

**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
2. Применяемые расчетные методы.....	3
3. Описание данных.....	4
3.1. Описание входных данных.....	4
3.2. Описание выходных данных	4
3.3. Описание переменных с промежуточными и выходными данными.....	4
4. Тестирование программы.....	6
4.1. Корректные значения	6
4.2. Некорректные значения	9
ПРИЛОЖЕНИЕ 1.....	10
Список литературы	10
ПРИЛОЖЕНИЕ 2.....	11
Код программы с комментариями	11

1. Текст задания

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

2. 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

$$\frac{dx_1}{dy_1} = \frac{dx_2}{dy_2} \quad (1)$$

Это равенство эквивалентно следующему:

$$dx_1 dy_2 = dx_2 dy_1 \quad (2)$$

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

3. Описание данных

3.1. 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: $\{x_1 y_1 x_2 y_2\}$, где x_i – абсцисса i -той точки отрезка, y_i – ордината j -той точки отрезка.

3.2. Описание выходных данных

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

3.3. 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	Следующий номер индекса за номером индексом ячейки, в котором хранится последняя координата отрезка 1
M2	Dd	8	Следующий номер индекса за номером индексом ячейки, в котором хранится последняя координата отрезка 1

M3	Dd	12	Следующий номер индекса за номером индексом ячейки, в котором хранится последняя координата отрезка 1
N	Dd	16	Размер массива A
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 A
heapB	dd	?	Куча массива B
A	dd	?	Первый элемент массива A
B	dd	?	Первый элемент массива B
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 X
Dx2	Dd	?	Длина проекции второго отрезка на X
Dy1	Dd	?	Длина проекции первого отрезка на Y
Dy2	Dd	?	Длина проекции второго отрезка на Y
Left	dd	?	Значение произведения $dx1*dy2$

4. Тестирование программы

4.1. Корректные значения

```

Enter coordinates of segment 1 (x1 y1 x2 y2) [-100, 100]
1
2
3
4
Enter coordinates of segment 2 (x1 y1 x2 y2) [-100, 100]
2
3
4
5
Enter coordinates of segment 3 (x1 y1 x2 y2) [-100, 100]
3
4
5
6
Enter coordinates of segment 4 (x1 y1 x2 y2) [-100, 100]
4
5
6
7
Pairs of parallel segments are:
12
13
14
23
24
34
Program is finished

```

Рисунок 1. Все отрезки параллельны.

```

Enter coordinates of segment 1 (x1 y1 x2 y2) [-100, 100]
1
3
1
4
Enter coordinates of segment 2 (x1 y1 x2 y2) [-100, 100]
2
4
2
5
Enter coordinates of segment 3 (x1 y1 x2 y2) [-100, 100]
2
3
4
3
Enter coordinates of segment 4 (x1 y1 x2 y2) [-100, 100]
4
5
6
5
Pairs of parallel segments are:
12
34
Program is finished

```

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

```

Enter coordinates of segment 1 (x1 y1 x2 y2) [-100, 100]
1
2
3
4
Enter coordinates of segment 2 (x1 y1 x2 y2) [-100, 100]
1
3
1
4
Enter coordinates of segment 3 (x1 y1 x2 y2) [-100, 100]
2
5
10
32
Enter coordinates of segment 4 (x1 y1 x2 y2) [-100, 100]
2
5
1
5
Pairs of parallel segments are:
Program is finished

```

Рисунок 3. Параллельных отрезков нет.

```

Enter coordinates of segment 1 (x1 y1 x2 y2) [-100, 100]
1
2
3
4
Enter coordinates of segment 2 (x1 y1 x2 y2) [-100, 100]
2
3
4
5
Enter coordinates of segment 3 (x1 y1 x2 y2) [-100, 100]
3
4
5
6
Enter coordinates of segment 4 (x1 y1 x2 y2) [-100, 100]
4
-3
9
0
Pairs of parallel segments are:
12
13
23
Program is finished

