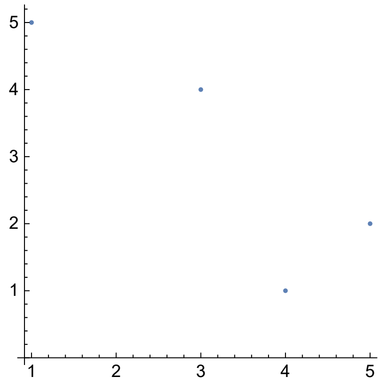
```
(* описаниые нормы в плоскости *)
(* порожденной случайным набором точек lstA в первом квадранте *)
(* !! выпуклая, центрально симметричная орестность нуля *)
lstA = Table[{0, 0}, {nt, 1, 16}];
lstA[[1]] = {4, 1};
lstA[[2]] = {5, 2};
lstA[[3]] = {3, 4};
lstA[[4]] = {1, 5};
(* положение точек в первом квадранте*)
ListPlot[{lstA[[1]], lstA[[2]], lstA[[3]], lstA[[4]]}, AspectRatio → 1]
```

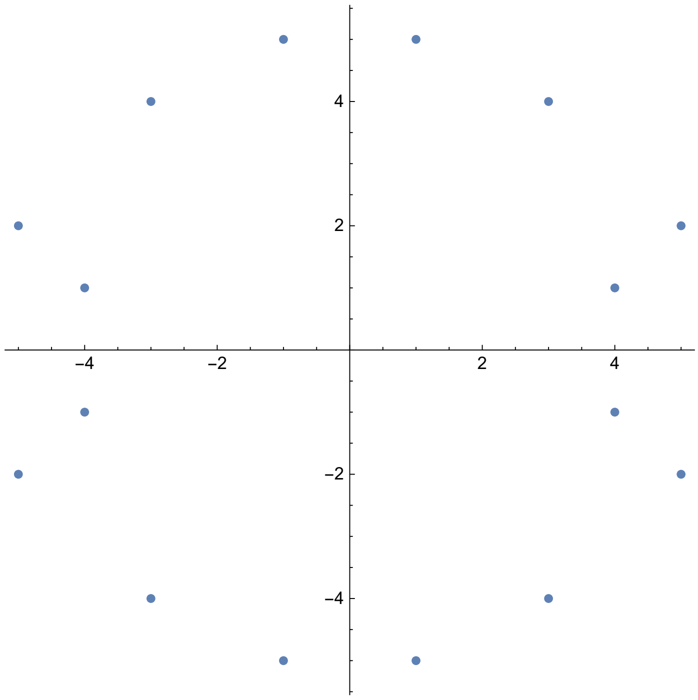
Out[∘]=



In[∘]:=
```
(* продолжение во второй квадрант -- "усиленная" симметрия *)
lstA[[5]] = {-lstA[[4, 1]], lstA[[4, 2]]};
lstA[[6]] = {-lstA[[3, 1]], lstA[[3, 2]]};
lstA[[7]] = {-lstA[[2, 1]], lstA[[2, 2]]};
lstA[[8]] = {-lstA[[1, 1]], lstA[[1, 2]]};
(* третий квадрант *)
lstA[[9]] = {-lstA[[1, 1]], -lstA[[1, 2]]};
lstA[[10]] = {-lstA[[2, 1]], -lstA[[2, 2]]};
lstA[[11]] = {-lstA[[3, 1]], -lstA[[3, 2]]};
lstA[[12]] = {-lstA[[4, 1]], -lstA[[4, 2]]};
(* четвертый квадрант *)
lstA[[13]] = {lstA[[4, 1]], -lstA[[4, 2]]};
lstA[[14]] = {lstA[[3, 1]], -lstA[[3, 2]]};
lstA[[15]] = {lstA[[2, 1]], -lstA[[2, 2]]};
lstA[[16]] = {lstA[[1, 1]], -lstA[[1, 2]]};
(* положение всех точек *)
Print[lstA];
ListPlot[lstA, AspectRatio → 1]
```

