**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

Дисциплина: «Архитектура вычислительных систем»

**Программа, решающая вопрос о нахождении пар параллельных отрезков**Пояснительная записка

**Выполнил:**  
Гурова Екатерина Александровна,  
*студент гр. БПИ198, 1 подгруппа*

**Москва**2020

Содержание

[1. Текст задания 2](#_Toc55066584)

[2. Применяемые расчетные методы 3](#_Toc55066585)

[3. Описание данных 4](#_Toc55066586)

[3.1. Описание входных данных 4](#_Toc55066587)

[3.2. Описание выходных данных 4](#_Toc55066588)

[3.3. Описание переменных с промежуточными и выходными данными 4](#_Toc55066589)

[4. Тестирование программы 6](#_Toc55066590)

[4.1. Корректные значения 6](#_Toc55066591)

[4.2. Некорректные значения 9](#_Toc55066592)

[ПРИЛОЖЕНИЕ 1 10](#_Toc55066597)

[Список литературы 10](#_Toc55066598)

[ПРИЛОЖЕНИЕ 2 11](#_Toc55066599)

[Код программы с комментариями 11](#_Toc55066600)

1. Текст задания

Разработать программу, которая решает вопрос о нахождении пар параллельных отрезков из общего числа N=4 отрезков, заданных координатами концевых точек.

1. Применяемые расчетные методы

Для определения параллельности отрезков вычисляются длины их проекций на оси X и Y. Для этого из координат правой точки соответственно вычитаются координаты левой точки, после чего первое число равно длине проекции отрезка на ось X (dx), а второе число равно длине проекции отрезка на ось Y (dy).

Отрезки являются параллельными, если отношения их длин проекций на оси равны, то есть

= (1)

Это равенство эквивалентно следующему:

(2)

Данные два отрезка параллельны тогда и только тогда, когда равенство (2) соблюдается.

1. Описание данных
   1. Описание входных данных

Входными данными являются шестнадцать целых чисел на отрезке [-100; 100], которые являются координатами концевых точек четырех отрезков.

Каждые четыре числа относятся к одному из четырех отрезку. Отрезки номеруются от 1 до 4. Координаты следует вводить в следующем порядке: {x1 y1 x2 y2}, где xi – абсцисса i-той точки отрезка, yi – ордината j-той точки отрезка.

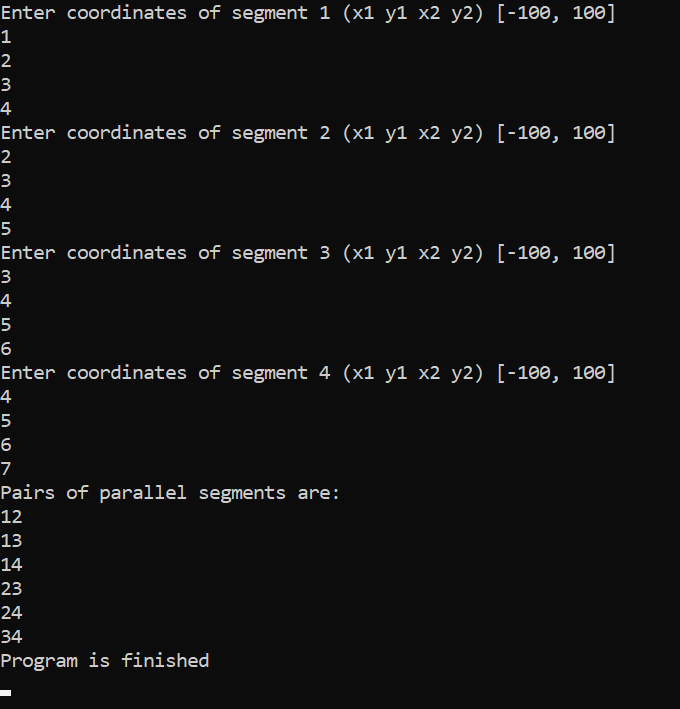
* 1. Описание выходных данных

Выходные данные представляют собой пары номеров параллельных, записанных на каждой линии отдельно.