```

Рисунок 4. Три из четырех отрезков параллельны.


```

Enter coordinates of segment 1 (x1 y1 x2 y2) [-100, 100]
4
8
5
-1
Enter coordinates of segment 2 (x1 y1 x2 y2) [-100, 100]
2
6
3
9
Enter coordinates of segment 3 (x1 y1 x2 y2) [-100, 100]
4
12
6
18
Enter coordinates of segment 4 (x1 y1 x2 y2) [-100, 100]
9
7
-50
5
Pairs of parallel segments are:
23
Program is finished

```

Рисунок 5. Два из четырех отрезков параллельны.

```

Enter coordinates of segment 1 (x1 y1 x2 y2) [-100, 100]
100
100
0
0
Enter coordinates of segment 2 (x1 y1 x2 y2) [-100, 100]
0
0
-100
-100
Enter coordinates of segment 3 (x1 y1 x2 y2) [-100, 100]
100
-100
50
-50
Enter coordinates of segment 4 (x1 y1 x2 y2) [-100, 100]
1
2
3
4
Pairs of parallel segments are:
12
14
24
Program is finished

```

Рисунок 6. Ввод крайних значений.

4.2. Некорректные значения

```

Enter coordinates of segment 1 (x1 y1 x2 y2) [-100, 100]
101
Wrong input
1
-101
Wrong input
2
3
4
Enter coordinates of segment 2 (x1 y1 x2 y2) [-100, 100]
2
3
4
5
Enter coordinates of segment 3 (x1 y1 x2 y2) [-100, 100]
9
8
7
6
Enter coordinates of segment 4 (x1 y1 x2 y2) [-100, 100]
8
7
6
5
Pairs of parallel segments are:
12
13
14
23
24
34
Program is finished

```

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

```

Enter coordinates of segment 1 (x1 y1 x2 y2) [-100, 100]
1
2
1
2
Enter coordinates of segment 1 (x1 y1 x2 y2) [-100, 100]
1
2
3
4
Enter coordinates of segment 2 (x1 y1 x2 y2) [-100, 100]
2
3
4
5
Enter coordinates of segment 3 (x1 y1 x2 y2) [-100, 100]
1
-2
3
-4
Enter coordinates of segment 4 (x1 y1 x2 y2) [-100, 100]
2
-3
4
-5
Pairs of parallel segments are:
12
34
Program is finished

```

Рисунок 8. Ввод координат точки.

Список литературы

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

Код программы с комментариями

```

1. ;Программа, находящая пары параллельных отрезков из общего
   количества отрезков N=4.
2.
3. format PE console
4.
5. entry Start
6.
7. include 'win32a.inc'
8.
9. section '.data' data readable writable
10.             enterStr1      db          'Enter coordinates of segment 1
   (x1 y1 x2 y2) [-100, 100]', 10, 13, 0
11.             enterStr2      db          'Enter coordinates of segment 2
   (x1 y1 x2 y2) [-100, 100]', 10, 13, 0
12.             enterStr3      db          'Enter coordinates of segment 3
   (x1 y1 x2 y2) [-100, 100]', 10, 13, 0
13.             enterStr4      db          'Enter coordinates of segment 4
   (x1 y1 x2 y2) [-100, 100]', 10, 13, 0
14.             wrongStr       db          'Wrong input', 10, 13, 0
15.             D               db          '%d', 0
16.             newLine         db          '', 10, 13, 0
17.             resultStr       db          'Pairs of parallel segments
   are:', 10, 13, 0
18.             finishStr       db          'Program is finished', 10, 13,
   0
19.             heapA           dd          ?
20.             anc             dd          ?
21.             heapB           dd          ?
22.             A               dd          ?
23.             B               dd          ?

