**2023 12 13**

**std::istream\_iterator**

**int** main**()**

**{**

**using** **namespace** std**;**

istream\_iterator**<int>** iter**{**cin**};** ***// streamdeki öğeleri alıyoruz***

**int** val**{};**

val **=** **\***iter**;**

cout **<<** "val = " **<<** val **<<** "\n"**;**

**++**iter**;**

val **=** **\***iter**;**

**++**iter**;**

cout **<<** "val = " **<<** val "\n"**;**

**}**

**int** main**()**

**{**

**using** **namespace** std**;**

istringstream iss**{**"burak kose damla kubat furkan mert melike kaptan"**};**

***// begin end***

copy**(**istream\_iterator**<**string**>{**iss**},** istream\_iterator**<**string**>{},**

ostream\_iterator**<**string**>{**cout**,** "\n"**});**

istringstream iss1**{**"98 123 43 523 123 54 64 12 56 64 23"**};**

**auto** sum **=** accumulate**(**istream\_iterator**<int>{**iss1**},** **{},** 0**);**

**auto** max **=** **\***max\_element**(**istream\_iterator**<int>{**iss1**},** **{},** 0**);**

**}**

**int** main**()**

**{**

**using** **namespace** std**;**

ostream os\_hex**{** cout**.**rdbuf**()** **};**

ostream os\_oct**{** cout**.**rdbuf**()** **};**

os\_hex **<<** hex **<<** uppercase **<<** showbase**;**

os\_oct **<<** oct **<<** showbase**;**

Irand myrand**{**34523**,** 123412**};**

**for** **(int** **=** 0**;** i **<** 10**;** **++**i**)**

**{**

**auto** val **=** myrand**();**

cout **<<** val **<<** "\n"**;** ***// decimal***

os\_hex **<<** val **<<** "\n"**;** ***// hex***

os\_oct **<<** val **<<** "\n"**;** ***// octal***

**}**

**}**

**Dosya İşlemleri**

***/\****

***std::ofstream yazma işlemleri için***

***std::ifstream okuma işlemleri için***

***std::fstream yazma ve okuma işlemleri için***

***#include <fstream>***

***\*/***

***/\****

***ifstream ctor ve open fonksiyonu parametre olarak:***

***dosyanın ismi***

***açış modulus***

***ios::in okuma***

***ios::out yazma***

***ios::app append / sona yazma***

***ios::trunc truncate***

***ios::ate at end***

***ios::binary***

***\*/***

#include <fstream>

**int** main**()**

**{**

**using** **namespace** std**;**

ofstream ofs**{**"emre.txt"**,** ios**::**trunc **|** ios**::**out**};** ***// default ofstream***

ifstream ifs**{**"emre.txt"**,** ios**::**in**};**

**}**

**int** main**()**

**{**

**using** **namespace** std**;**

ofstream ofs**{**"kutay.txt"**};**

**if** **(!**ofs**)** ***// !ofs.good() ya da !ofs.fail()***

**{**

cerr **<<** "cannot open file\n"**;**

**return** 1**;**

**}**

std**::**cout **<<** "file opened succesfully"**;**

**}**

**int** main**()**

**{**

**using** **namespace** std**;**

ifstream ifs**;**

boolalpha**(**cout**);**

***// is\_open streamin state kontrol etmez stream nesnesiyle ilişkilenmiş açık bir dosya var mı kontrol eder***

cout **<<** "ifs.is\_open() = " **<<** ifs**.**is\_open**()** **<<** "\n"**;** ***// false***

ifs**.**open**(**"main.cpp"**);**

cout **<<** "ifs.is\_open() = " **<<** ifs**.**is\_open**()** **<<** "\n"**;** ***// true***

ifs**.**close**();**

cout **<<** "ifs.is\_open() = " **<<** ifs**.**is\_open**()** **<<** "\n"**;** ***// false***

