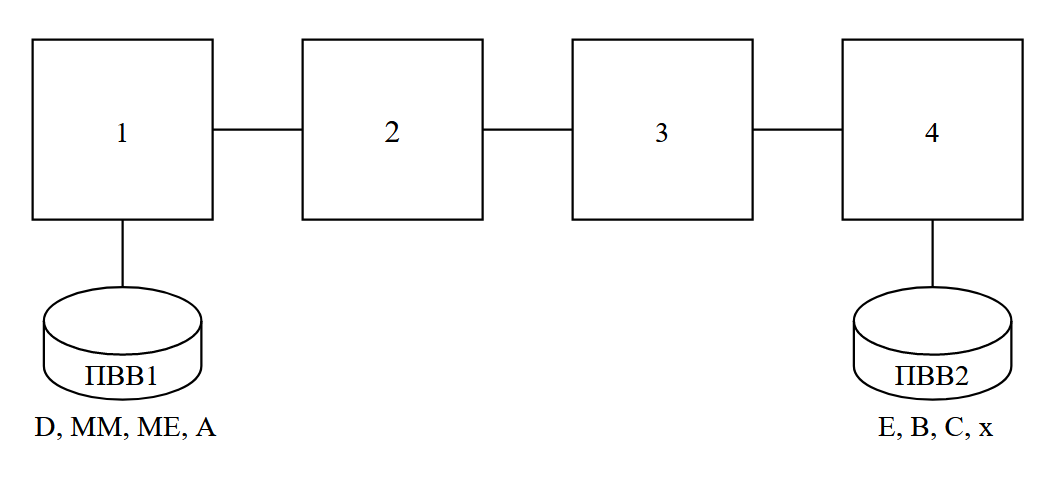
**Завдання**

Мова програмування: Ада

Засоби організації взаємодії потоків: механізм рандеву.

Математична задача: A= sort(D\*(ME\*MM) + (B\*C)\*E\*x

Структура ПКС ЛП:

**Розробка програми**

Етап 1. Побудова паралельного алгоритму

1. ai = (BH\*ZH), i = 1..4

2. a = a + a1, i = 1..4

3. KH = sort(D\*(ME\*MMH))

4. K2H = mergesort(KH, KH)

5. K = mergesort(K2H, K2H)

6. AH = K + a\*EH\*x

Етап 2. Розробка алгоритмів

Задача Т1

1. Ввести D, ME, MM.

2. Передати D, ME, MM3Н задачі Т2.

3. Отримати EH, BН, СH, d від задачі Т2.

4. Обчислити a1 = (BH\*CH).

5. Обчислити KH = sort(D\*(ME\*MMH)).

6. Передати а1, KH задачі Т2.

7. Отримати а, K від Т2.

8. Обчислити AH = K + a\*EH\*x.

9. Отримати А3Н від Т2.

10. Вивести А.

Задача Т2

1. Отримати D, ME, MM3Н від задачі Т1.

2. Передати D, ME, MM2Н задачі Т3.

3. Отримати E2H, B2Н, С2H, d від задачі Т3.

4. Передати EH, BН, СH, d задачі Т1.

5. Отримати а1, КН від Т1.

6. Обчислити a2 = (BH\*ZH).

7. Обчислити KH = sort(D\*(ME\*MMH)).

8. Обчислити а2 = а2 + а1.

9. Обчислити K2H = mergesort(KH, KH)

10. Передати а2, КH, K2H задачі Т3.

11. Отримати а, K від задачі Т3.

12. Обчислити AH = K + a\*EH\*x.

13. Отримати А2Н від Т2.

14. Передати А3Н від Т1.

Задача Т3

1. Отримати D, ME, MM2Н від задачі Т2.

2. Отримати E3H, B3Н, С3H, d від задачі Т4.

3. Передати D, ME, MMН задачі Т4.

4. Передати E2H, B2Н, С2H, d задачі Т2.

5. Обчислити a3 = (BH\*ZH).

6. Обчислити KH = sort(D\*(ME\*MMH)).

7. Отримати а2, KH, K2H від задачі Т2.

