Міністерство освіти і науки, молоді та спорту України

Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**Лабораторна робота №2**

З дисципліни «ПРО»

Виконав: Перевірив:

Студент групи ІО-21 доц. Корочкін О.В.

Коноз А.О.

Дата здачі\_\_\_\_\_\_\_\_\_\_\_\_\_

Захищено з балом\_\_\_\_\_

Київ 2014

**Завдання**

1.23 2.13 3.17

1. F1: E:=A+B+C+D(MA\*MZ).

2. F2: MZ:=Min(MA)\*MB+Max(MT)\*(MX\*MZ).

3. F3: d:=Min(A\*Trans(MB\*MM)+B\*Sort(MC).

**Лістинг коду**

GNAT GPL 2014 (20140331)

Copyright 1992-2014, Free Software Foundation, Inc.

Compiling: lab2.adb (source file time stamp: 2014-09-19 17:17:54)

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

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

3. with System.Multiprocessors; use System.Multiprocessors;

4.

5. -------------Основна програма---------------------------------

6. --ПРО Lab2

7. --Коноз Андрій Олександрович

8. --ІО-21

9. --19.09.14

10. --F1: E:=A+B+C+D(MA\*MZ)

11. --F2: MZ:=Min(MA)\*MB+Max(MT)\*(MX\*MZ)

12. --F3: d:=Min(A\*Trans(MB\*MM)+B\*Sort(MC))

13.

14. procedure Lab2 is

15. --Розмірність векторів і матриць

16. n : Integer :=3;

17. --Ініціалізація пакету

18. package Lab1Data is new Data(n);

19. use Lab1Data;

20. --задання номерів процесора

21. CPU\_1 : CPU\_Range:=0;

22. CPU\_2 : CPU\_Range:=1;

23. --Оголошення задачі для виконання функції F1

24. --Задання пріоритету, назви задачі, виділення стеку,

25. -- номер процесору на якому буде відбуватися виконання

26. task T1 is

27. pragma Priority(1);

28. pragma Task\_Name ("T1");

29. pragma Storage\_Size(1000);

30. pragma CPU (CPU\_2);

31. end T1;

32. --Тіло задачі

33. task body T1 is

34.

35. A,B,C,D,E: Vect;

36. MA,MZ: Matrix;

37.

38. begin

39. --Затримка виконання на 10с

40. delay 10.0;

41. --Введення даних для F1

42.

43. Put("Enter vector A for F1");

44. New\_Line;

45. readVect(A);

46. New\_Line;

47.

48. Put("Enter vector B for F1");

49. New\_Line;

50. readVect(B);

51. New\_Line;

52.

53. Put("Enter vector C for F1");

54. New\_Line;

55. readVect(C);

56.

57. New\_Line;

58. Put("Enter vector D for F1");

59. New\_Line;

60. readVect(D);

61.

62. New\_Line;

63. Put("Enter MA for F1");

64. New\_Line;

65. readMatr(MA);

66.

67. New\_Line;

68. Put("Enter MZ for F1");

69. New\_Line;

70. readMatr(MZ);

71. --Виконання функції F1

72. F1(A,B,C,D,MA, MZ,E);

73. --Виведення результатів функції

74. Put("F1 = ");

75. showVect(E);

76. New\_Line;

77. put("TF1 finifed");

78.

79. end T1;

80. --Оголошення задачі для виконання функції F2

81. --Задання пріоритету, назви задачі, виділення стеку,

82. -- номер процесору на якому буде відбуватися виконання

83. task T2 is

84. pragma Priority(5);

85. pragma Task\_Name ("T2");

86. pragma Storage\_Size(1000);

87. pragma CPU (CPU\_1);

88. end T2;

89.

90. task body T2 is

91. MA,MB,MT,MX,MZ,ME: Matrix;

92.

93. begin

94. --затримка виконання задачі на 5с

95. delay 5.0;

96. --Введення даних для F2

97.

98. New\_Line;

99. Put("Enter MA for F2");

100. New\_Line;

101. readMatr(MA);

102.

103. New\_Line;

104. Put("Enter MB for F2");

105. New\_Line;

106. readMatr(MB);

107.

108. New\_Line;

109. Put("Enter MX for F2");

110. New\_Line;

111. readMatr(MX);

112.

113. New\_Line;

114. Put("Enter MZ for F2");

115. New\_Line;

116. readMatr(MZ);

117.

118. New\_Line;

119. Put("Enter MT for F2");

120. New\_Line;

121. readMatr(MT);

122. --Виконання функції F2

123. F2(MA,MB,MT,MX,MZ,ME);

124. --Виведення результатів функції

125. Put("F2 = ");

126. showMatr(ME);

127. New\_Line;

128. Put("TF2 finished");

129.

