

```
1 {
  File:      interval.dpr

  Compiler. Borland Delphi 7.0
```

Задача. Два отрезка на плоскости заданы целочисленными координатами своих концов в декартовой системе координат. Требуется определить, существует ли у них общая точка.

Входной файл input.txt содержит координаты четырех точек, задающих отрезки. В первой строке содержатся координаты первого конца первого отрезка, во второй – второго конца первого отрезка, в третьей и четвертой – координаты концов второго отрезка. Все координаты – целые числа, не превосходящие 10000 по абсолютной величине.

В выходной файл OUTPUT.TXT следует вывести слово "Yes", если общая точка есть, или слово "No" – в противном случае.

Input.txt	Output.txt
0 0	Yes
1 1	
0 1	
1 0	

Решение. Калмыков Вадим (ProCrypt),
г. Сургут, ЦНИТ "Северная Звезда",
14:28, 07.06.2008

Source : <http://acm.dvpion.ru>

Editor. MSP, 14.12.09, 15:36; LIST 6.0, (c) 1999–2009

```
}
```

```
30 {$N+}
31 Program Intervals;

32 Const
33     Eps = 10e-10;

34 Var
35     dx2, dy2,
36     dx1, dy1 : Integer; // изменение координаты dx – "дельта x"

37     k1, k2 : Extended; // угловые коэффициенты прямых

38     C1, C2 : Extended; // постоянные в уравнении прямой
```

```

39     x, y : Extended; // координаты точки пересечения прямых
40     x1, x2, x3, x4,
41     y1, y2, y3, y4 : Integer; // координаты концов отрезков

42     // функция вычисления расстояния между двумя точками
43     Function Len (x1, y1, x2, y2 : Extended) : Extended;

44     begin
45         Len := Sqrt (Sqr (x2 - x1) + Sqr (y2 - y1));
46     end;

47     Begin
48         Assign (Input, 'input.txt');
49         Assign (Output, 'output.txt');

50         Reset (Input);
51         Rewrite (Output);

52         ReadLn (x1, y1);
53         ReadLn (x2, y2);
54         ReadLn (x3, y3);
55         ReadLn (x4, y4);

56         Close (Input);

57         {
           Для удобства работы с отрезками обеспечим, что x1 лежит левее
           x2, и, также x3 левее x4.
         }
61         if (x1 > x2)
62             then
63                 begin
64                     dx1 := x1; // используем dx1 для обмена значений
65                     x1 := x2;
66                     x2 := dx1;

67                     dx1 := y1;
68                     y1 := y2;
69                     y2 := dx1;
70                 end;

71         // проделываем то же самое и с координатами второго отрезка
72         if (x3 > x4)
73             then
74                 begin
75                     dx1 := x3;
76                     x3 := x4;
77                     x4 := dx1;

```

```

78         dx1 := y3;
79         y3 := y4;
80         y4 := dx1;
81     end;

82     {
        Найдём уравнения прямых, на которых лежат заданные отрезки.
         $y := kx + C$ ,  $k$  – угловой коэффициент,  $C$  – какая – то постоянная

            dy
        k = --
            dx
    }

89     // находим уравнение прямой, содержащей первый отрезок
90     dx1 := x2 - x1; // находим изменение координаты по оси OX
91     dy1 := y2 - y1; // находим изменение координаты по оси OY

92     if (dx1 <> 0)
93     then
94         begin
95             k1 := dy1 / dx1;

96             // вычисляем постоянную C1
97             {
                Так как наше уравнение прямой имеет вид  $y = kx + C$ , то
                 $C = y - kx$ , где  $x$  и  $y$  координаты любой точки, лежащей на
                данной прямой. Отрезок наш лежит на этой прямой, поэтому
                 $C1 = y1 - k1x1$ 
            }
103            C1 := y1 - k1 * x1;
104        end;

105     // находим уравнение прямой, содержащей второй отрезок
106     dx2 := x4 - x3; // находим изменение координаты по оси OX
107     dy2 := y4 - y3; // находим изменение координаты по оси OY

108     if (dx2 <> 0)
109     then
110         begin
111             k2 := dy2 / dx2;

112             // вычисляем постоянную C2
113             C2 := y3 - k2 * x3;
114         end;

115     if (dx1 <> 0)
116     and
117     (dx2 <> 0)
118     then
119         if (Abs (k1 - k2) < Eps) // угловые коэффициенты равны

```

```

120     then
121         if (Abs (C1 - C2) > Eps) // постоянные не равны
122             then
123                 begin
124                     // в этом случае прямые, а следовательно и отрезки,
125                     // параллельны
126                     Write ('No');
127                     Close (Output);

128                     Exit;
129                 end
130             else
131             {
                Если  $k_1 = k_2$ ,  $C_1 = C_2$ , то отрезки лежат на одной
                прямой. Поэтому, если они пересекаются, или
                правильнее сказать, один продолжает другой, то одна
                из координат концов первого отрезка лежит между
                координатами другого
            }
138         if (x1 >= x3)
139             and
140             (x1 <= x4)
141             or
142             (x2 >= x3)
143             and
144             (x2 <= x4)
145         then
146             begin
147                 Write ('Yes');
148                 Close (Output);

149                 Exit;
150             end
151         else
152             begin
153                 Write ('No');
154                 Close (Output);

155                 Exit;
156             end
157     else
158     begin
159     {
                Если угловые коэффициенты не равны, то прямые
                пересекаются. Найдем точку пересечения прямых.

                 $y = k_1x + C_1$ ,
                 $y = k_2x + C_2$ , раз они пересекаются, то имеют общие
                точки  $x$  и  $y$ . Поэтому  $k_1x + C_1 = k_2x + C_2$ , откуда

                 $C_2 - C_1$ 
            }

