#include <iostream>

#include <cmath>

#include <cstdlib>

#include <fstream>

#include <string>

using namespace std;

void swap(int& a, int& b) //прописую функцію swap, яка в подальшому буде використовуватись в сортуванні бульбашкобю, зміннює місцями значення двох змінних

{

int temp = a;

a = b;

b = temp;

}

int randomnumber() //прописую функцію, яку далі буду використовувати для задання рандомного значення в межах від 0 до 9(для цього спеціально беру остачу від ділення рандомного числа,

//що буде в межах від 0 до приблизно 32600, на 10

{

srand(time(nullptr));

int x = rand() % 10;

return x;

}

void bubblesort(int\*\* array, int size, int extra = 889) //прописую функцію бабл-сортінга, що буде послідовно сортувати за спаданням кожен рядок масива, використовую раніше прописану функцію swap

{

for (int k = 0; k < 10; k++)

{

for (int i = 0; i < size - 1; i++)

{

for (int j = 0; j < size - i - 1; j++)

{

if (array[k][j] < array[k][j + 1])

{

swap(array[k][j], array[k][j + 1]);

}

}

}

}

}

void bubblesort(double\*\* array, int size, double extra = 889) //прописую перевантажену функцію для типу doublе(вона ніде не буде використовуватись, але потрібна за умовою)

{

for (int k = 0; k < 10; k++)

{

for (int i = 0; i < size - 1; i++)

{

for (int j = 0; j < size - i - 1; j++)

{

if (array[k][j] > array[k][j + 1])

{

swap(array[k][j], array[k][j + 1]);

}

}

}

}

}

int\*\* matrix\_create() // створюю функцію типу int\*\*(подвійний поінтер(поінтер на поінтер)) типу int, для того, щоб функція повертала двовимірний масив)

{ //функція взята з розрахункової роботи, і до неї я ще додав множник 1000, щоб на виході отримувати адекватні цілі значення

int\*\* matrix = new int\* [10];

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

{

matrix[i] = new int[10];

}

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

{

for (int j = 0; j < 10; j++)

{

matrix[i][j] = 1000 \* (sin(pow(i, 3)) + pow(cos(j), 2));

}

}

return matrix;

}

int\* sumcount(int\*\* array)//створюю функцію, яка рахує суму 4 рандомних елементів в кожному рядку(індекси задаю через раніше прописану рандомну функцію), і повертає масив з 10 даних сум

{

int \*sum = new int[10];

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

{

int index1 = randomnumber();

int index2 = randomnumber();

int index3 = randomnumber();

int index4 = randomnumber();

sum[i] = array[i][index1] + array[i][index2] + array[i][index3] + array[i][index4];

}

return sum;

}

bool createfile(int array[10]) //функція за допомогою пакета fstream створює файл і записує в кожний рядок значення суми 4 рандомних елементів кожного рядка нашої матриці

{ //якщо операція успішна, то функція повертає true, інакше - false

ofstream out("new.txt");

if (!out)

{

return false;

}

else

{

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

out << array[i] << endl;

}

}

int openfile() //за допомогою пакета fstream відкриваю щойно записаний файл і зчитую з нього построчно кожне число

{

ifstream in("new.txt");

string str;

long double multiply = 1; //для коректного зчитування кожного рядка я зчитую числа у форматі стрічок за допомогою функції getline

while (getline(in, str))

{

long double z = stoi(str); //далі я конвертую дані стрічки назад в числа за допомогою стандартної функції stoi(string to integer)

multiply = multiply \* z/1000; //під час множення я кожний раз ділю результат на 1000, оскільки без цього добуток виходить дуже великий і не може бути записаний навіть в long long

}

int multiint = multiply; //відбрасую дробову частину, яка могла виникнути внаслідок добутку(ми ділили кожний множник на 1000)

return multiint; //функція повертає результат, який в 10^30(10 в 30 степені) разів менший за реальний добуток

}

void removefile() //функція видаляє створений файл

{

remove("new.txt");

cout << "File was successfully removed";

}

int main()

{

srand(time(nullptr)); //функція створює реальний генератор рандомних чисел з прив'язкою до часу

const int a = 889;

int\*\* testarray = matrix\_create(); //створюємо двовимірний масив через дабл-поінтер і ініціалізуємо його за допомогою раніше написаної функції matrix\_create()

cout << "Our new handmade matrix is: " << endl;

for (int i = 0; i < 10; i++) //виводимо на екран утворену матрицю

{

for (int j = 0; j < 10; j++)

{

cout << testarray[i][j] << " ";

}

cout << endl;

}

bubblesort(testarray, 10); //сортуємо кожен рядок матриці за спаданням за допомогою раніше написаної функції

cout << endl;

cout << "Sorted array is: " << endl; //виводимо посортовану матрицю

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

{

for (int j = 0; j < 10; j++)

{

cout << testarray[i][j] << " ";

}

cout << endl;

}

cout << endl;

int\* sumarray = sumcount(testarray); //утворюю масив і ініціалізую його 10 сумами 4 рандомно взятих елементів кожного рядка нашої матриці

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

{

cout << "The sum of row " << i + 1 << " is " << sumarray[i] << endl; //виводжу масив даних 10 сум на екран

}

createfile(sumarray); //створюю файл і вписую в нього раніше знайдені суми 4 випаддкових елементів в кожному рядку

if (createfile(sumarray)) //якщо операція успішна, то виводимо повідомлення про успішність

{

cout << "File was successfulle created" << endl;

}

else //якщо операція провалена, виводимо повідомлення про помилку

{

cout << "Error while creating" << endl;

}

int mul = openfile(); //знаходимо число, що буде в 10^30 разів менше за фактичний добуток чисел, вписаних у файл

cout << "The multiply is " << mul << "\*10^30" << endl; //виводимо добуток на екран у форматі "mul \* 10^30"

removefile(); //видаляємо файл

}