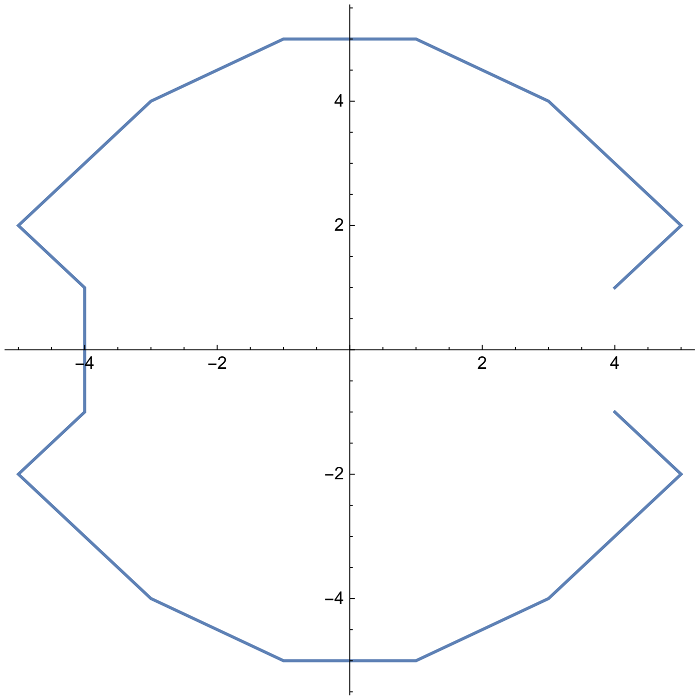
```
{{4, 1}, {5, 2}, {3, 4}, {1, 5}, {-1, 5}, {-3, 4}, {-5, 2}, {-4, 1},
 {-4, -1}, {-5, -2}, {-3, -4}, {-1, -5}, {1, -5}, {3, -4}, {5, -2}, {4, -1}}
```

Out[∘]=



In[∘]:=
```
(* последовательный обход всех точек *)
ListLinePlot[lstA, AspectRatio → 1]
```

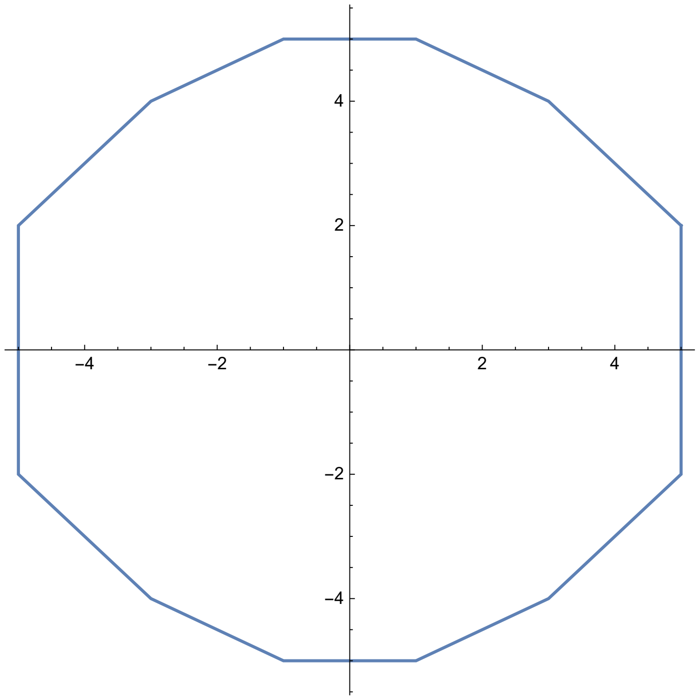
Out[∘]=



In[∘]:=
```
(* исключение "внутренних" точек ,  многоуольник, опредляющий норму*)
(* ||x||=1 -- "x"- лежит на границе*)
(* ||x||=N -- "1/N x"- лежит на границе*)
(* вершины -- КРАЙНИЕ ТОЧКИ *)
lstA = {lstA[[2]], lstA[[3]], lstA[[4]], lstA[[5]], lstA[[6]], lstA[[7]], lstA[[10]],
   lstA[[11]], lstA[[12]], lstA[[13]], lstA[[14]], lstA[[15]], lstA[[2]]};
Print[lstA];
ListLinePlot[lstA, AspectRatio → 1]
```