8. Обчислити а3 = а3 + а2.

9. Обчислити K2H = mergesort(KH, KH)

10. Передати а3, KH, K2H задачі Т4.

11. Отримати а, K від задачі Т4.

12. Передати а,K задачі Т2.

13. Обчислити AH = K + a\*EH\*x.

14. Отримати АН від задачі Т4.

15. Передати А2Н задачі Т2.

Задача Т4

1. Ввести E, B, C, d.

2. Передати E3H, B3Н, С3H, d задачі Т3.

3. Отримати D, ME, MMН від задачі Т3.

4. Обчислити a4 = (BH\*ZH).

5. Обчислити KH = sort(D\*(ME\*MMH)).

6. Отримати а3, KH, K2H від задачі Т3.

7. Обчислити K2H = mergesort(KH, KH).

8. Обчислити K = mergesort(K2H, K2H).

9. Обчислити а4 = а4 + а3.

10. Обчислити а = а + а4.

11. Передати а, K задачі Т3.

12. Обчислити AH = K + a\*EH\*x.

13. Передати АН задачі Т3.

Етап 3. Розробка структурної схеми взаємодії потоків

DataT1toT2, DataT1toT3, DataT1toT4 - для передачі даних, які вводяться в задачі Т2.

DataT4toT3, DataT4toT2, DataT4toT1 - для передачі даних, які вводяться в задачі Т4.

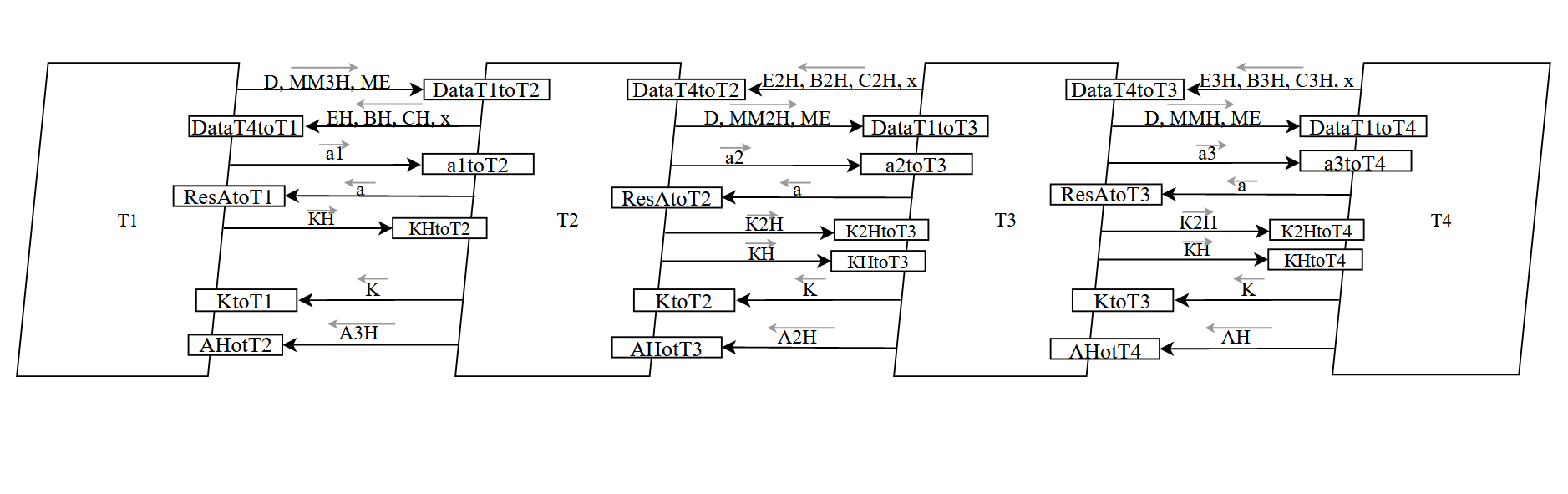
a1toT2, a2toT3, a3toT4 - для передачі частин векторного добутку.

ResAtoT3, ResAtoT2, ResAtoT1 - для передачі результату векторного добутку.