130. end T2;

131. --Оголошення задачі для виконання функції F3

132. --Задання пріоритету, назви задачі, виділення стеку,

133. -- номер процесору на якому буде відбуватися виконання

134. task T3 is

135. pragma Priority(8);

136. pragma Task\_Name ("3");

137. pragma Storage\_Size(1000);

138. pragma CPU (CPU\_1);

139. end T3;

140.

141. task body T3 is

142. MB,MM,MC: Matrix;

143. A,B: Vect;

144. y: Integer;

145.

146. begin

147. delay 2.0;

148. --Введення даних для F3

149.

150. Put("Enter vector A for F3");

151. New\_Line;

152. readVect(A);

153.

154. New\_Line;

155. Put("Enter vector B for F3");

156. New\_Line;

157. readVect(B);

158.

159. New\_Line;

160. Put("Enter MB for F3");

161. New\_Line;

162. readMatr(MB);

163.

164. New\_Line;

165. Put("Enter MM for F3");

166. New\_Line;

167. readMatr(MM);

168.

169. New\_Line;

170. Put("Enter MC for F3");

171. New\_Line;

172. readMatr(MC);

173. --Виконання функції F3

174. F3(A,B,MB, MM,MC,y);

175. --Виведення результатів функції

176. Put("F3 = ");

177. Put(y);

178.

179. New\_Line;

180. put("TF3 finished");

181. end T3;

182.

183. begin

184. put("Lab 2");

185. end Lab2;

186.

186 lines: No errors

GNAT GPL 2014 (20140331)

Copyright 1992-2014, Free Software Foundation, Inc.

Compiling: data.adb (source file time stamp: 2014-09-19 14:45:08)

1. with Ada.Text\_IO, Ada.Integer\_Text\_IO;

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

3. with Ada.Streams; use Ada.Streams;

4. ---------Тіло пакету Data--------------

5. package body Data is

6. --Функція виконує додавання двох векторів

7. function AddVect (A, B :Vect) return Vect is

8. g : Vect;

9. begin

10. for index in 1..N loop

11. g(index):=A(index)+B(index);

12. end loop;

13. return g;

14. end AddVect;

15. --Функція виконує множення двох матриць

16. function MultMatr (A, B : Matrix) return Matrix is

17. Cell : Integer;

18. Result : Matrix;

19. begin

20. for i in 1..N loop

21. for j in 1..N loop

22. Cell := 0;

23. for k in 1..N loop

24. Cell := Cell + A(i,k)\*B(k,j);

25. end loop;

26. Result(i,j):= Cell;

27. end loop;

28. end loop;

29. return Result;

30. end MultMatr;

31. --Функція транспонує матрицю

32. function Transp (A : Matrix) return Matrix is

33. Result : Matrix;

34. begin

35. for i in 1..N loop

36. for j in 1..N loop

37. Result(i,j) := A(j,i);

38. end loop;

39. end loop;

40. return Result;

41. end Transp;

42. --Функція виконує пошук максимального елементу матриці

43. function SearchMax(A : Matrix) return Integer is

44. Var : Integer := A(1,1);

45. begin

46. for i in 1..N loop

47. for j in 1..N loop

48. if A(i,j) > Var then

49. Var := A(i,j);

50. end if;

51. end loop;

52. end loop;

53. return Var;

54. end SearchMax;

55. --Функція виконує пошук мінімального елемента матриці

56. function SearchMin(A : Matrix) return Integer is

57. Var : Integer := A(1,1);

58. begin

59. for i in 1..N loop

60. for j in 1..N loop

61. if A(i,j) < Var then

62. Var := A(i,j);

63. end if;

64. end loop;

65. end loop;

66. return Var;

67. end SearchMin;

68. --Функція виконує пошук мінімального елемента в векторі

69. function SearchVectMin(A: Vect) return Integer is

70. min : Integer;

71. begin

72. min := A(1);

73. for i in 1..N loop

74. if A(i) < min then

75. min := A(i);

76. end if;

77. end loop;

78. return min;

79. end SearchVectMin;

80. --Функція виконує додавання двох матриць

81. function AddMatr(A, B : Matrix) return Matrix is

82. Result : Matrix;

83. begin

84. for i in 1..N loop

85. for j in 1..N loop

86. Result(i,j) := A(i,j) + B(i,j);

87. end loop;

88. end loop;

89. return Result;

90. end AddMatr;

91. --Функція виконує множення вектора на матрицю

92. function MultVectMatr(A : Vect; B : Matrix) return Vect is

93. Cell : Integer;

94. Result : Vect;

95. begin

96. for j in 1..N loop

97. Cell := 0;

98. for k in 1..N loop

99. Cell := Cell + A(k)\*B(k,j);

100. end loop;

101. Result(j):= Cell;

102. end loop;

103. return Result;

104. end MultVectMatr;

105. --Функція виконує сортування вектора за зростанням

106. function SortVect(A : Vect) return Vect is

107. m : Integer;

108. Result: Vect;

109. begin

110. for i in 1..N loop

111. Result(i):= A(i);

112. end loop;

113. for i in reverse 1..N loop

114. for j in 1..(i-1) loop

115. if Result(j) > Result(j+1) then

116. m := Result(j);

117. Result(j):=Result(j+1);

118. Result(j+1):=m;

119. end if;

120. end loop;

121. end loop;

122. return Result;

123. end SortVect;

124. --Функція виконує сортування матриці по спаданню

125. function sortMatr(MA: Matrix) return Matrix is

126. Result : Matrix;

127. m: Integer;

128. begin

129. for i in 1..N loop

130. for j in 1..N loop

131. Result(i,j):=MA(i,j);

132. end loop;

133. end loop;

134.

