**202211341 컴퓨터공학과 이윤희**

Question1

// Fraction.h

#ifndef FRACTION\_H

#define FRACTION\_H

class Fraction

{

private:

int numer;

int denom;

public:

Fraction(int num, int den);

Fraction();

Fraction(const Fraction& fract);

~Fraction();

int getNumer() const;

int getDenom() const;

void print();

void setNumer(int num);

void setDenom(int den);

private:

void normalize();

int gcd(int n, int m);

};

#endif

// Fraction.cpp

#include "Fraction.h"

#include <iostream>

using namespace std;

Fraction::Fraction(int num, int den) :numer(num), denom(den) {

}

Fraction::Fraction() :numer(0), denom(1) {

}

Fraction::Fraction(const Fraction& fract)

:numer(fract.numer), denom(fract.denom) {

}

Fraction::~Fraction() {

}

int Fraction::getNumer() const {

return numer;

}

int Fraction::getDenom() const {

return denom;

}

void Fraction::print() {

normalize();

cout << numer << "/" << denom << endl;

}

void Fraction::setNumer(int num) {

numer = num;

}

void Fraction::setDenom(int den) {

denom = den;

}

void Fraction::normalize() {

if (denom < 0) {

numer \*= -1;

denom \*= -1;

}

int g = gcd(numer, denom);

numer /= g;

denom /= g;

}

int Fraction::gcd(int n, int m) {

int g, i;

if (n > m)

i = n;

else

i = m;

for (; i > 0; i--) {

if ((n % i == 0) && (m % i == 0)) {

g = i;

break;

}

}

return g;

}

// Main.cpp

#include "Fraction.h"

#include <iostream>

using namespace std;

int main(void) {

Fraction fract1;

Fraction fract2(14, 21);

Fraction fract3(11, -8);

Fraction fract4(fract3);

cout << "Printing four fractions after constructed : " << endl;

cout << "fract1 : ";

fract1.print();

cout << "fract2 : ";

fract2.print();

cout << "fract3 : ";

fract3.print();

cout << "fract4 : ";

fract4.print();

cout << "Changing the first two fractions and printing them : " << endl;

fract1.setNumer(4);

cout << "fract1 : ";

fract1.print();

fract2.setDenom(-5);

cout << "fract2 : ";

fract2.print();

cout << "Testing the changes in two fractions : " << endl;

cout << "Numerator of fract1 : " << fract1.getNumer() << endl;

cout << "Denomirator of fract2: " << fract2.getDenom() << endl;

return 0;

}

텍스트이(가) 표시된 사진

자동 생성된 설명

Question2

// Time.h

#ifndef TIME\_H

#define TIME\_H

class Time

{

private:

int hours;

int minutes;

int seconds;

public:

Time(int hours, int minutes, int seconds);

Time();

~Time();

void print() const;

void tick();

private:

void normalize();

};

#endif

// Time.cpp

#include "Time.h"

#include <iostream>

using namespace std;

Time::Time(int h, int m, int s) :

hours(h), minutes(m), seconds(s) {

}

Time::Time() : hours(0), minutes(0), seconds(0) {

}

Time::~Time() {

}

void Time::print() const {

cout << hours << ":" << minutes << ":" << seconds << endl;

}

void Time::tick() {

seconds++;

normalize();

}

void Time::normalize() {

if (!(seconds < 60 && seconds >= 0)) {

seconds -= 60;

minutes++;

}

if (!(minutes < 60 && minutes >= 0)) {

minutes -= 60;

hours++;

}

if (!(hours < 24 && hours >= 0)) {

hours -= 24;

}

}

// Main.cpp

#include "time.h"

#include <iostream>

using namespace std;

int main(void) {

Time time(4, 5, 27);

cout << "Original time : ";

time.print();

for (int i = 0; i < 143500; i++) {

time.tick();

}

cout << "Time after 143500 ticks : ";

time.print();

return 0;