KHotT2, K2HotT3, K2HotT4 - для передачі частин сортування .

KtoT3, KtoT2, KtoT1 - для передачі результату сортування.

AHotT4, A2HotT2, AHotT3 - для передачі частин результуючого вектора А після завершення обчислень.

Рис. 1 – Структурна схема взаємодії задач

Етап 4. Лістинг програми

1. -- Lab #4.

2. -- A = sort(D\*(ME\*MM) + (B\*C)\*E\*x

3. --

4. -- Date

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

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

7. procedure Lab4 is

8. N: Integer:=4;

9. P: Integer:=4;

10. H: Integer:=N/P;

11. type Vector\_All is array(Integer range <>) of Integer;

12. subtype Vector is Vector\_All(1..N);

13. subtype VectorH is Vector\_All(1..1\*H);

14. subtype Vector2H is Vector\_All(1..2\*H);

15. subtype Vector3H is Vector\_All(1..3\*H);

16. type Matrix\_All is array(Integer range <>) of Vector;

17. subtype Matrix is Matrix\_All(1..N);

18. subtype MatrixH is Matrix\_All(1..1\*H);

19. subtype Matrix2H is Matrix\_All(1..2\*H);

20. subtype Matrix3H is Matrix\_All(1..3\*H);

21. procedure InputNumber(x: out Integer) is

22. begin

23. x := 1;

24. end InputNumber;

25. -----Procedure for Output Vector-----

26. procedure VectorInput(V: out Vector) is

27. begin

28. for i in 1..N loop

29. V(i) := 1;

30. end loop;

31. end VectorInput;

32. procedure MatrixInput(M: out Matrix) is

33. begin

34. for i in 1..N loop

35. for j in 1..N loop

36. M(i)(j) := 1;

37. end loop;

38. end loop;

39. end MatrixInput;

40. -----Procedure for Output Vector-----

41. procedure VectorOutput(V: in Vector; str: in String) is

42. begin

43. Put("Vector " & str & ":");

44. for i in 1..N loop

45. put(V(i));

46. Put\_Line(" ");

47. end loop;

48. Put\_Line("");

49. end VectorOutput;

50. -----procedure for KH = D\*(ME\*MMH)-----

51. procedure PreHSort(one : Integer; ash : Integer; D : Vector; ME : Matrix; MMH : MatrixH; KH : out VectorH) is

52. sum1 : Integer;

53. begin

54. for i in one..ash loop

55. for j in one..ash loop

56. sum1:=0;

57. for k in 1..N loop

58. sum1 := sum1 + ME(k)(j) \* MMH(i)(k);

59. KH(i) := D(k) \* sum1;

60. end loop;

61. end loop;

62. end loop;

63. end PreHSort;

64. -----Procedure of calculation AH = K + a\*EH\*x-----

65. procedure Calculate(a : Integer; EH : VectorH; K : Vector; x : Integer; AH : out VectorH) is

66. begin

67. for i in 1..H loop

68. for j in 1..N loop

69. -----calculation AH-----

70. AH(i) := K(j) + a \* EH(i) \* x;

71. end loop;

72. end loop;

73. end Calculate;

74. procedure tasksProcedure is

75. ------------------------------T1------------------------------

76. task T1 is

77. pragma Storage\_Size(300000000);

78. entry DataT4toT1(VCH : in VectorH; VBH : in VectorH; VEH : in VectorH; b : Integer);

79. entry ResAtoT1(a : Integer);

80. entry KtoT1(V: in Vector);

81. entry ResA3HotT2(ResA3H : Vector3H);

82. end T1;

83. ------------------------------T2------------------------------

84. task T2 is

85. pragma Storage\_Size(300000000);

86. entry DataT1toT2(V : in Vector; M : in Matrix; M3H : in Matrix3H);

87. entry DataT4toT2(VC2H : in Vector2H; VB2H : in Vector2H; VE2H : in Vector2H; b : Integer);

88. entry a1toT2(a1 : Integer);

89. entry ResAtoT2(a : Integer);

