstream_iterator<int> kullanımlı ile.
yaptı:



```

using namespace std;

auto ifs = open_text_file("primes10000.txt");

vector<int> ivec{ istream_iterator<int>{ifs}, {} }};

std::cout << "ivec.size() = " << ivec.size() << "\\n";

```

→ Bu yöntemle, bir dosya içeriğini
string olmak üzere okunacak form tükenebilir.
Content string de bir container.

```

std::string file_to_str(const std::string& filename)
{
    auto ifs = open_text_file(filename);
    ifs >> std::noskipws; ]→ no skip whitespace / Bu olmadan basitlikler ..... cokte yorgar.
    return std::string{ std::istream_iterator<char>{ifs}, {} };
}

int main()
{
    using namespace std;

    auto str = file_to_str("nutility.h");

    //std::cout << "str.size() = " << str.size() << "\n";
    cout << str << "\n";
}

```

I
 Gorler
 string centeren
 Coker teter.

* Istream Iteratör ile standart inputları okuyarak devreye alma:

12 34 56 23 41 5 9 22 0 -5 6 19^Z
 C:\nec\c.exe (process 16184) exited with code 0.
 Press any key to close this window . . .

→ hemen gidiyor kod

```

int main()
{
    using namespace std;

    cout << *max_element(istream_iterator<int>{cin}, {});
}

min_element
iseki en lakin dandorardo.

```

minmax_element "par"
 return eder!

*Binary Formatter Dosya İşlemi:

```
using namespace std;

int n;
cout << "ilk kac asal sayi: ";
cin >> n;

string filename{ "asal"s + to_string(n) + ".dat" };
auto ofs = create_binary_file(filename);

int prime_count{};
int x = 2;
while (prime_count < n) {
    if (isprime(x)) {
        ofs.write(reinterpret_cast<const char*>(&x), sizeof(x));
        ++prime_count;
    }
    ++x;
}
}
```

Formatterıza yarın, Tileni

Ayrıca: eğer const tileni kullanırsak data okunur, bu nedenle tespit etmek data kolay.

→ Dosyalar okuma:

```
#include <iostream>
#include <iostream>
#include <string>
#include "nutility.h"
#include "date.h"
#include <conio.h>

int main()
{
    using namespace std;

    auto ifs{ open_binary_file("asal345.dat") };

    int x;

    while (ifs.read(reinterpret_cast<char*>(&x), sizeof(x))) {
        cout << x << " ";
        _getch();
    }
}
```

Klavyeden input alırsınız.

```
int main()
{
    using namespace std;
    auto ifs{ open_binary_file("date333.dat") };

    Date dt;

    while (ifs.read(reinterpret_cast<char*>(&dt), sizeof(dt))) {
        cout << dt;
        getchar();
    }
}
```

* Neden Standard Output Yerine Error Output: cout <> yerine cerr <>

- Okunabilirlik Orta

- Standard Output Yenidenisletilir: Uneren standard output, bu dosya
yerine else, hata mesaji da dosyaya yezildi. Fakat hata mesajini bilmeyen
yenilebilir. Hata mesajları, hata okumada kullanlı veriye karsola basmali

* ifstream::in.get Fonksiyonu: → while loop kullanır.

```
auto ifs = open_binary_file(argv[1]);
char c;
int file_count{};
const int chunk{ std::atoi(argv[2]) };
int byte_count{};

ofstream ofs;
while (ifs.get(c)) {
    if (!ofs.is_open()) {
        ostringstream oss;
        oss.fill('0');
        oss << "part" << setw(3) << file_count + 1 << ".par";
        ofs.open(oss.str(), ios::binary);
        ++file_count;
    }
    // get, redireks sementisti ne dosyadaki 1 karakteri,
    // bu char c'ye oter.
    // while döngüsü dosya bitine son bulur!
    // (eof)
}
```

* Verimli Sırasız Dosya Adı ve Size Düşük Hafıza, Dosya, O kadar Parçaya Bolme:

```
int main(int argc, char **argv)
{
    using namespace std;

    if (argc != 3) {
        cerr << "usage : <fdiv> <filename> <chunk>\n";
        return 1;
    }

    auto ifs = open_binary_file(argv[1]);
    char c;
    int file_count{};
    const int chunk{ std::atoi(argv[2]) };
    int byte_count{};

    ofstream ofs;

    while (ifs.get(c)) {
        if (!ofs.is_open()) {
            ostringstream oss;
            oss.fill('0');
            oss << "part" << setw(3) << file_count + 1 << ".par";
            ofs.open(oss.str(), ios::binary);
            ++file_count;
        }
        ofs.put(c);
        ++byte_count;
        if (byte_count % chunk == 0) {
            ofs.close();
        }
    }

    cout << argv[1] << "dosyasi " << chunk << " byte'lik " << file_count << " parçaya bolundu\n";
}
```

