

```
In[ ]:= vertex = {1, 2, 3, 4, 5, 6, 7}
```

```
Out[ ]:= {1, 2, 3, 4, 5, 6, 7}
```

```
In[ ]:= edge = {1 ↔ 7, 1 ↔ 3, 2 ↔ 1, 2 ↔ 5, 3 ↔ 6,  
               3 ↔ 4, 4 ↔ 7, 5 ↔ 1, 5 ↔ 3, 5 ↔ 4, 6 ↔ 5, 7 ↔ 3, 7 ↔ 5}
```

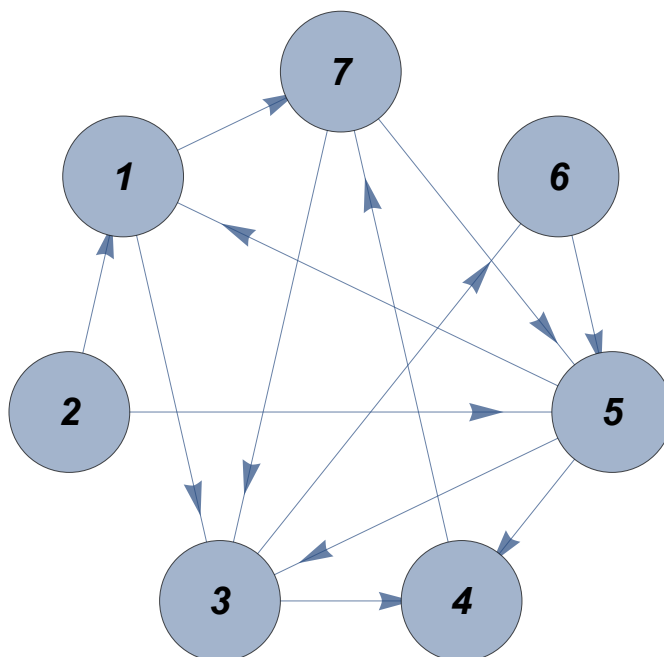
```
Out[ ]:= {1 ↔ 7, 1 ↔ 3, 2 ↔ 1, 2 ↔ 5, 3 ↔ 6, 3 ↔ 4,  
          4 ↔ 7, 5 ↔ 1, 5 ↔ 3, 5 ↔ 4, 6 ↔ 5, 7 ↔ 3, 7 ↔ 5}
```

```
In[ ]:=
```

```
In[ ]:=
```

```
In[ ]:= Graph[vertex, edge, VertexLabels → Placed["Name", Center],  
          [граф [метки для вершин [расположен [центр  
          VertexLabelStyle → Directive[Black, Italic, 20],  
          [стиль меток вершин [директива [чёрный [курсив  
          GraphLayout → "CircularEmbedding", VertexSize → 0.5,  
          [укладка графа [размер вершины  
          EdgeShapeFunction → GraphElementData["Arrow", "ArrowSize" → 0.05]]  
          [функция формы ребра [стрелка
```

```
Out[ ]:=
```



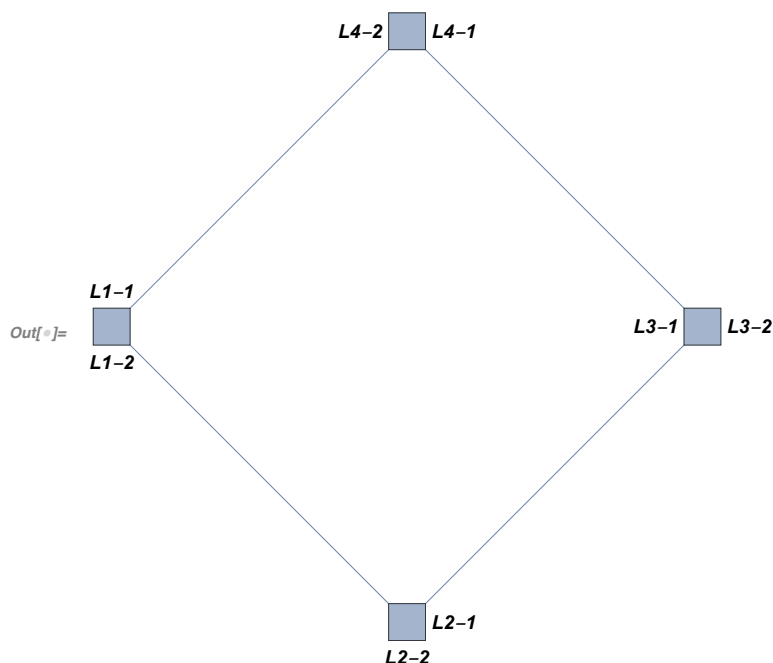
```
In[ ]:= vertex = {1, 2, 3, 4}
```

```
Out[ ]:= {1, 2, 3, 4}
```