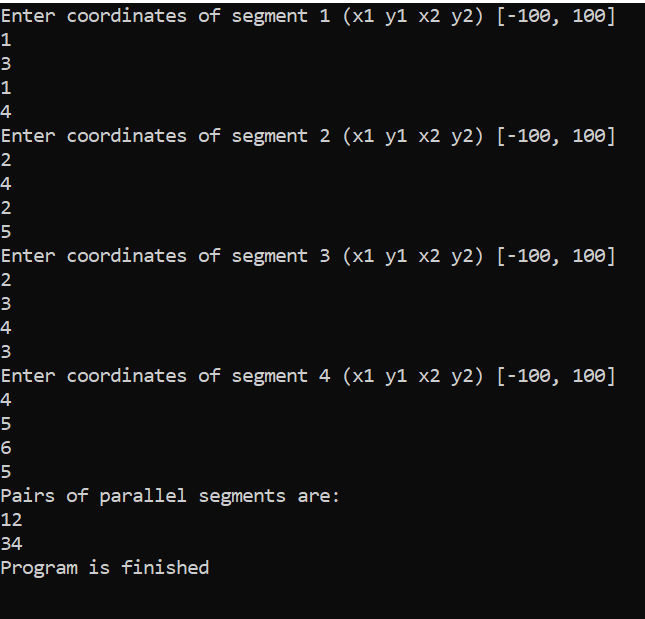
* 1. Описание переменных с промежуточными и выходными данными

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип | Значение | Описание |
| enterStr1 | Db | 'Enter coordinates of segment 1 (x1 y1 x2 y2) [-100, 100]', 10, 13, 0 | Фраза для ввода данных отрезка 1 |
| enterStr2 | Db | 'Enter coordinates of segment 2 (x1 y1 x2 y2) [-100, 100]', 10, 13, 0 | Фраза для ввода данных отрезка 2 |
| enterStr3 | Db | 'Enter coordinates of segment 3 (x1 y1 x2 y2) [-100, 100]', 10, 13, 0 | Фраза для ввода данных отрезка 3 |
| enterStr4 | Db | 'Enter coordinates of segment 4 (x1 y1 x2 y2) [-100, 100]', 10, 13, 0 | Фраза для ввода данных отрезка 4 |
| wrongStr | Db | ‘Wrong input’, 10, 13 ,0 | Фраза для данных, выходящих из диапазона допустимых значений |
| D | Db | ‘%d’, 10, 13, 0 | Переменная для приведения числа к строке |
| newLine | Db | ‘’, 10, 13, 0 | Пустая строка |
| resultStr | Db | 'Pairs of parallel segments are:', 10, 13, 0 | Вывод результата |
| finishStr | Db | 'Program is finished', 10, 13, 0 | Конец программы |
| M1 | Dd | 4 | Следющий номер индекса за номером индексом ячейки, в котором хранится последняя координата отрезка 1 |
| M2 | Dd | 8 | Следющий номер индекса за номером индексом ячейки, в котором хранится последняя координата отрезка 1 |
| M3 | Dd | 12 | Следющий номер индекса за номером индексом ячейки, в котором хранится последняя координата отрезка 1 |
| N | Dd | 16 | Размер массива А |
| minVal | Dd | -100 | Минимальное значение диапазона допустимых чисел |
| maxVal | Dd | 100 | Максимальное значение диапазона допустимых чисел |
| Cur | Dd | ? | Текущий читаемый элемент |
| heapA | Dd | ? | Куча массива А |
| heapB | dd | ? | Куча массива В |
| A | dd | ? | Первый элемент массива А |
| B | dd | ? | Первый элемент массива В |
| Anc | dd | ? | Начальное значение регистра esp |
| I | Dd | ? | Первый счетчик |
| J | Dd | ? | Второй счетчик |
| X1 | Dd | ? | Абсцисса первой точки |
| X2 | Dd | ? | Абсцисса второй точки |
| Y1 | Dd | ? | Ордината первой точки |
| Y2 | Dd | ? | Ордината второй точки |
| Dx1 | Dd | ? | Длина проекции первого отрезка на Х |
| Dx2 | Dd | ? | Длина проекции второго отрезка на Х |
| Dy1 | Dd | ? | Длина проекции первого отрезка на Y |
| Dy2 | Dd | ? | Длина проекции второго отрезка на Y |
| Left | dd | ? | Значение произведения dx1\*dy2 |

