/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\* Laba 5

\*\* Matviychuk Bohdan

\*\* IO - 91

\*\* 1.7 b = (A\*SORT(C))

\*\* 2.20 MD = MA + MB – MC

\*\* 3.18 p = MAX(SORT(MS) + MA\*MB)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include "stdafx.h"

#include <iostream>

#include <conio.h>

#include <windows.h>

using std::cout;

using std::endl;

const int n = 10;

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*

void VectorInput(int V[n])

{

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

{

V[i] = 1;

}

}

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*

void MatrixInput(int M[n][n])

{

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

{

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

{

M[i][j] = 1;

}

}

}

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*

void VectorOutput(int V[n])

{

if (n <= 10)

{

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

{

cout<<V[i]<<" ";

}

cout<<endl;

}

}

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*

void MatrixOutput(int M[n][n])

{

if (n < 10)

{

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

{

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

{

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

}

cout<<endl;

}

cout<<endl;

}

}

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*

void SortVector(int V[])

{

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

{

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

{

if (V[j] > V[j + 1])

{

int z = V[j];

V[j] = V[j + 1];

V[j + 1] = z;

}

}

}

}

void SortMatrix(int V[][])

{

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

{

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

{

if (V[j][i] > V[j + 1][i])

{

int z = V[j][i];

V[j][i] = V[j + 1][i];

V[j + 1][i] = z;

}

}

}

}

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*

// 1.7 b = (A\*SORT(C))

void GoFunc1()

{

cout<<"Func1 started"<<endl;

int A[n];

int b=0;

int C[n];

VectorInput(A);

VectorInput(C);

SortVector(C);

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

{

b=b+A[i]\*C[i];

}

cout<<b<<endl;

cout<<"Func1 finished."<<endl;

}//GoFunc1

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*

////2.20 MD = MA + MB – MC

void GoFunc2()

{

cout<<"Func2 started"<<endl;

int MA[n][n];

int MB[n][n];

int MC[n][n];

int MD[n][n];

MatrixInput(MA);

MatrixInput(MB);

MatrixInput(MC);

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

{

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

{

MD[i][j]=MA[i][j]+MB[i][j]-MC[i][j];

}

}

MatrixOutput(MD);

cout<<"Func2 finished."<<endl;

}//GoFunc2

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*

//3.18 p = MAX(SORT(MS) + MA\*MB)

void GoFunc3()

{

cout<<"Func3 started"<<endl;

int p;

int MA[n][n];

int MB[n][n];

int MS[n][n];

MatrixInput(MA);

MatrixInput(MB);

MatrixInput(MS);

SortMatrix(MS);

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

{

MS[i][j] = 1;

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

{

MS[i][j] += MA[i][j] \* MB[i][j];

}

int p = MS[0][0];

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

{

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

{

if (p < MS[i][j])

{

p = MS[i][j];

}

}

}

}

cout<<p<<endl;

cout<<"Func3 finished."<<endl;

}//GoFunc3

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*

void main()

{

cout<<"\* - Laba5 started"<<endl;

DWORD f1id, f2id, f3id;

HANDLE F1 = CreateThread(NULL, 97000, (LPTHREAD\_START\_ROUTINE)GoFunc1, NULL, CREATE\_SUSPENDED, &f1id);

HANDLE F2 = CreateThread(NULL, 588001000, (LPTHREAD\_START\_ROUTINE)GoFunc2, NULL, CREATE\_SUSPENDED, &f2id);

HANDLE F3 = CreateThread(NULL, 588065000, (LPTHREAD\_START\_ROUTINE)GoFunc3, NULL, CREATE\_SUSPENDED, &f3id);

SetThreadPriority(F1, THREAD\_PRIORITY\_HIGHEST);

SetThreadPriority(F2, THREAD\_PRIORITY\_NORMAL);

SetThreadPriority(F3, THREAD\_PRIORITY\_LOWEST);

ResumeThread(F1);

ResumeThread(F2);

ResumeThread(F3);

WaitForSingleObject(F1, INFINITE);

WaitForSingleObject(F2, INFINITE);

WaitForSingleObject(F3, INFINITE);

cout<<"\* - Laba5 finished."<<endl;

getchar();

}