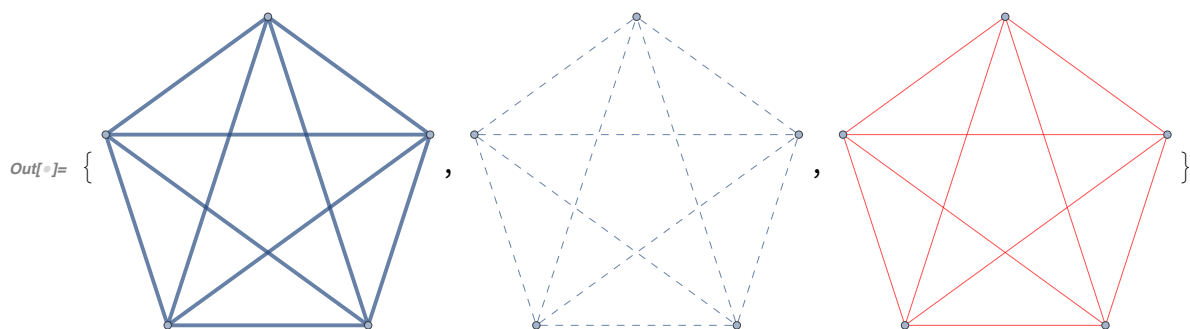
```
In[ ]:= edge = {1 ↔ 2, 2 ↔ 3, 3 ↔ 4, 4 ↔ 1}
```

```
Out[ ]:= {1 ↔ 2, 2 ↔ 3, 3 ↔ 4, 4 ↔ 1}
```

```
In[ ]:= Graph[{Labeled[1, Placed[{"L1-1", "L1-2"}, {Above, Below}]],
  Labeled[2, Placed[{"L2-1", "L2-2"}, {After, Below}]],
  Labeled[3, Placed[{"L3-1", "L3-2"}, {Before, After}]],
  Labeled[4, Placed[{"L4-1", "L4-2"}, {After, Before}]]}, edge,
  VertexShapeFunction → "Square", VertexLabelStyle → Directive[Black, Italic, 10],
  GraphLayout → "CircularEmbedding", VertexSize → 0.1]
```