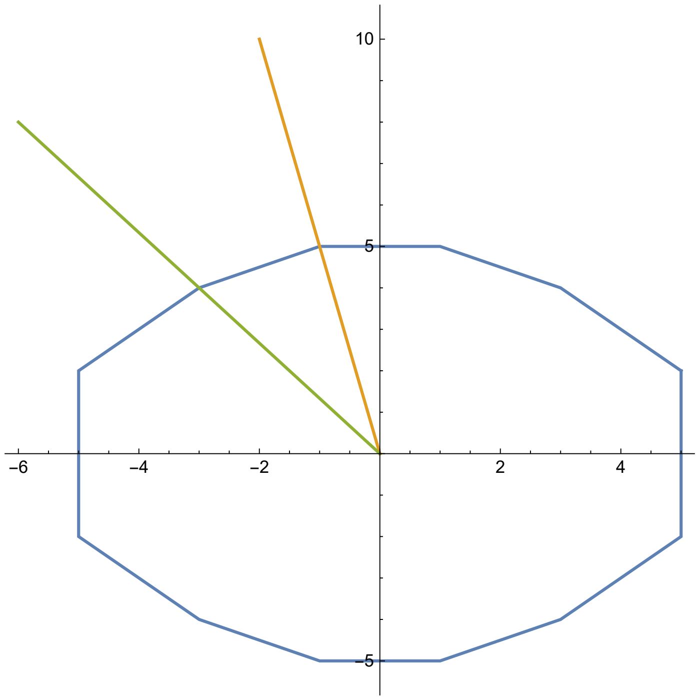
```
{{5, 2}, {3, 4}, {1, 5}, {-1, 5}, {-3, 4}, {-5, 2},
 {-5, -2}, {-3, -4}, {-1, -5}, {1, -5}, {3, -4}, {5, -2}, {5, 2}}
```

Out[∘]=



In[•]:=
```
(* Вычисление  нормы в угле, образованном соседними вершинами *)
rb = 4; re = rb + 1; (* номера соседних вершин *)
lstLb = {{0, 0}, 2 lstA[[rb]]};
lstLe = {{0, 0}, 2 lstA[[re]]};
(* рисунок угла, в котором вычислим норму*)
ListLinePlot[{lstA, lstLb, lstLe}, AspectRatio → 1]
```

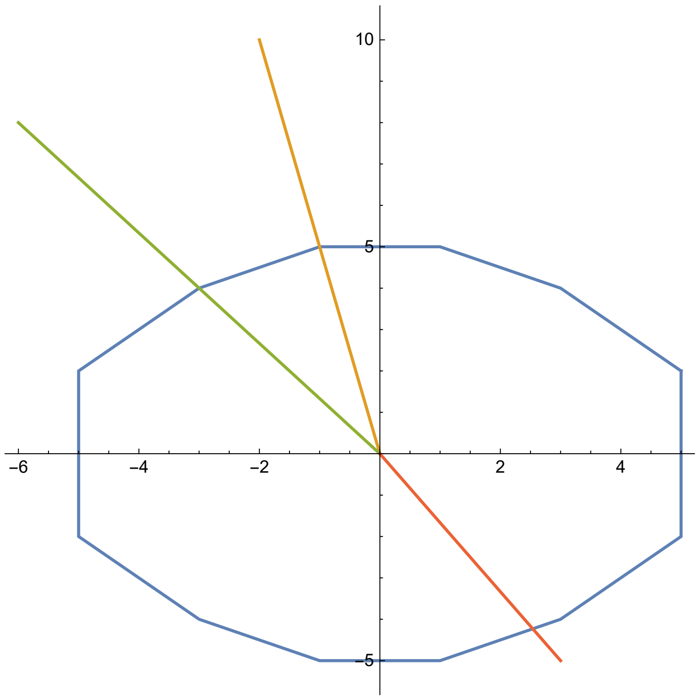
Out[•]=



```
(* базис угла *)
vbb = lstA[[rb]];
vbe = lstA[[re]];
(* двойственный ( биортогональный ) базис *)
voe = {-lstA[[rb, 2]], lstA[[rb, 1]]};
vob = {-lstA[[re, 2]], lstA[[re, 1]]};
(* номировка ортогонального  базиса *)
kn =  vbb[[1]] × vob[[1]] + vbb[[2]] × vob[[2]];
vob = 1 / kn vob;

kn =  vbe[[1]] × voe[[1]] + vbe[[2]] × voe[[2]];
voe = 1 / kn voe;
(* вычисление координат произвольной точки в базисе угла *)
P = {RandomInteger[{-7, 7}], RandomInteger[{-7, 7}]};
lstP = {{0, 0}, P};
(* многоугольник, угол, "точка" *)
ListLinePlot[{lstA, lstLb, lstLe, lstP}, AspectRatio → 1]
(* P= kb vbb+ ke vbe*)
kb = P[[1]] × vob[[1]] + P[[2]] × vob[[2]];
ke = P[[1]] × voe[[1]] + P[[2]] × voe[[2]];
Print[{"kb", kb, "ke", ke, "kb+ke", kb + ke}];
```