```

24.	i	dd	?
25.	j	dd	?
26.	M1	dd	4
27.	M2	dd	8
28.	M3	dd	12
29.	N	dd	16
30.	cur	dd	?
31.	minVal	dd	-100
32.	maxVal	dd	100
33.	x1	dd	?
34.	x2	dd	?
35.	y1	dd	?
36.	y2	dd	?
37.	dx1	dd	?
38.	dx2	dd	?
39.	dy1	dd	?
40.	dy2	dd	?
41.	left	dd	?
42.			
43.	section '.code' code readable executable		
44.	Start:		
45.	call CreateA		
46.	call ReadCoord1		
47.	call ReadCoord2		
48.	call ReadCoord3		
49.	call ReadCoord4		
50.	call CreateB		
51.	call FormB		
52.	call IsParallel		
53.			
54.	cinvoke printf, finishStr		

```

55.
56.             invoke HeapFree,[heapA],HEAP_NO_SERIALIZE, [A]
57.             invoke HeapFree,[heapB],HEAP_NO_SERIALIZE, [B]
58.
59.             invoke getch
60.             invoke ExitProcess
61.
62.             ;Перебирает всевозможные пары отрезков, рассчитывает, являются
               ли они параллельными
63.             IsParallel:
64.                 cinvoke printf, resultStr
65.                 mov [anc], esp
66.                 mov [i], -2
67.                 IsParallelLoop1:
68.                     add [i], 2
69.                     mov ebx, [i]
70.                     mov [j], ebx
71.                 IsParallelLoop2:
72.                     add [j], 2
73.                     mov eax, [B]
74.
75.                 ; Чтение значений проекций для данной
               пары отрезков из массива B
76.                 mov ecx, [i]
77.                 mov ebx, [eax + ecx*4]
78.                 mov [dx1], ebx
79.                 add ecx, 1
80.                 mov ebx, [eax + ecx*4]
81.                 mov [dy1], ebx
82.                 mov ecx, [j]
83.                 mov ebx, [eax + ecx*4]

```

```

84.      mov [dx2], ebx
85.      add ecx, 1
86.      mov ebx, [eax + ecx*4]
87.      mov [dy2], ebx
88.
89.      mov ebx, [dx1]
90.      imul ebx, dword [dy2]
91.      mov [left], ebx
92.      mov ebx, [dy2]
93.      mov ebx, [dx2]
94.      imul ebx, dword [dy1]
95.      cmp ebx, [left]
96.      je PrintIsParallel
97.      cmp [i], 4
98.      je Finish
99.      cmp [j], 6
100.     je IsParallelLoop1
101.     cmp [j], 6
102.     jl IsParallelLoop2
103.     mov esp, [anc]
104.     ret
105.
106.     ;Печатает номера пары отрезков, если они параллельные
107.     PrintIsParallel:
108.     ; Номера отрезков определяются по значениям счетчиков
      (i / 2 + 1)
109.     mov eax, [i]
110.     mov ecx, 2
111.     mov edx, 0
112.     div ecx
113.     add eax, 1

```

```

114.          cinvoke printf, D, eax
115.          mov eax, [j]
116.          mov ecx, 2
117.          mov edx, 0
118.          div ecx
119.          add eax, 1
120.          cinvoke printf, D, eax
121.          cinvoke printf, newLine
122.          cmp [i], 4
123.          je Finish
124.          cmp [j], 6
125.          je IsParallelLoop1
126.          cmp [j], 6
127.          jl IsParallelLoop2
128.
129.      ; Окончание цикла
130.      Finish:
131.          ret
132.
133.      ; Формирует массив значений проекций отрезков на оси X и Y
134.      FormB:
135.          mov [anc], esp
136.          mov [i], -1
137.          mov [j], 0
138.          FormLoop1:
139.              ; Чтение координат концевых точек данного
отрезка
140.          mov eax, [A]
141.          add [i], 1
142.          mov ecx, [i]
143.          mov ebx, [eax + ecx*4]