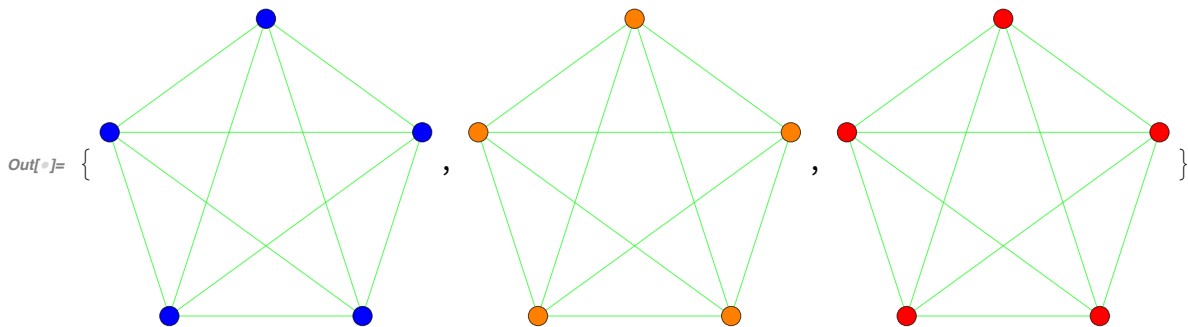
```
In[ ]:= Table[CompleteGraph[5, EdgeStyle → style], {style, {Thick, Dashed, Red}}]
```



```

In[ ]:= Table[CompleteGraph[5, EdgeStyle -> Green, VertexSize -> Small,
  [табл... [полный граф [стиль ребра [зелё... [размер верш... [малый
    VertexStyle -> style], {style, {Blue, Orange, Red}}]
    [стиль вершины [синий [оранж... [красный

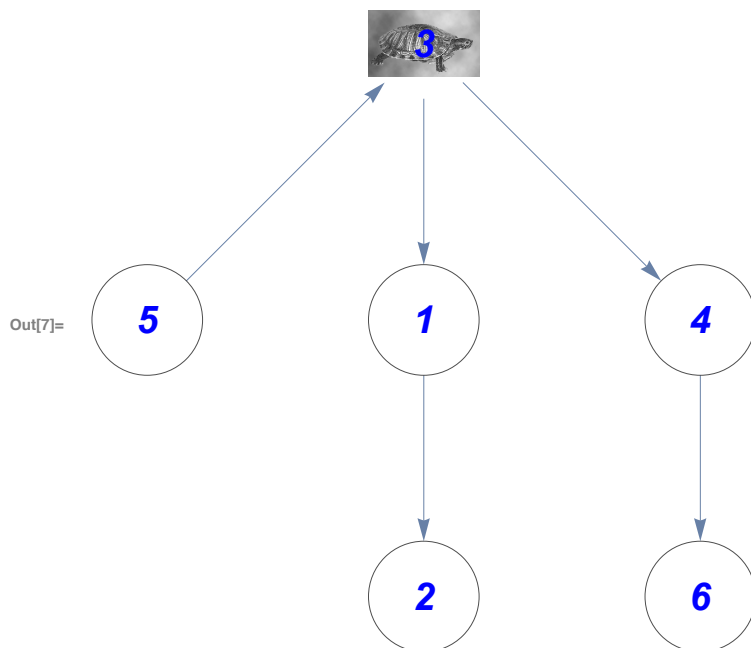
```



```

In[7]:= Graph[{1 -> 2, 3 -> 1, 3 -> 4, 4 -> 6, 5 -> 3}, VertexSize -> Large,
  [граф [размер верш... [крупный
  VertexShape -> {3 -> Import["ExampleData/turtle.jpg"]},
  [форма вершины [импорт [данные для примеров
  VertexStyle -> White, VertexLabels -> Placed["Name", Center],
  [стиль вершины [белый [метки для вершин [расположен [центр
  VertexLabelStyle -> Directive[Blue, Italic, 20], GraphLayout -> "LayeredEmbedding"]
  [стиль меток вершин [директива [синий [курсив [укладка графа

