1. Observe the following code and explain:

```
# 1.Load a variable with sentences
sentence = "Peter Piper picked a peck of pickled peppers A peck of pickled \
peppers Peter Piper picked If Peter Piper picked a peck of pickled \
peppers Wheres the peck of pickled peppers Peter Piper picked"

# 2.Initialize a dictionary object
word_dict = {}

# 3.Perform the word count
for word in sentence.split():
        if word not in word_dict:
                word_dict[word] =1
        else:
                word_dict[word]+=1

# 4.print the outputprint (word_dict)
```

   Alternative for a few lines:
   In step 2, we initialized the dictionary. Its empty when initialized. When a new key is added to a dictionary,
   accessing the dictionary through the new key will throw KeyError.
   In the preceding example in step 3, we included an if statement in the for loop to handle this situation.
   However, we can also use the following:
   word_dict.setdefault(word,0)
   With every key access to the dictionary, this statement has to be repeated if we are adding elements to a dictionary
   in a loop, as in a loop, we are not aware of new keys. Rewriting step 3 using setdefault will look as follows:

```
for word in sentence.split():
        word_dict.setdefault(word,0)
        word_dict[word]+=1
```

   try attempting the same using the class defaultdict from collections module.
2. Observe the following code:

```
For key, value in dictionary_name.items():
        print key,value
```

   the above code is used to get keys and values of a dictionary at a time. Consider a dictionary and write a piece of code to interchange keys and values of a dictionary.
3. Consider the following table.

User/Movie	LOR1	LOR2	LOR3	SW1	SW2
Alice	4	5	3	5	3
Huntsman	1	2	1	4	4
Snipe	3	4	4	2	1

In the first column, we have three users and the rest of the columns are movies.
The cell values are ratings given by a user for a movie. Let's say we want to represent this in memory
so that some other part of a larger code base can easily access this information.
How can you use a dictionary of dictionaries to achieve this objective?
Hint: use class defaultdict from collections module