```



```

144.          mov [x1], ebx
145.
146.          mov eax, [A]
147.          add [i], 1
148.          mov ecx, [i]
149.          mov ebx, [eax + ecx*4]
150.          mov [y1], ebx
151.
152.          mov eax, [A]
153.          add [i], 1
154.          mov ecx, [i]
155.          mov ebx, [eax + ecx*4]
156.          mov [x2], ebx
157.
158.          mov eax, [A]
159.          add [i], 1
160.          mov ecx, [i]
161.          mov ebx, [eax + ecx*4]
162.          mov [y2], ebx
163.
164.          ; Определние, какая из точек является правой
или левой (по абсциссе)
165.          mov ebx, [x1]
166.          ;Если x1 меньше x2, значит вторая точка
является правой
167.          cmp ebx, [x2]
168.          jl ProjectionX2
169.          ;Если x1 равно x2, значит отрезок параллелен
оси X
170.          cmp ebx, [x2]
171.          je ProjectionX2
172.          ;Если x1 больше x2, значит первая точка

```

является правой

```

173.                cmp ebx, [x2]
174.                jg ProjectionX1
175.
176.                mov esp, [anc]
177.                ret
178.
179.    ; Добавляет значения проекций в массив B, если первая точка
    является правой
180.    ProjectionX1:
181.        mov ebx, [x1]
182.        sub ebx, dword [x2]
183.        mov eax, [B]
184.        mov ecx, [j]
185.        mov [eax + 4*ecx], ebx
186.        add [j], 1
187.        mov ebx, [y1]
188.        sub ebx, dword [y2]
189.        mov eax, [B]
190.        mov ecx, [j]
191.        mov [eax + 4*ecx], ebx
192.        add [j], 1
193.        cmp [i], 16
194.        jl FormLoop1
195.        ret
196.
197.    ; Добавляет значения проекций в массив B, если вторая точка
    является правой или точки имеют одинаковое значение абсциссы
198.    ProjectionX2:
199.        mov ebx, [x1]
200.        sub dword [x2], ebx

```

```

201.      mov eax, [B]
202.      mov ecx, [j]
203.      imul ecx, dword 4
204.      add eax, ecx
205.      mov ebx, [x2]
206.      mov [eax], ebx
207.      add [j], 1
208.      mov ebx, [y1]
209.      sub dword [y2], ebx
210.      mov eax, [B]
211.      mov ecx, [j]
212.      imul ecx, dword 4
213.      add eax, ecx
214.      mov ebx, [y2]
215.      mov [eax], ebx
216.      add [j], 1
217.      cmp [i], 16
218.      j1 FormLoop1
219.      ret
220.
221.      ; Выделение памяти для массива A, в котором хранятся
      координаты концевых точек
222.      CreateA:
223.      mov [anc], esp
224.      invoke HeapCreate,HEAP_NO_SERIALIZE,0,0
225.      mov [heapA],eax
226.      invoke HeapAlloc,[heapA],HEAP_NO_SERIALIZE,N*8
227.      mov [A],eax
228.      mov esp, [anc]
229.      ret
230.

```

```

231.      ; Выделение памяти для массива B, в котором хранятся значения
        проекций
232.      CreateB:
233.          mov [anc], esp
234.          invoke HeapCreate,HEAP_NO_SERIALIZE,0,0
235.          mov [heapB], eax
236.          invoke HeapAlloc,[heapB],HEAP_NO_SERIALIZE,M2*8
237.          mov [B],eax
238.          mov esp, [anc]
239.          ret
240.
241.      ; Чтение координат концевых точек 1 отрезка
242.      ReadCoord1:
243.          mov esp, [anc]
244.          mov [anc], esp
245.          invoke printf, enterStr1
246.          mov [i], 0
247.          ReadLoop1:
248.              ;Чтение значения и проверка на границы
249.              cinvoke scanf, D, cur
250.              cmp [cur], -100
251.              jl WrongInput1
252.              cmp [cur], 100
253.              jg WrongInput1
254.
255.              mov eax, [A]
256.              mov ecx, [i]
257.
258.              mov ebx, [cur]
259.              mov [eax + 4*ecx], ebx
260.