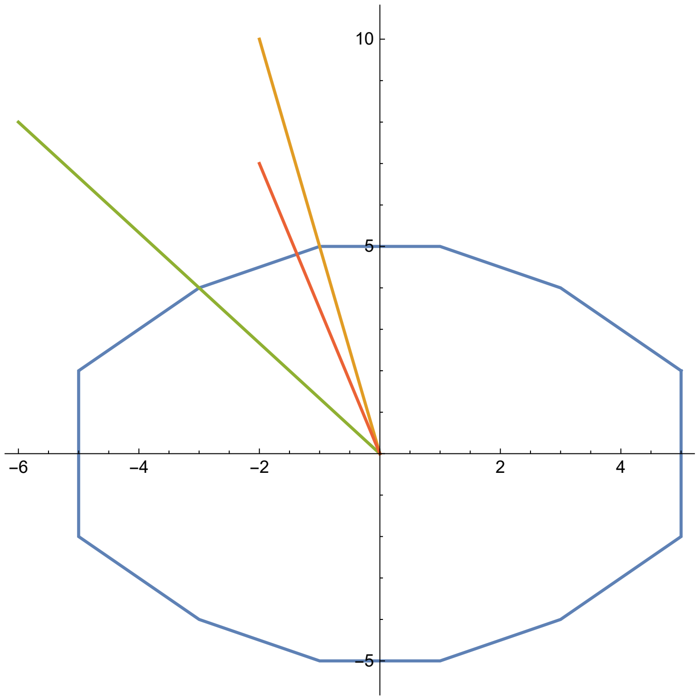
Out[•]=



$\left\{kb, -\frac{3}{11}, ke, -\frac{10}{11}, kb{+}ke, -\frac{13}{11}\right\}$

In[•]:=
```
(* точка внутри угла *)
P = {-2, 7}; lstP = {{0, 0}, P};
ListLinePlot[{lstA, lstLb, lstLe, lstP}, AspectRatio → 1]
(* P= kb vbb+ ke vbe*)
kb = P[[1]] × vob[[1]] + P[[2]] × vob[[2]];
ke = P[[1]] × voe[[1]] + P[[2]] × voe[[2]];
Print[{"kb", kb, "ke", ke, "kb+ke", kb + ke}];
```

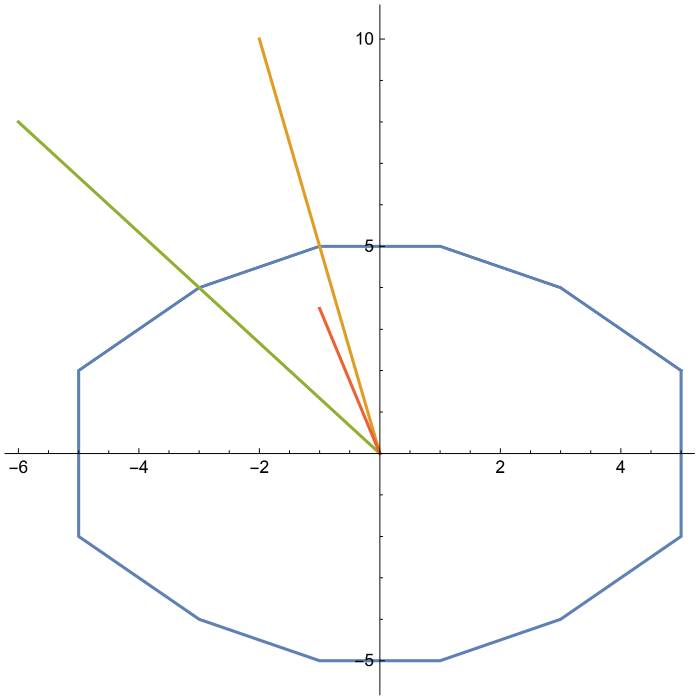
Out[•]=



$\left\{\text{kb}, \frac{13}{11}, \text{ke}, \frac{3}{11}, \text{kb+ke}, \frac{16}{11}\right\}$

