```
messages = [{''role'': ''system'', ''content'': ''You are an AI visual assistant in a 3D scene. The scene contains some objects, which compose a scene graph in json format. Each entity in the scene graph denotes an object instance, with a class label and an object id. The 'attributes' describes the attributes of the object itself, such as 'color', 'material', etc. The 'relations' describes the spatial relations with other objects.
```

For example, from the scene graph {'sofa-1': {'attributes': {'color': 'red'}, 'relations': ['to the right of chair-2', 'in front of table-3']}, 'chair-2': {'attributes': {'color': 'red'}, 'relations': ['to the right of chair-2', 'in front of table-3']}, 'chair-2': {'attributes': {'color': 'red'}, 'relations': ['to the right of chair-2', 'in front of table-3']}, 'chair-2': {'attributes': {'color': 'red'}, 'relations': ['to the right of chair-2', 'in front of table-3']}, 'chair-2': {'attributes': {'color': 'red'}, 'relations': ['to the right of chair-2', 'in front of table-3']}, 'chair-2': {'attributes': {'color': 'red'}, 'relations': ['to the right of chair-2', 'in front of table-3']}, 'chair-2': {'attributes': {'color': 'red'}, 'relations': ['to the right of chair-2', 'in front of table-3']}, 'chair-2': {'attributes': {'color': 'red'}, 'relations': ['to the right of chair-2', 'in front of table-3']}, 'chair-2': {'attributes': {'color': 'red'}, 'relations': ['to the right of chair-2', 'in front of table-3']}, 'chair-2': {'attributes': {'color': 'red'}, 'relations': ['to the right of chair-2', 'in front of table-3']}, 'chair-2': {'attributes': {'color': 'red'}, 'relations': ['to the right of chair-2', 'in front of table-3']}, 'chair-2': {'attributes': {'color': 'red'}, 'relations': ['to the right of chair-2', 'in front of table-3']}, 'chair-2': {'attributes': {'color': 'red'}, 'relations': ['to the right of chair-2', 'in front of table-3']}, 'chair-2': {'attributes': {'color': 'red'}, 'relations': ['to the right of chair-2', 'in front of table-3']}, 'chair-2': {'attributes': {'color': 'red'}, 'relations': ['to the right of chair-2', 'in front of table-3']}, 'chair-2': {'attributes': {'color': 'red'}, 'relations': ['to the right of chair-2', 'in front of table-3']}, 'chair-2': {'attributes': {'color': 'red'}, 'relations': {'color':

'brown'}, 'relations': ['to the left of sofa-1']}, 'table-3': { 'attributes': {'material': 'wood'}, 'relations': []}} we can know that 1) the sofa is red, 2) the chair is brown, 3) the football table is made of wood, 4) the chair is on the left of the sofa, 5) the chair is in front of the table.

All spatial positional relationships must be directly derivable from the 'relations', and any spatial relationship between objects with uncertainty cannot appear in the answer.

You need to generate 10-15 question-answer pairs based on the scene information. The question-answer pairs include the object types, counting the objects, object attributes, relative positions between objects. The questions should conform to the given scene information. The attributes of objects and spatial relations between objects can only be inferred from the 'attributes' and 'relations' in scene graph, respectively. The questions must be able to be answered correctly based on the scene graph. You need to provide the queried object. Note that all answers to the questions must be single words or phrases. The question answer pair should be following format:\nQ: <question>\nT: <queried object(s)>\nA: <Answer>. You can answer the question according to the queried object(s). If there is no information about the question, the <Answer> should be 'unkown'.' '}

for sample in few_shot_samples:
 messages.append({''role'': ''user'', ''content'': sample['content']})
 messages.append({''role'': ''assistant'', ''content'': sample['response']})
messages.append ({''role'': ''user'', ''content'': '\n'.join(sample['query'])})