**class** Validator(object):  
  
 **def** \_\_init\_\_(self):  
 **pass  
  
 def** is\_integer(self, v):  
 *'''  
 Validates a vertex to be a positive integer* **:param** *v:* **:return***: raises an error  
 '''* err = **"An error occured! "  
 if** v == **""**:  
 err += **"The command cannot be empty! "  
 raise** ValueError(err)  
 **if** v.isdigit() == **False**:  
 err += **"Please insert a positive integer!!! "  
 raise** ValueError(err)  
  
 **def** valid\_edge(self, x, y, edges):  
 *'''  
 Validates an edge* **:param** *x: vertex1* **:param** *y: vertex2* **:param** *edges: edges dictionary* **:return***: raises an error  
 '''* er = **"An error occured! "  
 try**:  
 x = int(x)  
 y = int(y)  
 **if** x < 0 **or** y < 0:  
 er += **"Please insert some positive numbers!"  
 raise** ValueError(er)  
 **except** ValueError **as** error:  
 er += **"Please insert integer numbers! "  
 raise** ValueError(er)  
 **if** int(x) **not in** edges **or** int(y) **not in** edges:  
 er += **"Please insert some valid vertices,"** \  
 **"these do not exist in the graph! "  
 raise** ValueError(er)  
  
  
 **def** vertex\_valid(self, v, out):  
 *'''  
 Vlidates a vertex.* **:param** *v: inetegr* **:param** *out: outbound dictionary* **:return***: Raises an error.  
 '''* err = **"An error occured! "  
 try**:  
 v = int(v)  
 **if** v < 0 :  
 err += **"Please input a positive integer"  
 raise** ValueError(err)  
 **except** ValueError **as** ve:  
 err += **"Please input a positive integer "  
 raise** ValueError(err)  
 **if** int(v) **not in** out:  
 err += **"Please input a valid vertex!"** \  
 **"The vertex must be in the graph! "  
 raise** ValueError(err)  
  
  
 **def** validate\_cost(self, cost):  
 *'''  
 Validates a cost* **:param** *cost: ineteger* **:return***: Raises an error.  
 '''* err = **"An error occured! "  
 if** cost == **""**:  
 err += **"The command can not be empty!"  
 raise** ValueError(err)  
 **try**:  
 cost = int(cost)  
 **except** ValueError **as** ve:  
 err += **"Please input a number as a cost"  
 raise** ValueError(err)  
  
 **def** update\_edge(self, x, y, c, edges):  
 *'''  
 Validates when updating an edge.* **:param** *x: integer* **:param** *y: inetger* **:param** *c: integer* **:param** *edges: edges disctioanry* **:return***: Raises an error.  
 '''* l = **"An error occured! "  
 try**:  
 x = int(x)  
 y = int(y)  
 c = int(c)  
 **if** x < 0 **or** y < 0:  
 l += **"Please input positive numbers"  
 raise** ValueError(l)  
 **except** ValueError **as** error:  
 l += **"Please input integer numbers!"  
 if** (x, y) **not in** edges:  
 l += **"This is not a valid edge!"  
 raise** ValueError(l)  
  
  
 **def** valid\_add\_v(self, vertex, out):  
 *'''  
 Function to validate when a vertex is added.* **:param** *vertex: int* **:param** *out: outbound dictionary* **:return***: Raises an error.  
 '''* l = **"An error occured! "  
 try**:  
 vertex = int(vertex)  
 **if** vertex < 0:  
 l += **"Please input a positive number!"  
 raise** ValueError(l)  
 **except** ValueError **as** error:  
 l += **"Please input a positive integer number!"  
 raise** ValueError(l)  
 **if** int(vertex) **in** out:  
 l += **"This vertex already exists! "  
 raise** ValueError(l)  
  
 **def** add\_edge(self, x, y, edges):  
 *'''  
 Validates when adding an edge.* **:param** *x: integer* **:param** *y: integer* **:param** *edges: edges' dictionary* **:return***:  
 '''* errors = **"An error occured! "  
 if** (x, y) **in** edges:  
 errors += **"This edge already exists!"** \  
 **"Please input an other one! "  
 raise** ValueError(errors)  
  
 **def** remove\_edge(self, x, y, edges):  
 *'''  
 Validates when removing an edge.* **:param** *x: integer* **:param** *y: integer* **:param** *edges: edges' dictionary* **:return***: Raises an error.  
 '''* errors = **"An error occured! "  
 if** (x, y) **not in** edges:  
 errors += **"This edge do not exist!"** \  
 **"Please input an existing one! "  
 raise** ValueError(errors)