}

텍스트이(가) 표시된 사진

자동 생성된 설명

Question3

#include <iostream>

using namespace std;

int main(void) {

int num = 0;

int arr[13] = { 0,31,28,31,30,31,30,31,31,30,31,30,31 };

while (1) {

cout << "Enter the month number (1 to 12) : ";

cin >> num;

if (num > 0 && num < 13) {

cout << "There are " << arr[num] << " days in this month." << endl;

break;

}

}

return 0;

}

텍스트이(가) 표시된 사진

자동 생성된 설명

Question4

#include <iostream>

#include <fstream>

using namespace std;

int main(void) {

ifstream file;

file.open("integerlist.txt");

if (!file) {

cout << "Error. Input file cannot be opened" << endl;

cout << "The program is terminated";

return 0;

}

int data = 0, num;

int frequency[10] = { 0 };

while (!file.eof()) {

file >> num;

if (num >= 0 && num < 10) {

data++;

frequency[num]++;

}

}

cout << "There are " << data << " valid data items." << endl;

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

cout << i << " ";

for (int j = 0; j < frequency[i]; j++) {

cout << "\*";

}

cout << " " << frequency[i] << endl;

}

file.close();

return 0;

}

텍스트이(가) 표시된 사진

자동 생성된 설명

Question5

#include <iostream>

#include <iomanip>

using namespace std;

void rowTransform(int\*\*, int\*, int, int);

void colTransform(int\*\*, int\*, int, int);

void printTwoDimensional(int\*\*, int, int);

void printOneDimensional(int\*, int);

int main(void) {

const int N = 2;

const int M = 4;

int\*\* original = new int\* [N];

for (int i = 0; i < N; i++) {

original[i] = new int[M];

}

for (int i = 0; i < N; i++) {

for (int j = 0; j < M; j++) {

original[i][j] = i \* 10 + j;

}

}

int\* rowarr = new int[N \* M];

int\* colarr = new int[N \* M];

cout << "Original Array" << endl;

printTwoDimensional(original, N, M);

cout << "Row-Transformed Array : ";

rowTransform(original, rowarr, N, M);

printOneDimensional(rowarr, N\*M);

cout<<endl;

cout << "Column-Transformed Array : ";

colTransform(original, colarr, N, M);

printOneDimensional(colarr, N \* M);

for (int i = 0; i < N; i++) {

delete[] original[i];

}

delete[] original;

delete[] rowarr;

delete[] colarr;

return 0;

}

void rowTransform(int\*\* original, int\* rowarr, int N, int M) {

for (int i = 0; i < N; i++) {

for (int j = 0; j < M; j++) {

rowarr[i \* M + j] = original[i][j];

}

}

}

void colTransform(int\*\* original, int\* colarr, int N, int M) {

for (int i = 0; i < N; i++) {

for (int j = 0; j < M; j++) {

colarr[j \* 2 + i] = original[i][j];

}

}

}

void printTwoDimensional(int\*\* arr, int N, int M) {

for (int i = 0; i < N; i++) {

for (int j = 0; j < M; j++) {

cout << setw(2) << arr[i][j] << " ";

}

cout << endl;

}

}

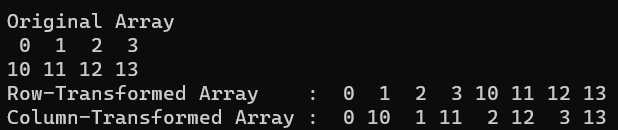
void printOneDimensional(int\* arr, int NM) {

for (int i = 0; i < NM; i++) {

cout << setw(2) << arr[i] << " ";

}

}



Question6

#include <iostream>

using namespace std;

void reverse(int\*, int);

int main(void) {

int arr[5] = { 10,11,12,13,14 };

reverse(arr, 5);

cout << "Reversed array : ";

for (int i = 0; i < 5; i++) {

cout << \*(arr + i) << " ";

}

return 0;

}

void reverse(int\* arr, int n) {

const int k = n;

int\* arr2;

arr2 = new int[k];

for (int i = 0; i < k; i++) {

arr2[i] = arr[k - i - 1];

}

for (int i = 0; i < k; i++) {

arr[i] = arr2[i];

}

delete[] arr2;

}