```

```

261.          add [i], 1
262.          mov ecx, [i]
263.          cmp ecx, [M1]
264.          jne ReadLoop1
265.          ;Проверка, образуют ли введенные координаты точку
266.          mov eax, [A]
267.          mov ebx, [eax]
268.          mov [x1], ebx
269.          mov ebx, [eax + 4]
270.          mov [y1], ebx
271.          mov eax, [A]
272.          mov ebx, [eax + 8]
273.          mov [x2], ebx
274.          mov eax, [A]
275.          mov ebx, [eax + 12]
276.          mov [y2], ebx
277.          mov ebx, [x1]
278.          cmp ebx, [x2]
279.          je IsItPoint1
280.          mov esp, [anc]
281.          ret
282.
283.          ; Чтение координат концевых точек 2 отрезка
284.          ReadCoord2:
285.          mov esp, [anc]
286.          mov [anc], esp
287.          invoke printf, enterStr2
288.          mov [i], 4
289.          ReadLoop2:
290.          ;Чтение значения и проверка на границы
291.          cinvoke scanf, D, cur

```

```
292.          cmp [cur], -100
293.          jl WrongInput2
294.          cmp [cur], 100
295.          jg WrongInput2
296.
297.          mov eax, [A]
298.          mov ecx, [i]
299.
300.          mov ebx, [cur]
301.          mov [eax + 4*ecx], ebx
302.
303.          add [i], 1
304.          mov ecx, [i]
305.          cmp ecx, [M2]
306.          jne ReadLoop2
307.          ;Проверка, образуют ли введенные координаты точку
308.          mov eax, [A]
309.          mov ebx, [eax + 16]
310.          mov [x1], ebx
311.          mov ebx, [eax + 20]
312.          mov [y1], ebx
313.          mov eax, [A]
314.          mov ebx, [eax + 24]
315.          mov [x2], ebx
316.          mov eax, [A]
317.          mov ebx, [eax + 28]
318.          mov [y2], ebx
319.          mov ebx, [x1]
320.          cmp ebx, [x2]
321.          je IsItPoint2
322.          mov esp, [anc]
```

```

323.          ret
324.
325.      ; Чтение координат концевых точек 3 отрезка
326.      ReadCoord3:
327.          mov esp, [anc]
328.          mov [anc], esp
329.          invoke printf, enterStr3
330.          mov [i], 8
331.          ReadLoop3:
332.              ;Чтение значения и проверка на границы
333.              cinvoke scanf, D, cur
334.              cmp [cur], -100
335.              jl WrongInput3
336.              cmp [cur], 100
337.              jg WrongInput3
338.
339.              mov eax, [A]
340.              mov ecx, [i]
341.
342.              mov ebx, [cur]
343.              mov [eax + 4*ecx], ebx
344.
345.              add [i], 1
346.              mov ecx, [i]
347.              cmp ecx, [M3]
348.              jne ReadLoop3
349.          ;Проверка, образуют ли введенные координаты точку
350.          mov eax, [A]
351.          mov ebx, [eax + 32]
352.          mov [x1], ebx
353.          mov ebx, [eax + 36]

```

```

354.      mov [y1], ebx
355.      mov eax, [A]
356.      mov ebx, [eax + 40]
357.      mov [x2], ebx
358.      mov eax, [A]
359.      mov ebx, [eax + 44]
360.      mov [y2], ebx
361.
362.      mov ebx, [x1]
363.      cmp ebx, [x2]
364.      je IsItPoint3
365.      mov esp, [anc]
366.      ret
367.
368.      ; Чтение координат концевых точек 4 отрезка
369.      ReadCoord4:
370.      mov esp, [anc]
371.      mov [anc], esp
372.      invoke printf, enterStr4
373.      mov [i], 12
374.      ReadLoop4:
375.      ;Чтение значения и проверка на границы
376.      cinvoke scanf, D, cur
377.      cmp [cur], -100
378.      jl WrongInput4
379.      cmp [cur], 100
380.      jg WrongInput4
381.
382.      mov eax, [A]
383.      mov ecx, [i]
384.