```















```

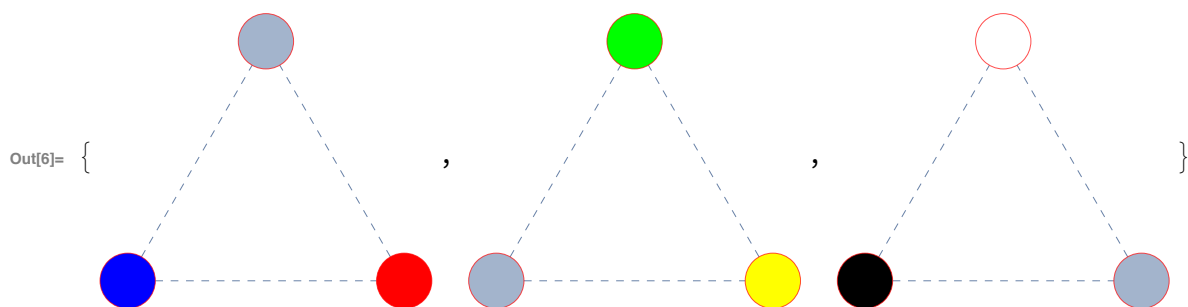
In[ ]:=










```

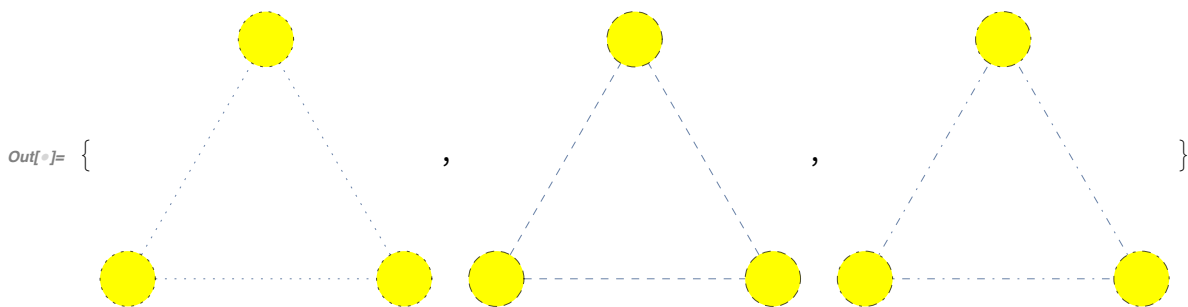
**Graph**: Options expected (instead of Null) beyond position 2 in  
 Graph[{1 → 2, 3 → 1, 3 → 4, 4 → 6, 5 → 3}, VertexSize → Large, Null, VertexStyle → , VertexLabels → Placed[Name, Center], VertexLabelStyle → Directive[, Italic, 20], GraphLayout → LayeredEmbedding]. An option must be a rule or a list of rules.

```
Out[ ]:= Graph[{1 → 2, 3 → 1, 3 → 4, 4 → 6, 5 → 3}, VertexSize → Large,
  Null, VertexStyle → , VertexLabels → Placed[Name, Center],
  VertexLabelStyle → Directive[, Italic, 20], GraphLayout → LayeredEmbedding]
```

```
In[6]:= Table[Graph[{1 → 2, 2 → 3, 3 → 1}], VertexStyle → style,
     
  VertexSize → 0.2, BaseStyle → EdgeForm[Red], EdgeStyle → Dashed],
       
  {style, {{1 → Blue, 2 → Red}, {2 → Yellow, 3 → Green}, {1 → Black, 3 → White}}}]
```



```
In[ ]:= Table[Graph[{1 → 2, 2 → 3, 3 → 1}], VertexStyle → Yellow,
     
  BaseStyle → EdgeForm[style], EdgeStyle → style, VertexSize → 0.2],
       
  {style, {Dotted, Dashed, DotDashed}}]
```

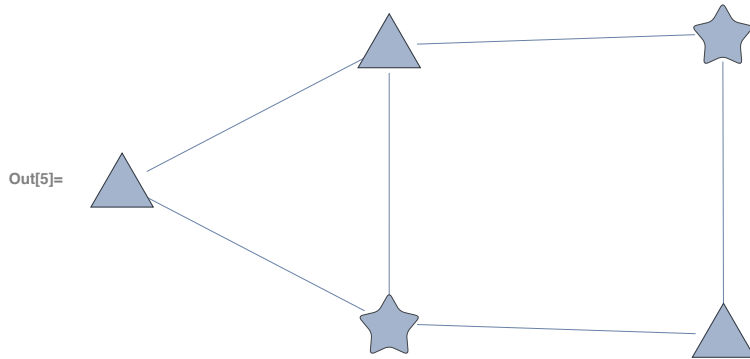


```
In[ ]:=
```

```

In[5]:= Graph[{1, 2, 3, 4, 5}, {1 ↔ 2, 1 ↔ 4, 2 ↔ 3, 2 ↔ 5, 3 ↔ 5, 3 ↔ 4},
  |граф
  VertexShapeFunction → {_?EvenQ →
  |функция формы вершины |чётное число?
    "ConcavePentagon", _?OddQ → "Triangle"}, VertexSize → 0.2]
  |нечётно... |треугольник |размер вершины

```



```

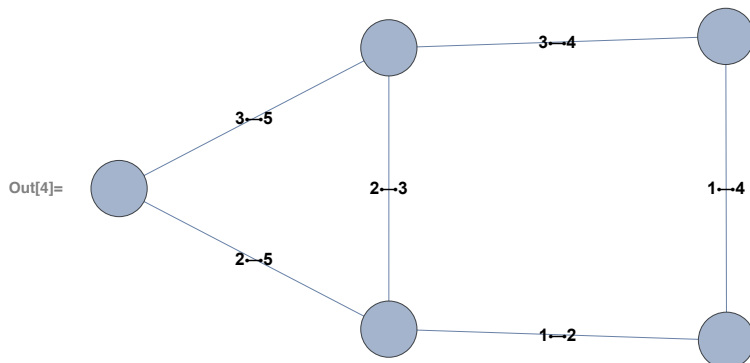
Out[5]= Graph[{1, 2, 3, 4, 5}, VertexShapeFunction →
  {EvenQ → ConcavePentagon, OddQ → Triangle}, VertexSize → 0.2]