In[ ]:=
```
(* перемещение точки по своему лучу*)
P = 1/2 {-2, 7}; lstP = {{0, 0}, P};
ListLinePlot[{lstA, lstLb, lstLe, lstP}, AspectRatio → 1]
(* P= kb vbb+ ke vbe*)
kb = P[[1]] × vob[[1]] + P[[2]] × vob[[2]];
ke = P[[1]] × voe[[1]] + P[[2]] × voe[[2]];
Print[{"kb", kb, "ke", ke, "kb+ke", kb + ke}];
```

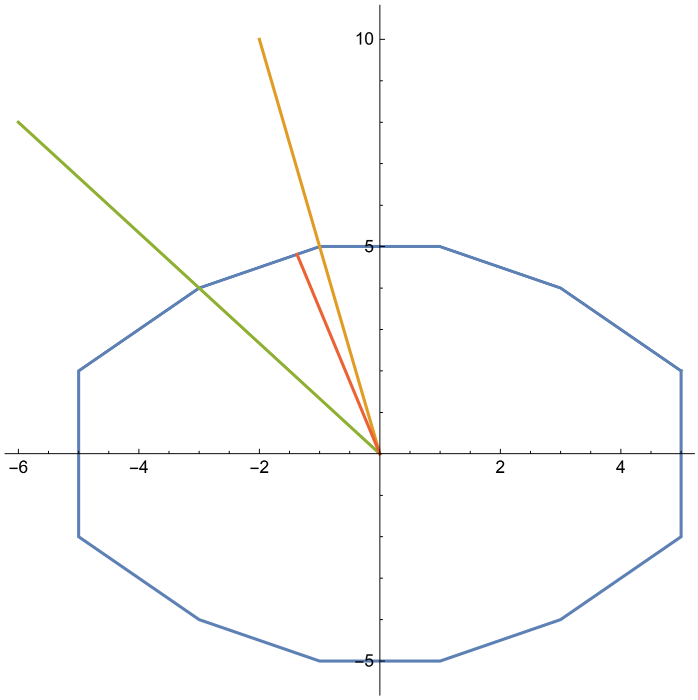
Out[ ]=



$\left\{kb, \frac{13}{22}, ke, \frac{3}{22}, kb+ke, \frac{8}{11}\right\}$

In[∘]:=
```
(* перемещение точки на границу -- "нормировка"*)
(* !!!    ||P||=1 *)
P = 11/16 {-2, 7}; lstP = {{0, 0}, P};
ListLinePlot[{lstA, lstLb, lstLe, lstP}, AspectRatio → 1]
(* P= kb vbb+ ke vbe*)
kb = P[[1]] × vob[[1]] + P[[2]] × vob[[2]];
ke = P[[1]] × voe[[1]] + P[[2]] × voe[[2]];
Print[{"kb", kb, "ke", ke, "kb+ke", kb + ke}];
```

Out[∘]=



$\left\{\text{kb}, \frac{13}{16}, \text{ke}, \frac{3}{16}, \text{kb+ke}, 1\right\}$

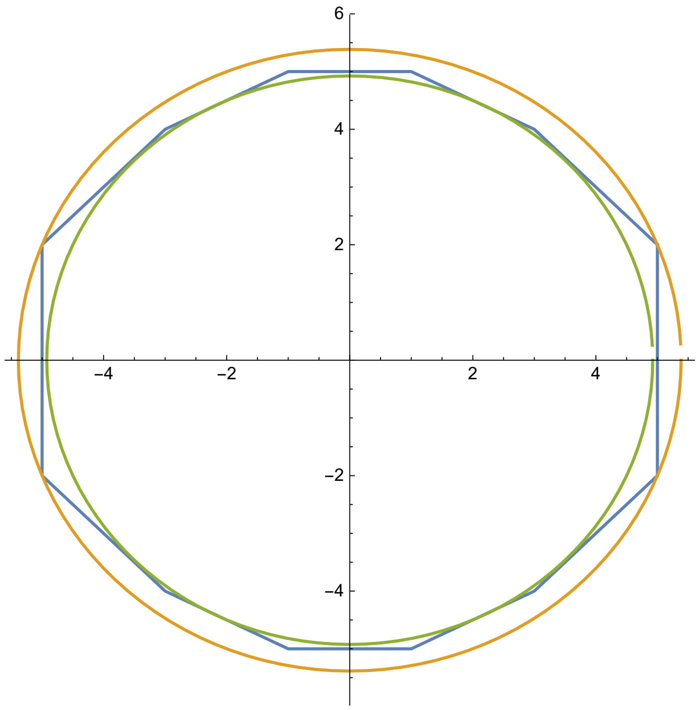
```
(* алгоритм вычисления ||P|| *)
(* найти угол между двумя соседними вершинами, которому принадлежит точка *)
(* ! обе координаты НЕ отрицательны *)
(* ||P|| =  сумма коэффициентов *)
```