```



```

385.             mov ebx, [cur]
386.             mov [eax + 4*ecx], ebx
387.
388.             add [i], 1
389.             mov ecx, [i]
390.             cmp ecx, [N]
391.             jne ReadLoop4
392.             ;Проверка, образуют ли введенные координаты точку
393.             mov eax, [A]
394.             mov ebx, [eax + 48]
395.             mov [x1], ebx
396.             mov ebx, [eax + 52]
397.             mov [y1], ebx
398.             mov eax, [A]
399.             mov ebx, [eax + 56]
400.             mov [x2], ebx
401.             mov eax, [A]
402.             mov ebx, [eax + 60]
403.             mov [y2], ebx
404.
405.             mov ebx, [x1]
406.             cmp ebx, [x2]
407.             je IsItPoint4
408.             mov esp, [anc]
409.             ret
410.
411.             ; Неверные данные при чтении данных отрезка 1
412.             WrongInput1:
413.             invoke printf, wrongStr
414.             jmp ReadLoop1
415.

```

```
416.      ; Неверные данные при чтении данных отрезка 2
417.      WrongInput2:
418.          invoke printf, wrongStr
419.          jmp ReadLoop2
420.
421.      ; Неверные данные при чтении данных отрезка 3
422.      WrongInput3:
423.          invoke printf, wrongStr
424.          jmp ReadLoop3
425.
426.      ; Неверные данные при чтении данных отрезка 4
427.      WrongInput4:
428.          invoke printf, wrongStr
429.          jmp ReadLoop4
430.
431.      ; Проверка, образуют ли введенные координаты для 1 отрезка
        точку
432.      IsItPoint1:
433.          mov ebx, [y1]
434.          cmp ebx, [y2]
435.          je ReadCoord1
436.          mov esp, [anc]
437.          ret
438.
439.      ; Проверка, образуют ли введенные координаты для 2 отрезка
        точку
440.      IsItPoint2:
441.          mov ebx, [y1]
442.          cmp ebx, [y2]
443.          je ReadCoord2
444.          mov esp, [anc]
```

```

445.          ret
446.
447.    ; Проверка, образуют ли введенные координаты для 3 отрезка
    точку
448.    IsItPoint3:
449.        mov ebx, [y1]
450.        cmp ebx, [y2]
451.        je ReadCoord3
452.        mov esp, [anc]
453.        ret
454.
455.    ; Проверка, образуют ли введенные координаты для 4 отрезка
    точку
456.    IsItPoint4:
457.        mov ebx, [y1]
458.        cmp ebx, [y2]
459.        je ReadCoord4
460.        mov esp, [anc]
461.        ret
462.
463.
464.    section 'idata' import data readable
465.
466.        library kernel, 'kernel32.dll',\
467.            msvcrt, 'msvcrt.dll',\
468.            user32, 'USER32.DLL'
469.            include 'api\user32.inc'
470.            include 'api\kernel32.inc'
471.
472.        import kernel,\
473.            ExitProcess, 'ExitProcess',\

```

```
474.             HeapCreate, 'HeapCreate',\  
475.             HeapFree, 'HeapFree',\  
476.             HeapAlloc, 'HeapAlloc'  
477.             include 'api\kernel32.inc'  
478.  
479.             import msvcrt,\  
480.             printf, 'printf',\  
481.             scanf, 'scanf',\  
482.             getch, '_getch'
```