**}**

**int** main**()**

**{**

**using** **namespace** std**;**

ofstream ofs**{** "kutay.txt" **};**

**if** **(!**ofs**)**

**{**

cerr **<<** "dosya olusturulamadi\n"**;**

**return** 1**;**

**}**

**for** **(int** i **=** 0**;** i **<** 100**;** **++**i**)**

**{**

ofs **<<** i **<<** " " **;**

**}**

***// ofstream dtor dosyayı kapatır.***

***// eğer stream good state ise dosyayı kapatıp başka bir dosyada işlem yapabiliriz***

**}**

***// dosyaya yazma işlemleri***

#include <format> ***// cpp 20***

**int** main**()**

**{**

**using** **namespace** std**;**

ofstream ofs **{** "emre.txt" **};**

**if** **(**ofs**.**fail**())**

**{**

cerr **<<** "dosya olusturulamadi\n"**;**

**}**

Irand myrand **{**0**,** 300'000**};**

ofs **<<** left**;**

**for** **(int** i **=** 0**;** i **<** 10'000**;** **++**i**)**

**{**

ofs **<<** format**(**"{:<12} {:<16} {:<20} {}\n"**,** myrand**(),** rname**(),** rfname**(),** rtown**());**

**}**

**}**

***// bir vektörü dosyaya yazma yöntemleri:***

**int** main**()**

**{**

**using** **namespace** std**;**

vector**<**string**>** svec**;**

rfill**(**svec**,** 10'000**,** rname**);**

ofstream ofs **{** "emre.txt" **};**

**if** **(!**ofs**)**

**{**

cerr **<<** "dosya olusturulamadi\n"**;**

**}**

***// 1. yöntem***

**for** **(const** **auto&** name **:** svec**)**

ofs **<<** name **<<** "\n"**;**

***// 2. yöntem***

copy**(**svec**.**begin**(),** svec**.**end**(),** ostream\_iterator**<**string**>(**ofs**,** "\n"**));**

copy\_if**(**svec**.**begin**(),** svec**.**end**(),** ostream\_iterator**<**string**>(**ofs**,** "\n"**),[](**cont string**&** s**)**

**{**

**return** s**.**length**()** **==** 5 **&&** s**.**front**()** **=** 'a'**;**

**});**

generate\_n**(**ostream\_iterator**<**string**>{**ofs **,** "\n"**},** 1000**,** **[]**

**{**

**return** rname**()** **+** ' '**+** rfname**();**

**});**

**}**

***// okuma işlemleri***

**int** main**()**

**{**

**using** **namespace** std**;**

ifstream ifs**{**"primes.txt"**};**

**if** **(!**ifs**)**

cerr **<<** "dosya acilamadi\n"**;**

**int** v**;**

***// (ifs >> word).operator bool***

**while(**ifs **>>** v**)** ***// akım fail olursa false döner ve while sonlanır***

**{**

cout **<<** v **<<** " "**;**

**}**

**}**

**void** func**(**std**::**ifstream**);**

**int** main**()**

**{**

ifstream ifs **{**"emre.txt"**};**

func**(**ifs**);** ***// error copy ctor yok***

func**(**std**::**move**(**ifs**));** ***// hata yok***

**}**

***/\****

***ostream'deki sınıfları incompling type olarak kullancaksak***

***#include<iosfwd> dahil etmek yeterli header'a iosfwd daha light bir include***

***\*/***

std**::**ifstream open\_text\_file**(const** std**::**string**&** filename**)**

**{**

std**::**ifstream ifs **{**filename**};**

**if** **(!**ifs**)**

**{**

**throw** std**::**runtime\_error**{** filename **+** " dosyayisi acilamiyor\n"**};**

**}**

**return** ifs**;**

**}**

std**::**ofstream create\_text\_file**(const** std**::**string**&** filename**)**

