НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ФАКУЛЬТЕТ ІНФОРМАТИКИ І ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

**Лабораторна робота №1**

з дисципліни **«**Паралельні та розподілені обчислення**»**

Виконав:

студент 3 курсу гр. ІО-42

Кочетов Данило

№ ЗК 4213

Перевірив:

Долголенко О. М.

Київ 2016 р.

***Завдання:***

1.13; 2.13; 3.13

F1: C = A\*(MA\*ME) + B + D

F2: ML = MIN(MF)\*MG + MAX(MH) \* (MK\*MF)

F3: T = (MO\*MP)\*S + MR\*SORT(S)

***Лістинг програми:***

*----------------------------------------*

***data.ads***

*----------------------------------------*

*with Ada.Text\_IO, Ada.Long\_Integer\_Text\_IO, System.Multiprocessors, Ada.Numerics.Discrete\_Random;*

*use Ada.Text\_IO, Ada.Long\_Integer\_Text\_IO, System.Multiprocessors;*

*generic N: Long\_Integer;*

*package Data is*

*type Vector is private;*

*type Matrix is private;*

*procedure Read\_Vector(A: out Vector);*

*procedure Read\_Matrix(MA: out Matrix);*

*procedure Fill\_Vector\_1(A: in out Vector);*

*procedure Fill\_Matrix\_1(MA: in out Matrix);*

*function Multiply\_Matrices(MA, MB: in Matrix) return Matrix;*

*function Multiply\_Vector\_Matrix(A: in Vector; MB: in Matrix) return Vector;*

*function Multiply\_Scalar\_Matrix(A: in Long\_Integer; MB: in Matrix) return Matrix;*

*function Sum\_Vectors(A, B: in Vector) return Vector;*

*function Sum\_Matrices(MA, MB: in Matrix) return Matrix;*

*function Min\_Vector(A: in Vector) return Long\_Integer;*

*function Min\_Matrix(MA: in Matrix) return Long\_Integer;*

*function Max\_Vector(A: in Vector) return Long\_Integer;*

*function Max\_Matrix(MA: in Matrix) return Long\_Integer;*

*function Sort\_Vector(A: in Vector) return Vector;*

*procedure Print\_Vector(A: in Vector);*

*procedure Print\_Matrix(MA: in Matrix);*

*procedure Funcs(C: out Vector; A: in Vector; MA, ME: in Matrix; B, D: in Vector; ML: out Matrix; MF, MG, MH, MK: in Matrix; T: out Vector; MO, MP: in Matrix; S: in Vector; MR: in Matrix);*

*private*

*type Vector is array(1..N) of Long\_Integer;*

*type Matrix is array(1..N, 1..N) of Long\_Integer;*

*end Data;*

*----------------------------------------*

***data.adb***

*----------------------------------------*

*package body Data is*

*procedure Read\_Vector(A: out Vector) is*

*begin*

*for i in 1..N loop*

*Get(A(i));*

*end loop;*

*end Read\_Vector;*

*procedure Read\_Matrix(MA: out Matrix) is*

*begin*

*for i in 1..N loop*

*for k in 1..N loop*

*Get(MA(i, k));*

*end loop;*

*end loop;*

*end Read\_Matrix;*

*procedure Fill\_Vector(A: in out Vector) is*

*subtype r is range 1..20;*

*package Random is new Ada.Numerics.Discrete\_Random(r);*

*use Random;*

*G: Generator;*

*D: Dice;*

*begin*

*Reset(G);*

*for i in 1..N loop*

*A(i) := Random(G);*

*end loop;*

*end Fill\_Vector;*

*procedure Fill\_Matrix(MA: in out Matrix) is*

*subtype r is range 1..20;*

*package Random is new Ada.Numerics.Discrete\_Random(r);*

*use Random;*

*G: Generator;*

*D: Dice;*

*begin*

*Reset(G);*

*for i in 1..N loop*

*for k in 1..N loop*

*MA(i, k) := Random(G);*

*end loop;*

*end loop;*

*end Fill\_Matrix;*

*function Multiply\_Matrices(MA, MB: in Matrix) return Matrix is*

*res: Matrix;*

*begin*

*for i in 1..N loop*

*for k in 1..N loop*

*res(i, k) := 0;*

*for j in 1..N loop*

*res(i, k) := res(i, k) + MA(i, j) \* MB(j, k);*

*end loop;*

*end loop;*

*end loop;*

*return res;*

*end Multiply\_Matrices;*

*function Multiply\_Vector\_Matrix(A: in Vector; MB: in Matrix) return Vector is*

