#include<iostream>

#include<fstream>

#include<string>

using namespace std;

// out write

// in read

// app

#pragma region WriteReadToTextFile

//void dataWriteToFile(string filename, string text) {

// ofstream fout(filename, ios::out);

// if (fout.is\_open()) {

// fout << text;

// }

// else {

// cout << filename << " does not exists" << endl;

// }

// fout.close();

//}

//

//

//void dataWriteToFileApp(const string& filename, const string& text) {

// ofstream fout(filename, ios::app);

// if (fout.is\_open()) {

// fout << text;

// }

// else {

// cout << filename << " does not exists" << endl;

// }

// fout.close();

//}

//

//

//string GetDataFromFile(const string& filename) {

// ifstream fin(filename, ios::in);

// string result = "";

// if (fin.is\_open()) {

// //fin >> result;

// //getline(fin, result);

//

// string data = "";

// while (!fin.eof()) {

// /\*fin >> data;\*/

// getline(fin, data);

// result.append(data + "\n");

// }

//

// }

// else {

// throw string("This file : " + filename + " does not exists");

// }

// fin.close();

// return result;

//}

//

//

//void main() {

// //dataWriteToFile("mario.txt", "Necesen");

// //dataWriteToFileApp("mario.txt", "\nNecesen");

// try

// {

//

// string result = GetDataFromFile("elvin.txt");

// cout << result << endl;

// }

// catch (const string&ex)

// {

// cout << "ERROR : " << ex << endl;

// }

//}

#pragma endregion

#pragma region WriteAndReadBinary

void writeDataToBinaryFile() {

ofstream fout("obj.bin", ios\_base::binary | ios\_base::app);

int obj[5]{ 11,22,33,44,55 };

if (fout.is\_open()) {

fout.write((char\*)obj, sizeof(int) \* 5);

}

fout.close();

}

void readDataFromBinaryFile() {

ifstream fin("obj.bin", ios\_base::binary);

int arr[10]{};

if (fin.is\_open()) {

fin.read((char\*)arr, sizeof(int) \* 10);

fin.close();

for (size\_t i = 0; i < 10; i++)

{

cout << arr[i] << " ";

} cout << endl;

}

}

void main() {

//writeDataToBinaryFile();

readDataFromBinaryFile();

}

#pragma endregion