**{**

std**::**ofstream ofs **{**filename**};**

**if** **(!**ofs**)**

**throw** std**::**runtime\_error**{**filename **+** "dosyayisi olusturulamadi\n"**};**

**return** ofs**;**

**}**

**int** main**()**

**{**

**try**

**{**

**auto** ofs **=** create\_text\_file**(**"emre.txt"**);**

**}**

**catch(const** std**::**exception **&**ex**)** **{**

std**::**cout **<<** "exception caught: " **<<** ex**.**what**()** **<<** "\n"**;**

**}**

**try**

**{**

**auto** ifs **=** open\_text\_file**(**"emre.txt"**);**

**}**

**catch(const** std**::**exception **&**ex**)** **{**

std**::**cout **<<** "exception caught: " **<<** ex**.**what**()** **<<** "\n"**;**

**}**

**}**

**int** main**()**

**{**

**using** **namespace** std**;**

**int** ival **=** 2345**;**

string name**{**"emrebahtiyar"**};**

**auto** filename **=** **(**ostringstream**{}** **<<** name **<<** "\_" **<<** ival **<<** ".txt"**).**str**();**

ofstream ofs**{**"out.txt"**}** **<<** "bugun hava cok soguk"**;**

**}**

***// byte byte dosya okuma***

**int** main**()**

**{**

ifstream ifs **{**"main.cpp"**};**

**if** **(!**ifs**)**

cerr **<<** "dosya acilamadi\n"**;**

**int** c**;**

**while** **((**c **=** ifs**.**get**())** **!=** EOF**)**

**{**

cout **<<** **static\_cast<char>(**c**);** ***// cout.put((char)c);***

**}**

**}**

***// dosyayı bir defa da okuma***

**int** main**()**

**{**

**auto** ifs **=** open\_text\_file**(**"emre.txt"**);**

cout **<<** ifs**.**rdbuf**();**

**}**

***// getline***

**int** main**()**

**{**

**auto** ifs **=** open\_text\_file**(**"emre.txt"**);**

string sline**;**

getline**(**ifs **,** sline**);** ***// ifs döndürür***

cout **<<** "[" **<<** sline **<<** "]"**;**

**while** **(**getline**(**ifs**,** sline**))** ***// getline (ifs, sline, "\n");***

**{**

cout **<<** sline**;**

**}**

**}**

***// dosyada veri okuyup, vector'e atıp, sıralama***

**int** main**()**

**{**

vector**<**string**>** linevec**;**

linevec**.**reserve**(**10'000**);**

**auto** ifs **=** open\_text\_file**(**"kutay.txt"**);**

string sline**;**

**while** **(**getline**(**ifs**,** sline**))**

**{**

linevec**.**push\_back**(**move**(**sline**));**

**}**

sort**(**linevec**.**begin**(),** linevec**.**end**());**

copy**(**linevec**.**begin**(),** linevec**.**end**(),** ostream\_iterator**<**string**>{**cout**,** "\n"**});**

**}**

**int** main**()**

**{**

**auto** ifs **=** open\_text\_file**(**"kutay.txt"**);**

***// 1***

vector**<int>** ivec**({**istream\_iterator**<int>{**ifs**},** **{}};**

***// 2***

vector**<int>** ivec1**;**

ivec**.**reserve**(**100'000**);**

ivec**.**assign**(**istream\_iterator**<int>{**ifs**},** **{});**

**}**

***// formatsız dosyaya yazma***

**int** main**()**

**{**

**using** **namespace** std**;**

ofstream ofs**{**"primes.dat"**,** ios**::**binary**};**

**int** prime\_count **=**0 **;**

**int** x **=** 2**;**

**while** **(**prime\_count **<** 1'000'000**)**

**{**

**if** **(**isprime**(**x**))**

**{**

ofs**.**write**(reinterpret\_cast<char\*>(&**x**),** **sizeof(int)));**

**++**prime\_count**;**

**}**

**++**x**;**

**}**

**}**