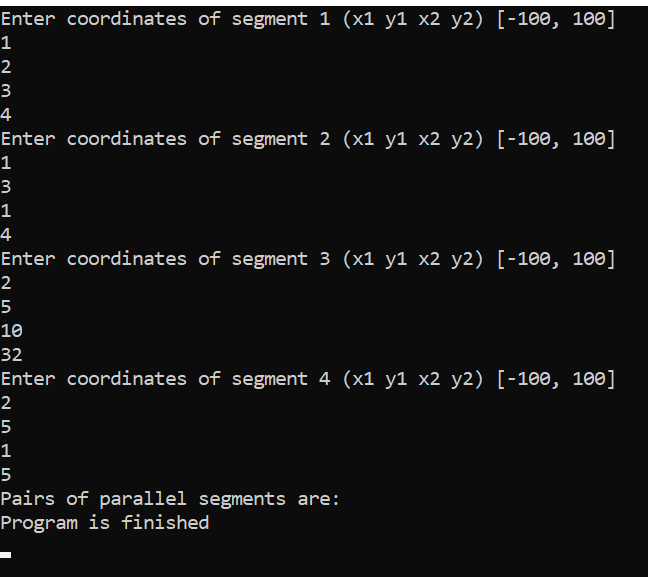
1. Тестирование программы
   1. Корректные значения



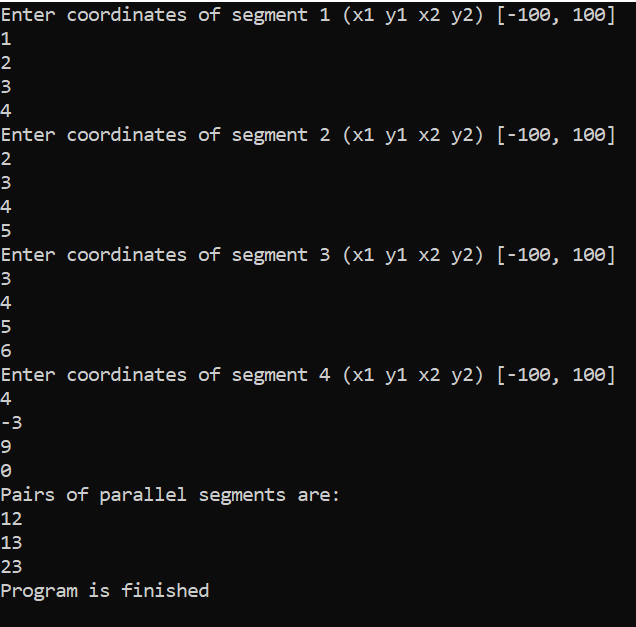
*Рисунок 1. Все отрезки параллельны.*

**

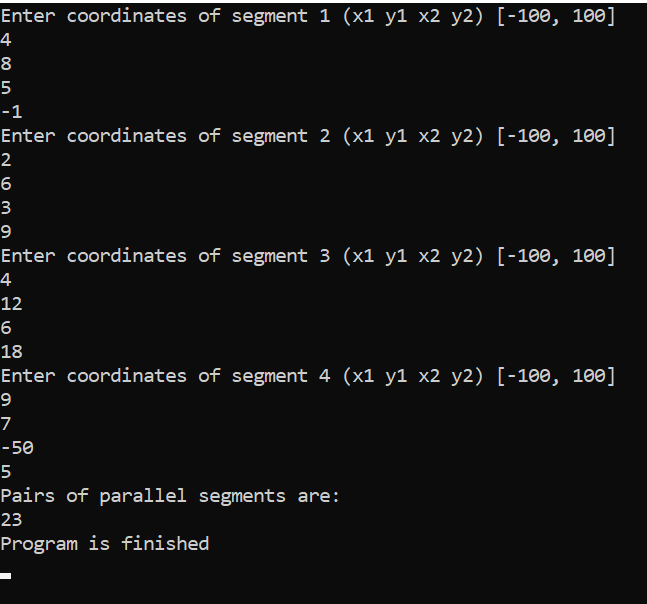
*Рисунок 2. Первые два отрезка параллельны оси X, а последние два отрезка параллельны оси Y.*