90. entry KHtoT2(VH : in VectorH);

91. entry KtoT2(V: in Vector);

92. entry ResA2HotT3(ResA2H : Vector2H);

93. end T2;

94. ------------------------------T3------------------------------

95. task T3 is

96. pragma Storage\_Size(300000000);

97. entry DataT1toT3(V : in Vector; M : in Matrix; M2H : in Matrix2H);

98. entry DataT4toT3(VC3H : in Vector3H; VB3H : in Vector3H; VE3H : in Vector3H; b : Integer);

99. entry a2toT3(a2 : Integer);

100. entry ResAtoT3(a : Integer);

101. entry K2HtoT3(V2H : in Vector2H);

102. entry KHtoT3(VH : in VectorH);

103. entry KtoT3(V: in Vector);

104. entry ResAHotT4(ResAH : VectorH);

105. end T3;

106. ------------------------------T4------------------------------

107. task T4 is

108. pragma Storage\_Size(300000000);

109. entry DataT1toT4(V : in Vector; M : in Matrix; MH : in MatrixH);

110. entry a3toT4(a3 : Integer);

111. entry K2HtoT4(V2H : in Vector2H);

112. entry KHtoT4(VH : in VectorH);

113. end T4;

114. ------------------------------T1------------------------------

115. task body T1 is

116. x, a1, resa, e : Integer;

117. D, A, resK: Vector;

118. ME, MM : Matrix;

119. EH, BH, CH, KH, AH : VectorH;

120. begin

121. Put\_Line("T1 started.");

122. --------------------------Enter D, MM, ME--------------------------

123. VectorInput(D);

124. MatrixInput(MM);

125. MatrixInput(ME);

126. ----------------------passing D, ME, MM3H to T2---------------------

127. T2.DataT1toT2(D, MM(H+1..4\*H), ME);

128. ----------------------get EH, BH, CH, x ot T2-----------------------

129. accept DataT4toT1 (VCH : in VectorH; VBH : in VectorH; VEH : in VectorH; b : in Integer) do

130. EH := VEH;

131. BH := VBH;

132. CH := VCH;

133. x := b;

134. end DataT4toT1;

135. ----------------------calculation a1 = BH \* CH----------------------

136. a1 := 0;

137. for i in 1..H loop

138. a1 := a1 + BH(i) \* CH(i);

139. end loop;

140. ------------------passing a1 to T2----------------------------------

141. T2.a1toT2(a1);

142. ----------------------------get a ot T2-----------------------------

143. accept ResAtoT1 (a : in Integer) do

144. resa := a;

145. end ResAtoT1;

146. -----------------calculation KH = sort(D\*(ME\*MMH))------------------

147. PreHSort(1, H, D => D, ME => ME, MH => MMH, KH => KH);

148. for i in 1..H loop

149. for j in 1..H+1 loop

150. if KH(i)>KH(j) then

151. e := KH(j);

152. KH(j):=KH(i);

153. KH(i):=e;

154. end if;

155. end loop;

156. end loop;

157. ----------------------passing KH to T2------------------------------

158. T2.KHtoT2(KH);

159. ----------------------get K ot T2-----------------------------------

160. accept KtoT1 (V : in Vector) do

161. resK := K;

162. end KtoT1;

163. ----------------------calculation AH = K + a\*EH\*x-------------------

164. Calculate(a => resa, EH => EH, K => resK, x => x, AH => AH);

165. ----------------------get A3H ot T2---------------------------------

166. accept ResA3HotT2 (ResA3H : in Vector3H) do

167. A(1..H) := AH;

168. A(H+1..N) := ResA3H;

169. end ResA3HotT2;

170. ---------------------print A----------------------------------------

171. if (N<10) then

172. delay(0.5);

173. VectorOutput(A, "A");

174. end if;

175. Put\_Line("T1 finished.");

176. end T1;

177. ---------------------------------T2------------------------------------

178. task body T2 is

179. x, a2, resa, e, i, j, k : Integer;