In[∘]:=
```
(*эквивалентность норм*)
M = Max[(lstA[[1, 1]]^2 + lstA[[1, 2]]^2)^(1/2), (lstA[[2, 1]]^2 + lstA[[2, 2]]^2)^
    (1/2), (lstA[[3, 1]]^2 + lstA[[3, 2]]^2)^(1/2)];
Q1 = 1/2 (lstA[[1]] + lstA[[12]]);
Q2 = 1/2 (lstA[[1]] + lstA[[2]]);
Q3 = 1/2 (lstA[[2]] + lstA[[3]]);
Q4 = 1/2 (lstA[[3]] + lstA[[4]]);
```

```
m = Min[(Q1[[1]]^2 + Q1[[2]]^2)^(1/2), (Q2[[1]]^2 + Q2[[2]]^2)^(1/2),
    (Q3[[1]]^2 + Q3[[2]]^2)^(1/2), (Q4[[1]]^2 + Q4[[2]]^2)^(1/2)];
(*  max[  (x1^1+x2^2)^(1/2,  ||(x1,x2)||=1  ]= M *)
(*  min[  (x1^1+x2^2)^(1/2),  ||(x1,x2)||=1  ]= m *)

(* m  (x1^1+x2^2)^(1/2)  ≤  ||(x1,x2||  ≤  M  (x1^1+x2^2)^(1/2)*)
(* "многоуольная" норма любого элемента  больше чем
 "m" евклидовых норм и меньше чем "M" евклидовых норм *)
lstRM = Table[M {Cos[2 Pi / 120 kt], Sin[2 Pi / 120 kt]}, {kt, 1, 120}];
lstRm = Table[m {Cos[2 Pi / 120 kt], Sin[2 Pi / 120 kt]}, {kt, 1, 120}];
ListLinePlot[{lstA, lstRM, lstRm}, AspectRatio → 1]
```

Out[ ]=




```
(* координаты в пространственном угле *)
(* вектора определяющие угол *)
A = {RandomInteger[{2, 12}], RandomInteger[{2, 12}], RandomInteger[{2, 12}]};
B = {RandomInteger[{2, 12}], RandomInteger[{2, 12}], RandomInteger[{2, 12}]};
H = {RandomInteger[{2, 12}], RandomInteger[{2, 12}], RandomInteger[{2, 12}]};
Print[A];
Print[B];
Print[H];
```

{10, 3, 12}

{11, 12, 12}

{11, 4, 7}

```
(* вормирование вектора ортогонального двум данным *)
prO[a_, b_] := (
   m1 = {{a[[2]], a[[3]]}, {b[[2]], b[[3]]}};
   m2 = {{a[[1]], a[[3]]}, {b[[1]], b[[3]]}};
   m3 = {{a[[1]], a[[2]]}, {b[[1]], b[[2]]}};
   vO = {Det[m1], -Det[m2], Det[m3]}
  );
(* скалярное произведение *)
prS[a_, b_] := (sp = a[[1]] × b[[1]] + a[[2]] × b[[2]] + a[[3]] × b[[3]]);
(*формировани двойственного (биортогонального)  базиса  Ao,Bo,Ho *)
prO[A, B];
Ho = vO;
prS[Ho, H];
sp = Abs[sp];
Ho = sp^(-1) Ho;
Print[{prS[A, Ho], prS[B, Ho], prS[H, Ho]}];

prO[A, H];
Bo = vO;
prS[Bo, B];
sp = Abs[sp];
Bo = sp^(-1) Bo;
Print[{prS[A, Bo], prS[B, Bo], prS[H, Bo]}];

prO[B, H];
Ao = vO;
prS[Ao, A];
sp = Abs[sp];
Ao = sp^(-1) Ao;
Print[N[{prS[A, Ao], prS[B, Ao], prS[H, Ao]}]];
```