```

```

In[4]:= Graph[{1, 2, 3, 4, 5},
  |граф
  {Labeled[1 ↔ 2, "1↔2"], Labeled[1 ↔ 4, "1↔4"], Labeled[2 ↔ 3, "2↔3"],
  |с пометкой |с пометкой |с пометкой
  Labeled[2 ↔ 5, "2↔5"], Labeled[3 ↔ 5, "3↔5"], Labeled[3 ↔ 4, "3↔4"]},
  |с пометкой |с пометкой |с пометкой
  VertexShapeFunction → "Circle", VertexSize → 0.2]
  |функция формы вершины |круг |размер вершины

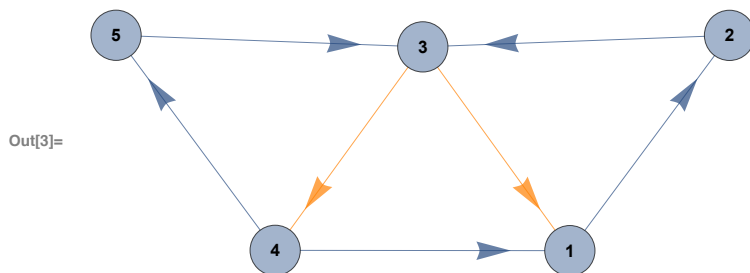
```



```

In[3]:= Graph[{1 → 2, 2 → 3, 3 → 1, 3 → 4, 4 → 1, 4 → 5, 5 → 3},
  |граф
  VertexSize → Medium, VertexLabels → Placed["Name", Center],
  |размер верш... |средний |метки для вершин |расположен |центр
  EdgeStyle → {3 ↔ 1 → Orange, 3 ↔ 4 → Orange},
  |стиль ребра |оранжевый |оранжевый
  EdgeShapeFunction → GraphElementData["Arrow", "ArrowSize" → 0.05]]
  |функция формы ребра |стрелка

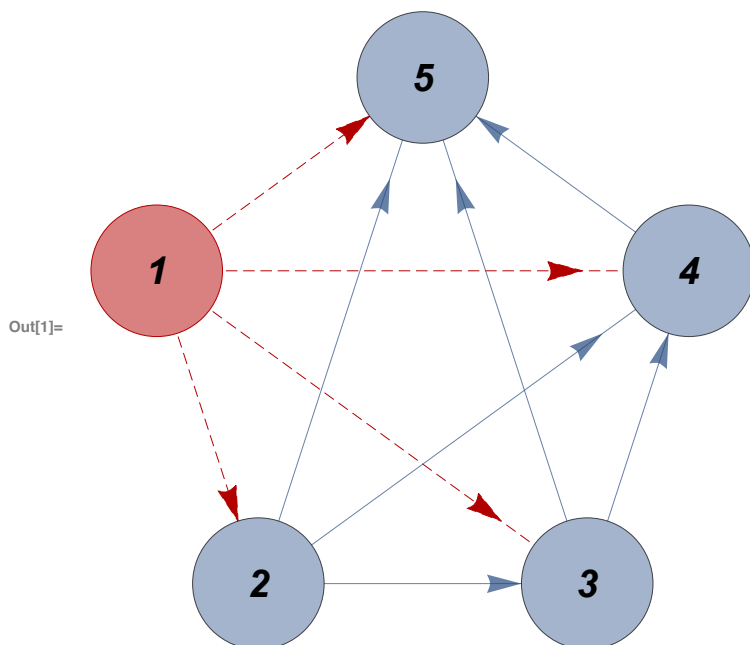
```



```

In[1]:= DirectedGraph[
  |ориентированный граф
  CompleteGraph[5, VertexSize → Large, VertexLabels → Placed["Name", Center],
  |полный граф |размер вершины |крупный |метки для вершин |расположен |центр
  VertexLabelStyle → Directive[Italic, 20], EdgeShapeFunction →
  |стиль меток вершин |директива |курсив |функция формы ребра
  GraphElementData["Arrow", "ArrowSize" → 0.05], GraphHighlightStyle → "Dashed",
  |стрелка |стиль выделенных элемент... |штриховой
  GraphHighlight → {1, 1 → 2, 1 → 3, 1 → 4, 1 → 5}], "Acyclic"]
  |выделить в графе

```



In[2]:=

```

HighlightGraph[DirectedGraph[
  _граф с подкраской _ориентированный граф
  CompleteGraph[5, VertexSize → Large, VertexLabels → Placed["Name", Center],
    _полный граф _размер вершины _крупный _метки для вершин _расположен _центр
    VertexLabelStyle → Directive[Italic, 20], EdgeShapeFunction →
    _стиль меток вершин _директива _курсив _функция формы ребра
    GraphElementData["Arrow", "ArrowSize" → 0.05]], "Acyclic"],
  _стрелка
  {1, 1 → 2, 1 → 3, 1 → 4, 1 → 5}, GraphHighlightStyle → "Dashed"]
  _стиль выделенных элемент... _штриховой пунктир

```

Out[2]=