*res: Vector;*

*begin*

*for i in 1..N loop*

*res(i) := 0;*

*for j in 1..N loop*

*res(i) := res(i) + MB(i, j) \* A(j);*

*end loop;*

*end loop;*

*return res;*

*end Multiply\_Vector\_Matrix;*

*function Multiply\_Scalar\_Matrix(A: in Long\_Integer; MB: in Matrix) return Matrix is*

*res: Matrix;*

*begin*

*for i in 1..N loop*

*for k in 1..N loop*

*res(i, k) := A \* MB(i, k);*

*end loop;*

*end loop;*

*return res;*

*end Multiply\_Scalar\_Matrix;*

*function Sum\_Vectors(A, B: in Vector) return Vector is*

*res: Vector;*

*begin*

*for i in 1..N loop*

*res(i) := A(i) + B(i);*

*end loop;*

*return res;*

*end Sum\_Vectors;*

*function Sum\_Matrices(MA, MB: in Matrix) return Matrix is*

*res: Matrix;*

*begin*

*for i in 1..N loop*

*for k in 1..N loop*

*res(i, k) := MA(i, k) + MB(i, k);*

*end loop;*

*end loop;*

*return res;*

*end Sum\_Matrices;*

*function Min\_Vector(A: in Vector) return Long\_Integer is*

*res: Long\_Integer;*

*begin*

*res := A(1);*

*for i in 2..N loop*

*if res < A(i) then*

*res := A(i);*

*end if;*

*end loop;*

*return res;*

*end Min\_Vector;*

*function Min\_Matrix(MA: in Matrix) return Long\_Integer is*

*res: Long\_Integer;*

*begin*

*res := MA(1, 1);*

*for i in 1..N loop*

*for k in 1..N loop*

*if res < MA(i, k) then*

*res := MA(i, k);*

*end if;*

*end loop;*

*end loop;*

*return res;*

*end Min\_Matrix;*

*function Max\_Vector(A: in Vector) return Long\_Integer is*

*res: Long\_Integer;*

*begin*

*res := A(1);*

*for i in 2..N loop*

*if res > A(i) then*

*res := A(i);*

*end if;*

*end loop;*

*return res;*

*end Max\_Vector;*

*function Max\_Matrix(MA: in Matrix) return Long\_Integer is*

*res: Long\_Integer;*

*begin*

*res := MA(1, 1);*

*for i in 1..N loop*

*for k in 1..N loop*

*if res > MA(i, k) then*

*res := MA(i, k);*

*end if;*

*end loop;*

*end loop;*

*return res;*

*end Max\_Matrix;*

*function Sort\_Vector(A: in Vector) return Vector is*

*res: Vector;*

*t: Long\_Integer;*

*begin*

*res := A;*

*for i in 1..N loop*

*for k in 1..N-i loop*

*if res(i) > res(i + 1) then*

*t := res(i);*

*res(i) := res(i + 1);*

*res(i + 1) := t;*

*end if;*

*end loop;*

*end loop;*

*return res;*

*end Sort\_Vector;*

*procedure Print\_Vector(A: in Vector) is*

*begin*

*for i in 1..N loop*

*Put(A(i));*

*end loop;*

*New\_Line;*

*end Print\_Vector;*

*procedure Print\_Matrix(MA: in Matrix) is*

*begin*

*for i in 1..N loop*

*for k in 1..N loop*

*Put(MA(i, k));*

*end loop;*

*New\_Line;*

*end loop;*

*end Print\_Matrix;*

*procedure Funcs(C: out Vector; A: in Vector; MA, ME: in Matrix; B, D: in Vector; ML: out Matrix; MF, MG, MH, MK: in Matrix; T: out Vector; MO, MP: in Matrix; S: in Vector; MR: in Matrix) is*