180. E2H, B2H, C2H, K2H : Vector2H; L, R: VectorAll

181. KH : VectorH;

182. D, resK : Vector;

183. ME : Matrix;

184. MM3H : Matrix3H;

185. begin

186. Put\_Line("T2 started.");

187. ---------------------get D, ME, MM3H ot T1--------------------------

188. accept DataT1toT2 (V : in Vector; M : in Matrix; M3H : in Matrix3H) do

189. D := V;

190. ME := M;

191. MM3H := M3H;

192. end DataT1toT2;

193. ----------------------get E2H, B2H, C2H, x ot T4--------------------

194. accept DataT4toT2 (VC2H : in Vector2H; VB2H : in Vector2H; VE2H : in Vector2H; b : in Integer) do

195. E2H := VE2H;

196. C2H := VC2H;

197. B2H := VB2H;

198. x := b;

199. end DataT4toT2;

200. ----------------------pass EH, BH, CH, x to T1----------------------

201. T1.DataT4toT1(E2H(H+1..2\*H), B2H(H+1..2\*H), C2H(H+1..2\*H), x);

202. ----------------------pass D, ME, MMH ot T1-------------------------

203. T3.DataT1toT3(D, ME, MM3H(H+1..3\*H));

204. ----------------------calculation a2 = BH \* CH----------------------

205. a2 := 0;

206. for i in 1..H loop

207. a2 := a2 + B2H(i) \* C2H(i);

208. end loop;

209. -------------accept a1 ot T1 and calculation a2 = a2 + a1-----------

210. accept a1toT2 (a1 : in Integer) do

211. a2 := a2 + a1;

212. end a1toT2;

213. ------------------------passing a2 to T3----------------------------

214. T3.a2toT3(a2);

215. ----------------------------get a ot T3-----------------------------

216. accept ResAtoT2 (a : in Integer) do

217. resa := a;

218. end ResAtoT2;

219. --------------------------pass a to T1------------------------------

220. T1.ResAtoT1(resa);

221. -------------------calculation KH = sort(D\*(ME\*MMH))----------------

222. PreHSort(H+1, 2\*H, D => D, ME => ME, MMH => MM3H, KH => KH2);

223. for i in 1..len(KH2) loop

224. for j in 1..len(KH2)+1 loop

225. if KH2(i)>KH2(j) then

226. e := KH2(j);

227. KH2(j):=KH2(i);

228. KH2(i):=e;

229. end if;

230. end loop;

231. end loop;

232. --------------------calculation K2H = mergesort(KH, KH)--------------

233. L := (KH);

234. R := (KH2);

235. i := 0;

236. j:=0;

237. k:=0;

238. while i<Len(L) and then j<Len(R) loop

239. if L(i)<R(j) then

240. K2H(k) := L(i);

241. i:=i+ 1;

242. else loop

243. K2H(k) := R(j);

244. j:=j+1;

245. end loop;

246. k:=k+1;

247. end if;

248. end loop;

249. --------------------------passing KH to T3--------------------------

250. T3.KHtoT3(KH2);

251. --------------------------passing K2H to T3-------------------------

252. T3.K2HtoT3(K2H);

253. ----------------------------get K ot T3-----------------------------

254. accept KtoT2 (V : in Vector) do

255. resK := K;

256. end KtoT2;

257. ----------------------------pass K to T1----------------------------

258. T1.KtoT1(resK);

259. ---------------------calculation AH = K + a\*EH\*x--------------------

260. Calculate(a => resa, EH => E2H(1..H), K => resK, x => x, AH => AH);

261. ---------------------------get A2H ot T3----------------------------

262. accept ResA2HotT3 (ResA2H : in Vector2H) do

263. AH(1..H) := AH;

264. A3H(H+1..3\*H) := ResA2H;

265. end ResA2HotT3;

266. --------------------------passing A3H to T1-------------------------

267. T1.ResA3HotT2(A3H);

268. Put\_Line("T2 finished.");

269. end T2;

270. ----------------------T3-------------------------

271. task body T3 is

272. x, a3, resa, e : Integer;