: \nec>fdiv wday.exe 500
lay.exedesyasi 500 byte'lik 21 parcaya bolundu

: \nec>dir *.par
Volume in drive C is Windows-SSD
Volume Serial Number is AC71-1599

Directory of C:\nec

26.10.2022 21:12	500 part001.par
26.10.2022 21:12	500 part002.par
26.10.2022 21:12	500 part003.par
26.10.2022 21:12	500 part004.par
26.10.2022 21:12	500 part005.par
26.10.2022 21:12	500 part006.par
26.10.2022 21:12	500 part007.par
26.10.2022 21:12	500 part008.par
26.10.2022 21:12	500 part009.par
26.10.2022 21:12	500 part010.par
26.10.2022 21:12	500 part011.par
26.10.2022 21:12	500 part012.par
26.10.2022 21:12	500 part013.par
26.10.2022 21:12	500 part014.par
26.10.2022 21:12	500 part015.par
26.10.2022 21:12	500 part016.par
26.10.2022 21:12	500 part017.par
26.10.2022 21:12	500 part018.par
26.10.2022 21:12	500 part019.par
26.10.2022 21:12	500 part020.par
26.10.2022 21:12	240 part021.par
21 File(s) 10.240 bytes	
0 Dir(s) 8.268.685.312 bytes free	

* 1. Örnek Orijinal Çıktıları Snip, Onları bulusturun. Kod :

```
auto ofs = create_binary_file(argv[1]);
int file_count{};
int byte_count{};
char c;

while (true) {
    ostringstream oss;
    oss << setfill('0') << "part" << setw(3) << file_count + 1 << ".par";
    ifstream ifs{ oss.str(), ios::binary };
    if (!ifs)
        break;
    while (ifs.get(c)) {
        ofs.put(c);
        ++byte_count;
    }
    ifs.close();
    if (std::remove(oss.str().c_str())) {
        cerr << oss.str() << " dosyası silinemedi\n";
        return 1;
    }
    ++file_count;
}
```

→ *remove dosyanın oluşturucustringstream'a 0'dan önceki karakterlerini siler.*

* Dosya Venm Problemleri / File Pointer:

fseek
rewind
ftell
fsetpos
fgetpos

→ C deki Filepointer set/get eden değişkenler bonus.

→ C++ de durum degrade cari istream ve ofstream tarih file pointerlar farklı.

// seekg	→ ifstream	// tellg	→ ifstream
// seekp	→ ofstream	// tellp	→ ofstream

```

int fseek(FILE *f, long n, int origin)

SEEK_SET
SEEK_CUR
SEEK_END

fseek(f, 20L, SEEK_CUR);
fseek(f, -20, SEEK_END);

```

```

int main()
{
    using namespace std;

    istringstream iss{ "emirhanulker" };
    iss.seekg(7, ios::beg); → bosun basla, 7. elemara boy
    string str;           → enum

    iss >> str;
    cout << "[" << str << "]\n"; → [ulker]

    iss.seekg(-8, ios::end); → sondan basla, 8 elem
    iss >> str;
    cout << "[" << str << "]\n"; → [hanulker]
}

```

* Stringstream ile hem seekg hem seekp kullanabilir:

```

int main()
{
    using namespace std;

    stringstream ss;

    ss << "125alican982";
    int ival;

    ss >> ival;
    std::cout << "ival = " << ival << "\n";
    ss.seekg(9);
    ss >> ival;
    std::cout << "ival = " << ival << "\n";
    ss.seekp(3);
    ss.put('4');
    ss.seekg(0);

    ss >> ival;
    std::cout << "ival = " << ival << "\n";
}

```

```

int main()
{
    stringstream ss;

    ss << "125alican982";
    int ival;

    ss.seekp(3);
    ss.put('4');
    cout << ss.str() << "\n";
    ss >> ival;

    std::cout << "ival = " << ival << "\n";

    cout << "ss.tellg() = " << ss.tellg() << "\n";
    cout << "ss.tellp() = " << ss.tellp() << "\n"; ] → ilerinin de
                                                değer 4

```

* 6 Çarçılı Fonksiyonu: → Formatta okuma yapmadan.. kaa böyle yaradıını söyle

```

int main()
{
    using namespace std;

    auto ifs = open_binary_file("asal345.dat");

    constexpr int size = 100;

    int a[size];
    ifs.read((char*)a, 100 * 4); = 400 → Dosyayı array'e okudu
    std::cout << "ifs.gcount() = " << ifs.gcount() << "\n";
}

```

* Hatalı Kod Örneği:

```

int main()
{
    using namespace std;

    auto ifs = open_binary_file("asal345.dat");

    constexpr int size = 100;

    vector<int> ivec(size);

    while (ifs.read((char*)ivec.data(), size * sizeof(int))) {
        print(ivec);
    }
}

```

→ Hatanın nedeni:

100'ler 100'ler okunup yollar. Ancak 4. adında, 100 değil, 15 okur.. Alınır fail durumu gelir.