**

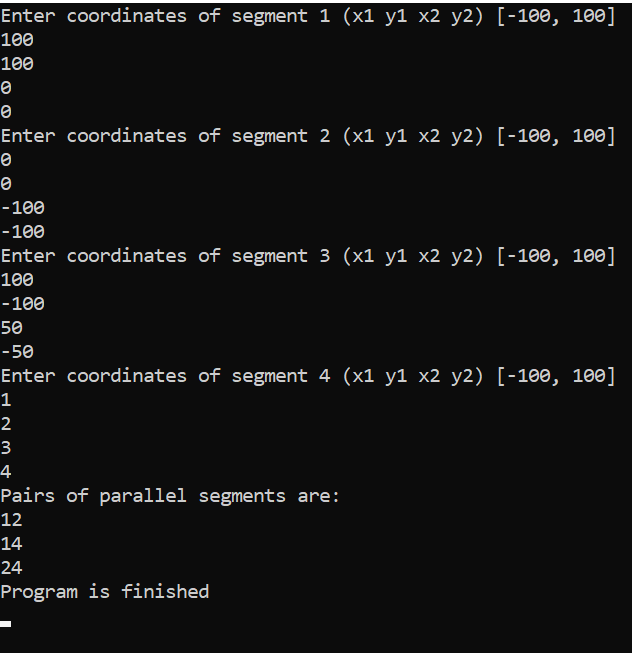
*Рисунок 3. Параллельных отрезков нет.*

**

*Рисунок 4. Три из четырех отрезков параллельны.*

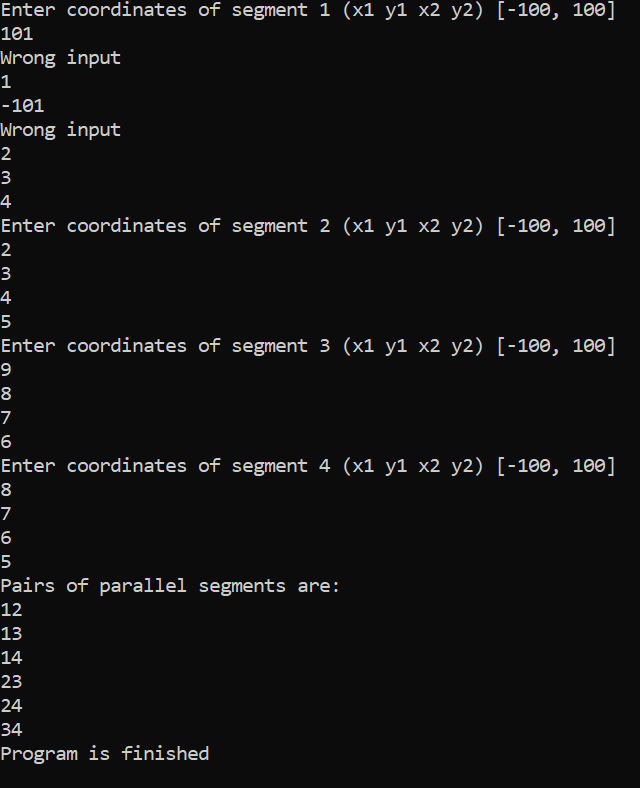
**

*Рисунок 5. Два из четырех отрезков параллельны.*

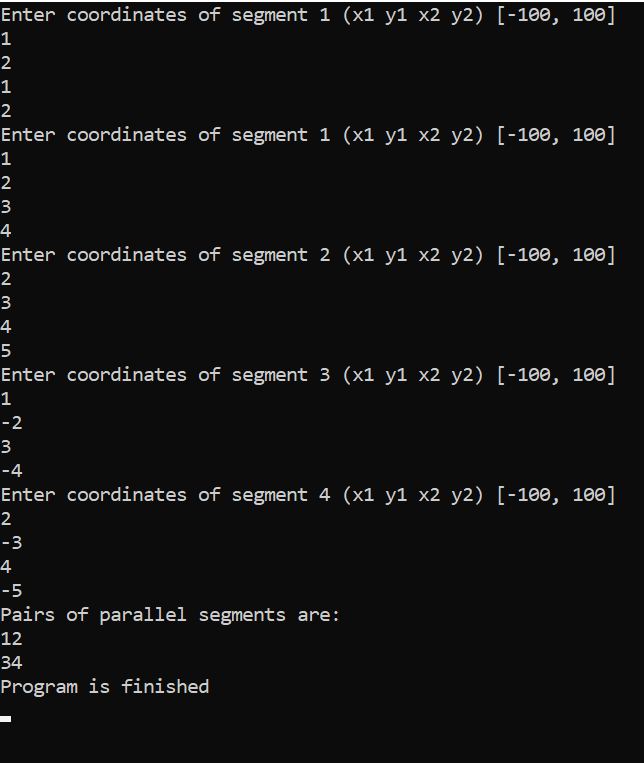
**

*Рисунок 6. Ввод крайних значений.*

* 1. Некорректные значения



*Рисунок 7. Значения входных данных не входят в допустимый интервал.*

**

*Рисунок 8. Ввод координат точки.*

ПРИЛОЖЕНИЕ 1

Список литературы

1. **Пересечение: Прямая (отрезок) и прямая (отрезок)**. [Электронный ресурс] // URL: http://algolist.ru/maths/geom/intersect/lineline2d.php (дата обращения: 31.10.2020)