273. D, resK: Vector;

274. ME : Matrix;

275. KH2 : VectorH;

276. A2H, K2H : Vector2H;

277. C3H, B3H, E3H : Vector3H;

278. begin

279. Put\_Line("T3 started.");

280. ---------------------get D, ME, MM3H ot T1--------------------------

281. accept DataT1toT3 (V : in Vector; M : in Matrix; M2H : in Matrix2H) do

282. D := V;

283. ME := M;

284. MM2H := M2H;

285. end DataT1toT3;

286. ---------------------get C, B, E, x ot T4--------------------------

287. accept DataT4toT3 (VC3H : in Vector3H; VB3H : in Vector3H; VE3H : in Vector3H; b : in Integer) do

288. C3H := VC3H;

289. B3H := VB3H;

290. E3H := VE3H;

291. x := b;

292. end DataT4toT3;

293. ----------------------pass E2H, B2H, C2H, x to T2----------------------

294. T2.DataT4toT2(C3H(H+1..3\*H), B3H(H+1..3\*H), E3H(H+1..3\*H), x);

295. ----------------------pass D, ME, MM2H to T4----------------------

296. T4.DataT1toT4(V, M, MM2H(H+1..2\*H));

297. ----------------------calculation a3 = BH \* CH----------------------

298. a3 := 0;

299. for i in 1..H loop

300. a3 := a3 + B2H(i) \* C2H(i);

301. end loop;

302. -----------accept a2 ot T2 and calculation a3 = a3 + a2------------

303. accept a2toT3 (a1 : in Integer) do

304. a3 := a2 + a1;

305. end a2toT3;

306. ------------------------passing a3 to T4----------------------------

307. T4.a3toT4(a3);

308. ----------------------------get a ot T4-----------------------------

309. accept ResAtoT1 (a : in Integer) do

310. resa := a;

311. end ResAtoT1;

312. --------------------------pass a to T2------------------------------

313. T2.ResAtoT2(a);

314. --------------------------get KH2 ot T2-----------------------------

315. accept KHtoT3 (VH : in VectorH) do

316. KH2 := VH;

317. end KHtoT3;

318. --------------------------get K2H ot T2-----------------------------

319. accept K2HtoT3 (VH : in VectorH) do

320. K2H := V2H;

321. end K2HtoT3;

322. -------------------calculation KH = sort(D\*(ME\*MMH))----------------

323. PreHSort(H+1, 3\*H, D => D, ME => ME, MMH => MM2H, KH => KH3);

324. for i in 1..len(KH3) loop

325. for j in 1..len(KH3)+1 loop

326. if KH3(i)>KH3(j) then

327. e := KH3(j);

328. KH3(j):=KH3(i);

329. KH3(i):=e;

330. end if;

331. end loop;

332. end loop;

333. --------------------calculation K2H = sort(KH, KH)------------------

334. L := KH2;

335. R := KH3;

336. i := 0;

337. j:=0;

338. k:=0;

339. while i<Len(L) and then j<Len(R) loop

340. if L(i)<R(j) then

341. K2H2(k) := L(i);

342. i:=i+1;

343. else loop

344. K2H(k) := R(j);

345. j:=j+1;

346. end loop;

347. k:=k+1;

348. end if;

349. end loop;

350. --------------------------passing KH to T4--------------------------

351. T4.KHtoT4(KH);

352. --------------------------passing K2H to T4-------------------------

353. T4.K2HtoT4(K2H);

354. ----------------------------get K ot T4-----------------------------

355. accept KtoT3 (V : in Vector) do

356. resK := K;

357. end KtoT3;

358. ----------------------------pass K to T2----------------------------

359. T2.KtoT2(resK);

360. ---------------------calculation AH = K + a\*EH\*x--------------------

361. Calculate(a => resa, EH => E3H(1..H), K => resK, x => x, AH => AH);

362. ---------------------------get A2H ot T4---------------------------

363. accept ResAHotT4 (ResAH : in VectorH) do

364. A2H(1..H) := AH;

365. A2H(H+1..2\*H) := ResAH;

366. end ResAHotT4;

367. --------------------------passing A2H to T2-------------------------

368. T2.ResA2HotT3(A2H);

369. Put\_Line("T3 finished.");

370. end T3;