*task Func1 with CPU=>1 is*

*pragma Priority(10);*

*pragma Storage\_Size(300\_000\_000);*

*end Func1;*

*task Func2 with CPU=>4 is*

*pragma Priority(9);*

*pragma Storage\_Size(300\_000\_000);*

*end Func2;*

*task Func3 with CPU=>3 is*

*pragma Priority(8);*

*pragma Storage\_Size(300\_000\_000);*

*end Func3;*

*task body Func1 is*

*begin*

*Put\_Line("Task 1 begin");*

*C := Sum\_Vectors(Multiply\_Vector\_Matrix(A, Multiply\_Matrices(MA, ME)), Sum\_Vectors(B, D));*

*Put\_Line("Task 1 end");*

*end Func1;*

*task body Func2 is*

*begin*

*Put\_Line("Task 2 begin");*

*ML := Sum\_Matrices(Multiply\_Scalar\_Matrix(Min\_Matrix(MF), MG), Multiply\_Scalar\_Matrix(Max\_Matrix(MH), Multiply\_Matrices(MK, MF)));*

*Put\_Line("Task 2 end");*

*end Func2;*

*task body Func3 is*

*begin*

*Put\_Line("Task 3 begin");*

*T := Sum\_Vectors(Multiply\_Vector\_Matrix(S, Multiply\_Matrices(MO, MP)), Multiply\_Vector\_Matrix(Sort\_Vector(S), MR));*

*Put\_Line("Task 3 end");*

*end Func3;*

*begin*

*null;*

*end Funcs;*

*end Data;*

*----------------------------------------*

***lab1.adb***

*----------------------------------------*

*with Data, Ada.Text\_IO, Ada.Integer\_Text\_IO;*

*use Ada.Text\_IO, Ada.Integer\_Text\_IO;*

*procedure Lab1 is*

*N: constant Long\_Integer := 1000;*

*package ConcreteData is new Data(N);*

*use ConcreteData;*

*A, B, C, D, T, S: Vector;*

*MA, ME, ML, MF, MG, MH, MK, MO, MP, MR: Matrix;*

*begin*

*Put\_Line("Enter vector A:");*

*Fill\_Vector(A);*

*Put\_Line("Enter vector B:");*

*Fill\_Vector(B);*

*Put\_Line("Enter vector D:");*

*Fill\_Vector(D);*

*Put\_Line("Enter maxtrix MA:");*

*Fill\_Matrix(MA);*

*Put\_Line("Enter maxtrix ME:");*

*Fill\_Matrix(ME);*

*Put\_Line("Enter maxtrix MF:");*

*Fill\_Matrix(MF);*

*Put\_Line("Enter maxtrix MG:");*

*Fill\_Matrix(MG);*

*Put\_Line("Enter maxtrix MH:");*

*Fill\_Matrix(MH);*

*Put\_Line("Enter maxtrix MK:");*

*Fill\_Matrix(MK);*

*Put\_Line("Enter maxtrix MO:");*

*Fill\_Matrix(MO);*

*Put\_Line("Enter maxtrix MP:");*

*Fill\_Matrix(MP);*

*Put\_Line("Enter vector MS:");*

*Fill\_Vector(S);*

*Put\_Line("Enter maxtrix MR:");*

*Fill\_Matrix(MR);*

*Funcs(C, A, MA, ME, B, D, ML, MF, MG, MH, MK, T, MO, MP, S, MR);*

*Put\_Line("F1 =");*

*Print\_Vector(C);*

*Put\_Line("F2 =");*

*Print\_Matrix(ML);*

*Put\_Line("F3 =");*

*Print\_Vector(T);*

*end Lab1;*