ПРИЛОЖЕНИЕ 2

Код программы с комментариями

1. ;Программа, находящая пары параллельных отрезков из общего количества отрезков N=4.
2. format PE console
3. entry Start
4. include 'win32a.inc'
5. section '.data' data readable writable
6. enterStr1 db 'Enter coordinates of segment 1 (x1 y1 x2 y2) [-100, 100]', 10, 13, 0
7. enterStr2 db 'Enter coordinates of segment 2 (x1 y1 x2 y2) [-100, 100]', 10, 13, 0
8. enterStr3 db 'Enter coordinates of segment 3 (x1 y1 x2 y2) [-100, 100]', 10, 13, 0
9. enterStr4 db 'Enter coordinates of segment 4 (x1 y1 x2 y2) [-100, 100]', 10, 13, 0
10. wrongStr db 'Wrong input', 10, 13, 0
11. D db '%d', 0
12. newLine db '', 10, 13, 0
13. resultStr db 'Pairs of parallel segments are:', 10, 13, 0
14. finishStr db 'Program is finished', 10, 13, 0
15. heapA dd ?
16. anc dd ?
17. heapB dd ?
18. A dd ?
19. B dd ?
20. i dd ?
21. j dd ?
22. M1 dd 4
23. M2 dd 8
24. M3 dd 12
25. N dd 16
26. cur dd ?
27. minVal dd -100
28. maxVal dd 100
29. x1 dd ?
30. x2 dd ?
31. y1 dd ?
32. y2 dd ?
33. dx1 dd ?
34. dx2 dd ?
35. dy1 dd ?
36. dy2 dd ?
37. left dd ?
38. section '.code' code readable executable
39. Start:
40. call CreateA
41. call ReadCoord1
42. call ReadCoord2
43. call ReadCoord3
44. call ReadCoord4
45. call CreateB
46. call FormB
47. call IsParallel
48. cinvoke printf, finishStr
49. invoke HeapFree,[heapA],HEAP\_NO\_SERIALIZE, [A]
50. invoke HeapFree,[heapB],HEAP\_NO\_SERIALIZE, [B]
51. invoke getch
52. invoke ExitProcess
53. ;Перебирает всевозможные пары отрезков, рассчитывает, являются ли они параллельными
54. IsParallel:
55. cinvoke printf, resultStr
56. mov [anc], esp
57. mov [i], -2
58. IsParallelLoop1:
59. add [i], 2
60. mov ebx, [i]
61. mov [j], ebx
62. IsParallelLoop2:
63. add [j], 2
64. mov eax, [B]
65. ; Чтение значений проекций для данной пары отрезков из массива В
66. mov ecx, [i]
67. mov ebx, [eax + ecx\*4]
68. mov [dx1], ebx
69. add ecx, 1
70. mov ebx, [eax + ecx\*4]
71. mov [dy1], ebx
72. mov ecx, [j]
73. mov ebx, [eax + ecx\*4]
74. mov [dx2], ebx
75. add ecx, 1
76. mov ebx, [eax + ecx\*4]
77. mov [dy2], ebx
78. mov ebx, [dx1]
79. imul ebx, dword [dy2]
80. mov [left], ebx
81. mov ebx, [dy2]
82. mov ebx, [dx2]
83. imul ebx, dword [dy1]
84. cmp ebx, [left]
85. je PrintIsParallel
86. cmp [i], 4
87. je Finish
88. cmp [j], 6
89. je IsParallelLoop1
90. cmp [j], 6
91. jl IsParallelLoop2
92. mov esp, [anc]
93. ret
94. ;Печатает номера пары отрезков, если они параллельные
95. PrintIsParallel:
96. ; Номера отрезков определяются по значениям счетчиков (i / 2 + 1)
97. mov eax, [i]
98. mov ecx, 2
99. mov edx, 0
100. div ecx
101. add eax, 1