{0, 0, -1}

{0, 1, 0}

{-1., 0., 0.}

```
(* разложение вектора P по базису A,B,H *)
P = {RandomInteger[{-12, 12}], RandomInteger[{-12, 12}], RandomInteger[{-12, 12}]};
Print[P];
(* P= ka A+kb B+ kh H*)
prS[P, Ao];
ka = sp;

prS[P, Bo];
kb = sp;

prS[P, Ho];
kh = sp;
Print[{ka, kb, kh}];
```

{11, 9, 3}

$\left\{\frac{209}{177}, \frac{94}{177}, -\frac{91}{59}\right\}$

```
Do[(
   lstA = Table[{0, 0}, {nt, 1, 4}];
   lstA[[1]] = {RandomInteger[{2, 6}], RandomInteger[{2, 6}]};
   lstA[[2]] = {RandomInteger[{2, 12}], RandomInteger[{2, 12}]};
   lstA[[3]] = {RandomInteger[{2, 12}], RandomInteger[{2, 12}]};
   lstA[[4]] = {RandomInteger[{2, 12}], RandomInteger[{2, 12}]};
   Print[{"v", vt, lstA}]
  ), {vt, 1, 30}];
(* самостоятельная рабта *)
(* построить мноооольник определяющий норму, (* исходя их данных точек *)
```

```
In[1]:= cond = {
   {"v", 1, {{2, 2}, {6, 10}, {4, 7}, {11, 2}}},
{"v", 2, {{2, 2}, {8, 9}, {6, 3}, {6, 2}}},
{"v", 3, {{4, 4}, {10, 2}, {8, 2}, {5, 6}}},
{"v", 4, {{5, 4}, {5, 9}, {6, 3}, {12, 10}}},
{"v", 5, {{3, 2}, {4, 4}, {8, 3}, {7, 9}}},
{"v", 6, {{4, 5}, {2, 3}, {2, 9}, {12, 9}}},
{"v", 7, {{4, 2}, {9, 9}, {6, 5}, {9, 8}}},
{"v", 8, {{3, 5}, {8, 7}, {4, 10}, {3, 9}}},
{"v", 9, {{6, 5}, {2, 9}, {7, 8}, {2, 12}}},
{"v", 10, {{5, 4}, {4, 11}, {10, 12}, {8, 3}}},
{"v", 11, {{5, 4}, {3, 4}, {4, 5}, {3, 4}}},
{"v", 12, {{6, 6}, {10, 10}, {3, 4}, {2, 4}}},
{"v", 13, {{5, 4}, {4, 8}, {5, 10}, {5, 5}}},
{"v", 14, {{5, 3}, {9, 9}, {9, 11}, {5, 10}}},
{"v", 15, {{2, 4}, {8, 3}, {12, 10}, {6, 7}}},
{"v", 16, {{4, 5}, {5, 2}, {8, 12}, {7, 5}}},
{"v", 17, {{6, 2}, {10, 9}, {2, 3}, {5, 7}}},
{"v", 18, {{4, 3}, {5, 11}, {5, 7}, {3, 2}}},
{"v", 19, {{3, 5}, {2, 7}, {3, 4}, {12, 7}}},
{"v", 20, {{2, 6}, {12, 11}, {3, 8}, {7, 6}}},
{"v", 21, {{6, 4}, {11, 10}, {11, 4}, {4, 11}}},
{"v", 22, {{5, 4}, {12, 12}, {12, 4}, {5, 3}}},
{"v", 23, {{4, 3}, {10, 5}, {3, 9}, {6, 6}}},
{"v", 24, {{4, 3}, {12, 7}, {2, 10}, {6, 4}}},
{"v", 25, {{6, 6}, {10, 11}, {12, 2}, {8, 2}}},
{"v", 26, {{2, 3}, {9, 5}, {4, 9}, {10, 6}}},
{"v", 27, {{2, 4}, {8, 7}, {12, 11}, {9, 8}}},
{"v", 28, {{3, 3}, {9, 6}, {11, 12}, {10, 6}}},
{"v", 29, {{6, 3}, {8, 11}, {12, 5}, {10, 10}}},
{"v", 30, {{5, 5}, {4, 7}, {5, 10}, {7, 11}}}};
```