371. ------------------------------T4------------------------------

372. task body T4 is

373. x, a, a4, e, i, j, k : Integer;

374. E, B, C, D, K : Vector;

375. R, L : Vector\_All;

376. ME : Matrix;

377. AH, KH3, KH4 : VectorH;

378. MMH : MatrixH;

379. K2H2, K2H3 : Vector2H;

380. begin

381. Put\_Line("T4 started.");

382. --------------------------Input E, B, C, x--------------------------

383. VectorInput(E);

384. MatrixInput(B);

385. MatrixInput(C);

386. InputNumber(x);

387. ----------------------passing E, B, C, x to T3----------------------

388. T3.DataT4toT3(E(H+1..N), B(H+1..N), C(H+1..N), x);

389. ----------------------get D, ME, MMH ot T3--------------------------

390. accept DataT1toT4 (V : in Vector; M : in Matrix; MH : in MatrixH) do

391. D := V;

392. ME := M;

393. MMH := MH;

394. end DataT1toT4;

395. ----------------------calculation a4 = BH \* CH-----------------------

396. a4 := 0;

397. for i in 1..H loop

398. a4 := a4 + BH(i) \* CH(i);

399. end loop;

400. -----------accept a2 ot T2 and calculation a3 = a3 + a2-------------

401. accept a3toT4 (a1 : in Integer) do

402. a4 := a4 + a3;

403. end a3toT4;

404. ------------------------------a=a4+a3-------------------------------

405. a := a4;

406. ------------------------passing a to T3-----------------------------

407. T3.ResAtoT3(a);

408. ------------------------get KH ot T3--------------------------------

409. accept KHtoT4 (VH : in VectorH) do

410. KH3 := VH;

411. end KHtoT4;

412. ------------------------get K2H ot T3--------------------------------

413. accept K2HtoT4 (V2H : in Vector2H) do

414. K2H2 := V2H;

415. end K2HtoT4;

416. -------------------calculation KH = sort(D\*(ME\*MMH))----------------

417. PreHSort(H+1, N, D => D, ME => ME, MMH => MMH, KH => KH4);

418. for i in 1..len(KH4) loop

419. for j in 1..len(KH4)+1 loop

420. if KH4(i)>KH4(j) then

421. e := KH4(j);

422. KH4(j):=KH4(i);

423. KH4(i):=e;

424. end if;

425. end loop;

426. end loop;

427. ----------------calculation K2H = mergesort(KH, KH)-----------------

428. L := KH3;

429. R := KH4;

430. i := 0;

431. j:=0;

432. k:=0;

433. while i<Len(L) and then j<Len(R) loop

434. if L(i)<R(j) then

435. K2H3(k) := L(i);

436. i:=i+1;

437. else loop

438. K2H3(k) := R(j);

439. j:=j+1;

440. end loop;

441. k:=k+1;

442. end if;

443. end loop;

444. ----------------calculation K = mergesort(K2H, K2H)-----------------

445. L := K2H2;

446. R := (K2H3);

447. i := 0;

448. j:=0;

449. k:=0;

450. while i<Len(L) and then j<Len(R) loop

451. if L(i)<R(j) then

452. K(k) := L(i);

453. i:=i+1;

454. else loop

455. K(k) := R(j);

456. j:=j+1;

457. end loop;

458. k:=k+1;

459. end if;

460. end loop;

461. ----------------------------pass K to T3----------------------------

462. T3.KtoT3(K);

463. ---------------------calculation AH = K + a\*EH\*x--------------------

464. Calculate(a => a4, EH => E(1..H), K => K, x => x, AH => AH);

465. --------------------------passing AH to T3-------------------------

466. T3.ResAHotT4(AH);

467. Put\_Line("T4 finished.");

468. end T4;

469. begin

470. Put\_Line("");

471. end tasksProcedure;

472. begin

473. Put\_Line("Lab4 started.");

474. New\_Line;

475. tasksProcedure;

476. Put\_Line("Lab4 finished.");

477. New\_Line;

478. end Lab4;