102. cinvoke printf, D, eax
103. mov eax, [j]
104. mov ecx, 2
105. mov edx, 0
106. div ecx
107. add eax, 1
108. cinvoke printf, D, eax
109. cinvoke printf, newLine
110. cmp [i], 4
111. je Finish
112. cmp [j], 6
113. je IsParallelLoop1
114. cmp [j], 6
115. jl IsParallelLoop2
116. ; Окончание цикла
117. Finish:
118. ret
119. ; Формирует массив значений проекций отрезков на оси X и Y
120. FormB:
121. mov [anc], esp
122. mov [i], -1
123. mov [j], 0
124. FormLoop1:
125. ; Чтение координат концевых точек данного отрезка
126. mov eax, [A]
127. add [i], 1
128. mov ecx, [i]
129. mov ebx, [eax + ecx\*4]
130. mov [x1], ebx
131. mov eax, [A]
132. add [i], 1
133. mov ecx, [i]
134. mov ebx, [eax + ecx\*4]
135. mov [y1], ebx
136. mov eax, [A]
137. add [i], 1
138. mov ecx, [i]
139. mov ebx, [eax + ecx\*4]
140. mov [x2], ebx
141. mov eax, [A]
142. add [i], 1
143. mov ecx, [i]
144. mov ebx, [eax + ecx\*4]
145. mov [y2], ebx
146. ; Определние, какая из точек является правой или левой (по абсциссе)
147. mov ebx, [x1]
148. ;Если x1 меньше x2, значит вторая точка является правой
149. cmp ebx, [x2]
150. jl ProjectionX2
151. ;Если x1 равно x2, значит отрезок параллелен оси Х
152. cmp ebx, [x2]
153. je ProjectionX2
154. ;Если x1 больше x2, значит первая точка является правой
155. cmp ebx, [x2]
156. jg ProjectionX1
157. mov esp, [anc]
158. ret
159. ; Добавляет значения проекций в массив В, если первая точка является правой
160. ProjectionX1:
161. mov ebx, [x1]
162. sub ebx, dword [x2]
163. mov eax, [B]
164. mov ecx, [j]
165. mov [eax + 4\*ecx], ebx
166. add [j], 1
167. mov ebx, [y1]
168. sub ebx, dword [y2]
169. mov eax, [B]
170. mov ecx, [j]
171. mov [eax + 4\*ecx], ebx
172. add [j], 1
173. cmp [i], 16
174. jl FormLoop1
175. ret
176. ; Добавляет значения проекций в массив В, если вторая точка является правой или точки имеют одинаковое значение абсциссы
177. ProjectionX2:
178. mov ebx, [x1]
179. sub dword [x2], ebx
180. mov eax, [B]
181. mov ecx, [j]
182. imul ecx, dword 4
183. add eax, ecx
184. mov ebx, [x2]
185. mov [eax], ebx
186. add [j], 1
187. mov ebx, [y1]
188. sub dword [y2], ebx
189. mov eax, [B]
190. mov ecx, [j]
191. imul ecx, dword 4
192. add eax, ecx
193. mov ebx, [y2]
194. mov [eax], ebx
195. add [j], 1
196. cmp [i], 16
197. jl FormLoop1
198. ret
199. ; Выделение памяти для массива А, в котором хранятся координаты концевых точек
200. CreateA:
201. mov [anc], esp
202. invoke HeapCreate,HEAP\_NO\_SERIALIZE,0,0
203. mov [heapA],eax
204. invoke HeapAlloc,[heapA],HEAP\_NO\_SERIALIZE,N\*8
205. mov [A],eax
206. mov esp, [anc]
207. ret
208. ; Выделение памяти для массива В, в котором хранятся значения проекций