```

$$x = \frac{\dots}{k_1 - k_2},$$

ординату y вычислим подстановкой x в любое уравнение.
}

```
170 x := (C2 - C1) / (k1 - k2);
171 y := k1 * x + C1;
```

```
172 {
    Отрезки пересекаются, если точка пересечения прямых
    принадлежит обоим отрезкам. Если точка C принадлежит
    отрезку AB, то AC + BC = AB, иначе получим треугольник
    ABC. Из неравенства треугольника следует,
    что AC + BC > AB
}
```

```
179 if (Abs (Len (x1, y1, x, y) + Len (x, y, x2, y2)
180       - Len (x1, y1, x2, y2)) < Eps)
181     and
182     (Abs (Len (x3, y3, x, y) + Len (x, y, x4, y4)
183       - Len (x3, y3, x4, y4)) < Eps)
184     then
185       Write ('Yes')
186     else
187       Write ('No');
```

```
188     Close (Output);
```

```
189     Exit;
```

```
190   end
```

```
191 else
```

```
192   begin
```

```
193     {
        Если dx1 или dx2 равны нулю, то одна из прямых
        вертикальная (или обе, если dx1 = dx2 = 0).
        Для определенности в пример возьмем dx1 = 0.
        Тогда x = x1 = x2, так как на других x
        первая прямая не определена, это единственный вариант.
        Тогда y = k2 * x + C2
    }
```

```
201   if (dx1 = 0)
```

```
202     then
```

```
203       if (dx2 <> 0)
```

```
204         then
```

```
205           begin
```

```
206             x := x1;
```

```
207             y := k2 * x + C2;
```

```
208           if (Abs (Len (x1, y1, x, y) + Len (x, y, x2, y2)
209                 - Len (x1, y1, x2, y2)) < Eps)
```

```

210         and
211         (Abs (Len (x3, y3, x, y) + Len (x, y, x4, y4)
212             - Len (x3, y3, x4, y4)) < Eps)
213     then
214         Write ('Yes')
215     else
216         Write ('No');
217
218         Close (Output);
219
220     Exit;
221 end
222 else
223     begin
224     {
225         Если обе прямые вертикальны, то они параллельны,
226         если  $x1 \neq x3$ . В противном случае, прямые
227         совпадают, поэтому отрезки имеют общие точки,
228         если одна из координат у первого отрезка лежит
229         между двумя другими второго. Но для этого
230         (как в начале программы мы делали для x)
231         придется обеспечить, чтобы  $y1$  лежал ниже  $y2$ ,
232         аналогично  $y3 < y4$ 
233     }
234     if (x1 <> x3)
235     then
236     begin
237         Write ('No');
238         Close (Output);
239
240         Exit;
241     end
242     else
243     begin
244         if (y1 > y2)
245         then
246         begin
247             dx1 := y1;
248             y1 := y2;
249             y2 := dx1;
250
251             dx1 := x1;
252             x1 := x2;
253             x2 := dx1;
254         end;
255
256         if (y3 > y4)
257         then
258         begin
259             dx1 := y3;
260             y3 := y4;

```

```

256             y4 := dx1;

257             dx1 := x3;
258             x3 := x4;
259             x4 := dx1;
260         end;

261         if (y1 >= y3)
262             and
263             (y1 <= y4)
264             or
265             (y2 >= y3)
266             and
267             (y2 <= y4)
268         then
269             Write ('Yes')
270         else
271             Write ('No');

272             Close (Output);

273             Exit;
274         end;
275     end
276 else
277     { dx1 <> 0, остается только проверить dx2 }
278     if (dx2 = 0)
279         then
280             begin
281                 x := x3;
282                 y := k1 * x + C1;

283                 if (Abs (Len (x1, y1, x, y) + Len (x, y, x2, y2)
284                     - Len (x1, y1, x2, y2)) < Eps)
285                     and
286                     (Abs (Len (x3, y3, x, y) + Len (x, y, x4, y4)
287                         - Len (x3, y3, x4, y4)) < Eps)
288                 then
289                     Write ('Yes')
290                 else
291                     Write ('No');

292                 Close (Output);

293                 Exit;
294             end;
295         end;
296     End.

```

Listing данной задачи опубликован в сети Internet по адресу
<http://www.Best-Listing.ru/color-10-task-660.html>

Sergey Mitrofanov, 13.09.14, 13:02

E-mail: infostar@mail.ru

© <http://www.Best-Listing.ru/>, 2006–2014