135. for k in 1..N loop

136. for i in reverse 1..N loop

137. for j in 1..(i-1) loop

138. if Result(k,j) < Result(k,j+1) then

139. m := Result(k,j);

140. Result(k,j):=Result(k,j+1);

141. Result(k,j+1):=m;

142. end if;

143. end loop;

144. end loop;

145. end loop;

146. return Result;

147. end sortMatr;

148. --Функція виконує множення числа на матрицю

149. function MultNumb(a: Integer; MA: Matrix) return Matrix is

150. Result : Matrix;

151. begin

152. for i in 1..N loop

153. for j in 1..N loop

154. Result(i,j):= a\*MA(i,j);

155. end loop;

156. end loop;

157. return Result;

158. end MultNumb;

159. --F1: E:=A+B+C+D(MA\*MZ)

160. procedure F1(A,B,C,D: in Vect; MA, MZ: in Matrix; E : out Vect) is

161. begin

162. E := AddVect(AddVect(A,B),AddVect(C,MultVectMatr(D,MultMatr(MA,MZ))));

163. end F1;

164. --F2: MZ:=Min(MA)\*MB+Max(MT)\*(MX\*MZ)

165. procedure F2(MA,MB,MT,MX,MZ: in Matrix; ME: out Matrix) is

166. begin

167. ME := AddMatr(MultNumb(SearchMin(MA),MB),MultNumb(SearchMax(MT),MultMatr(MX,MZ)));

168. end F2;

169. --F3: d:=Min(A\*Trans(MB\*MM)+B\*Sort(MC))

170. procedure F3(A,B: in Vect; MB, MM,MC: in Matrix; d: out Integer) is

171. begin

172. d:= SearchVectMin(AddVect(MultVectMatr(A,Transp(MultMatr(MB,MM))),MultVectMatr(B,sortMatr(MC))));

173. end F3;

174. --Процедура зчитування вектора з консолі

175. procedure readVect(A : out Vect) is

176. begin

177. for i in 1..N loop

178. Get(A(i));

179. end loop;

180. end readVect;

181. --Процедура відображення вектора в консолі

182. procedure showVect(A : in Vect) is

183. begin

184. for i in 1..N loop

185. Put(A(i));

186. end loop;

187. end showVect;

188. --Процедура зчитування матриці з консолі

189. procedure readMatr(MA: out Matrix) is

190. begin

191. for i in 1..N loop

192. New\_Line;

193. for j in 1..N loop

194. Get(MA(i,j));

195. end loop;

196. end loop;

197. end readMatr;

198. --Процедура відображення матриці в консоль

199. procedure showMatr(MA: in Matrix) is

200. begin

201. for i in 1..N loop

202. New\_Line;

203. for j in 1..N loop

204. Put(MA(i,j));

205. end loop;

206. end loop;

207. end showMatr;

208.

209.

210.

211. end Data;

Compiling: data.ads (source file time stamp: 2014-09-17 05:04:46)

1. -----Специфікація пакету Data----------

2. generic

3. N : in Integer;

4. package Data is

5. --Оголошення приватних типів

6. type Vect is private;

7. type Matrix is private;

8. --F1: E:=A+B+C+D(MA\*MZ)

9. procedure F1(A,B,C,D: in Vect; MA, MZ: in Matrix; E : out Vect);

10. --F2: MZ:=Min(MA)\*MB+Max(MT)\*(MX\*MZ)

11. procedure F2(MA,MB,MT,MX,MZ: in Matrix; ME: out Matrix);

12. --F3: d:=Min(A\*Trans(MB\*MM)+B\*Sort(MC))

13. procedure F3(A,B: in Vect; MB, MM,MC: in Matrix; d: out Integer);

14. --Процедура зчитування вектора з консолі

15. procedure readVect(A : out Vect);

16. --Процедура відображення вектора в консолі

17. procedure showVect(A : in Vect);

18. --Процедура зчитування матриці з консолі

19. procedure readMatr(MA: out Matrix);

20. --Процедура відображення матриці в консоль

21. procedure showMatr(MA: in Matrix);

22. --Визначення приватних типів

23. private

24. type Vect is array(1..N) of Integer;

25. type Matrix is array(1..N, 1..N) of Integer;

26. end Data;

211 lines: No errors