209. CreateB:
210. mov [anc], esp
211. invoke HeapCreate,HEAP\_NO\_SERIALIZE,0,0
212. mov [heapB], eax
213. invoke HeapAlloc,[heapB],HEAP\_NO\_SERIALIZE,M2\*8
214. mov [B],eax
215. mov esp, [anc]
216. ret
217. ; Чтение координат концевых точек 1 отрезка
218. ReadCoord1:
219. mov esp, [anc]
220. mov [anc], esp
221. invoke printf, enterStr1
222. mov [i], 0
223. ReadLoop1:
224. ;Чтение значения и проверка на границы
225. cinvoke scanf, D, cur
226. cmp [cur], -100
227. jl WrongInput1
228. cmp [cur], 100
229. jg WrongInput1
230. mov eax, [A]
231. mov ecx, [i]
232. mov ebx, [cur]
233. mov [eax + 4\*ecx], ebx
234. add [i], 1
235. mov ecx, [i]
236. cmp ecx, [M1]
237. jne ReadLoop1
238. ;Проверка, образуют ли введенные координаты точку
239. mov eax, [A]
240. mov ebx, [eax]
241. mov [x1], ebx
242. mov ebx, [eax + 4]
243. mov [y1], ebx
244. mov eax, [A]
245. mov ebx, [eax + 8]
246. mov [x2], ebx
247. mov eax, [A]
248. mov ebx, [eax + 12]
249. mov [y2], ebx
250. mov ebx, [x1]
251. cmp ebx, [x2]
252. je IsItPoint1
253. mov esp, [anc]
254. ret
255. ; Чтение координат концевых точек 2 отрезка
256. ReadCoord2:
257. mov esp, [anc]
258. mov [anc], esp
259. invoke printf, enterStr2
260. mov [i], 4
261. ReadLoop2:
262. ;Чтение значения и проверка на границы
263. cinvoke scanf, D, cur
264. cmp [cur], -100
265. jl WrongInput2
266. cmp [cur], 100
267. jg WrongInput2
268. mov eax, [A]
269. mov ecx, [i]
270. mov ebx, [cur]
271. mov [eax + 4\*ecx], ebx
272. add [i], 1
273. mov ecx, [i]
274. cmp ecx, [M2]
275. jne ReadLoop2
276. ;Проверка, образуют ли введенные координаты точку
277. mov eax, [A]
278. mov ebx, [eax + 16]
279. mov [x1], ebx
280. mov ebx, [eax + 20]
281. mov [y1], ebx
282. mov eax, [A]
283. mov ebx, [eax + 24]
284. mov [x2], ebx
285. mov eax, [A]
286. mov ebx, [eax + 28]
287. mov [y2], ebx
288. mov ebx, [x1]
289. cmp ebx, [x2]
290. je IsItPoint2
291. mov esp, [anc]
292. ret
293. ; Чтение координат концевых точек 3 отрезка
294. ReadCoord3:
295. mov esp, [anc]
296. mov [anc], esp
297. invoke printf, enterStr3
298. mov [i], 8
299. ReadLoop3:
300. ;Чтение значения и проверка на границы
301. cinvoke scanf, D, cur
302. cmp [cur], -100
303. jl WrongInput3
304. cmp [cur], 100
305. jg WrongInput3
306. mov eax, [A]
307. mov ecx, [i]
308. mov ebx, [cur]
309. mov [eax + 4\*ecx], ebx
310. add [i], 1
311. mov ecx, [i]
312. cmp ecx, [M3]
313. jne ReadLoop3
314. ;Проверка, образуют ли введенные координаты точку
315. mov eax, [A]
316. mov ebx, [eax + 32]
317. mov [x1], ebx
318. mov ebx, [eax + 36]
319. mov [y1], ebx
320. mov eax, [A]
321. mov ebx, [eax + 40]
322. mov [x2], ebx
323. mov eax, [A]
324. mov ebx, [eax + 44]
325. mov [y2], ebx
326. mov ebx, [x1]
327. cmp ebx, [x2]
328. je IsItPoint3
329. mov esp, [anc]
330. ret
331. ; Чтение координат концевых точек 4 отрезка
332. ReadCoord4:
333. mov esp, [anc]
334. mov [anc], esp
335. invoke printf, enterStr4
336. mov [i], 12
337. ReadLoop4:
338. ;Чтение значения и проверка на границы
339. cinvoke scanf, D, cur
340. cmp [cur], -100
341. jl WrongInput4
342. cmp [cur], 100
343. jg WrongInput4
344. mov eax, [A]
345. mov ecx, [i]
346. mov ebx, [cur]
347. mov [eax + 4\*ecx], ebx
348. add [i], 1
349. mov ecx, [i]
350. cmp ecx, [N]
351. jne ReadLoop4
352. ;Проверка, образуют ли введенные координаты точку
353. mov eax, [A]
354. mov ebx, [eax + 48]
355. mov [x1], ebx
356. mov ebx, [eax + 52]
357. mov [y1], ebx
358. mov eax, [A]
359. mov ebx, [eax + 56]
360. mov [x2], ebx
361. mov eax, [A]
362. mov ebx, [eax + 60]
363. mov [y2], ebx
364. mov ebx, [x1]
365. cmp ebx, [x2]
366. je IsItPoint4
367. mov esp, [anc]
368. ret
369. ; Неверные данные при чтении данных отрезка 1
370. WrongInput1:
371. invoke printf, wrongStr
372. jmp ReadLoop1
373. ; Неверные данные при чтении данных отрезка 2
374. WrongInput2:
375. invoke printf, wrongStr
376. jmp ReadLoop2
377. ; Неверные данные при чтении данных отрезка 3
378. WrongInput3:
379. invoke printf, wrongStr
380. jmp ReadLoop3
381. ; Неверные данные при чтении данных отрезка 4
382. WrongInput4:
383. invoke printf, wrongStr
384. jmp ReadLoop4
385. ; Проверка, образуют ли введенные координаты для 1 отрезка точку
386. IsItPoint1:
387. mov ebx, [y1]
388. cmp ebx, [y2]
389. je ReadCoord1
390. mov esp, [anc]
391. ret
392. ; Проверка, образуют ли введенные координаты для 2 отрезка точку
393. IsItPoint2:
394. mov ebx, [y1]
395. cmp ebx, [y2]
396. je ReadCoord2
397. mov esp, [anc]
398. ret
399. ; Проверка, образуют ли введенные координаты для 3 отрезка точку
400. IsItPoint3:
401. mov ebx, [y1]
402. cmp ebx, [y2]
403. je ReadCoord3
404. mov esp, [anc]
405. ret
406. ; Проверка, образуют ли введенные координаты для 4 отрезка точку
407. IsItPoint4:
408. mov ebx, [y1]
409. cmp ebx, [y2]
410. je ReadCoord4
411. mov esp, [anc]
412. ret
413. section 'idata' import data readable
414. library kernel, 'kernel32.dll',\
415. msvcrt, 'msvcrt.dll',\
416. user32,'USER32.DLL'
417. include 'api\user32.inc'
418. include 'api\kernel32.inc'
419. import kernel,\
420. ExitProcess, 'ExitProcess',\
421. HeapCreate, 'HeapCreate',\
422. HeapFree, 'HeapFree',\
423. HeapAlloc, 'HeapAlloc'
424. include 'api\kernel32.inc'
425. import msvcrt,\
426. printf, 'printf',\
427. scanf, 'scanf',\
428. getch, '\_getch'