1 (a)

$TF\text{-}IDF_{coffee\ doc1} = \log_{10}(25+1) \times \log_{10}(4/2)$
$= 0.4259$

$TF\text{-}IDF_{coffee\ doc2} = \log_{10}(18+1) \times \log_{10}(4/2)$
$= 0.3849$

$TF\text{-}IDF_{coffee\ doc3} = 0 \times \log_{10}(4/2)$
$= 0$

$TF\text{-}IDF_{coffee\ doc4} = 0 \times \log_{10}(4/2)$
$= 0$

1 b

$TF\text{-}IDF_{vacation\ doc1} = \log(20+1) \times \log(4/4)$
$= 0$

$TF\text{-}IDF_{vacation\ doc2} = \log(23+1) \times \log(4/4)$
$= 0$

$TF\text{-}IDF_{vacation\ doc3} = \log(18+1) \times \log(4/4)$
$= 0$

$TF\text{-}IDF_{vacation\ doc4} = \log(13+1) \times \log(4/4)$
$= 0$

~~TF-IDF indicates that~~

1C

TF IDF scores for coffee indicates that it is a discriminating term in doc1 and doc2 but not in DOC3 and DOC4. on the other hand, "vacation" is all zero in all document. This suggest that it is not relevant in any of the document.

[1 D]

TF IDF cup doc1 = $0 \times \log_{10}(4/3) = 0$

TF IDF cup doc2 = $\log_{10}(87+1) \times \log(4/3) = 0.2429$

TF IDF cup doc3 = $\log_{10}(78+1) \times \log(4/3) = 0.2371$

TF IDF cup doc4 = $\log(3+1) \times \log(4/3) = 0.0752$

TF IDF summer doc1 = $\log(10+1) \times \log(4/4) = 0$

TF IDF summer doc2 = $\log(12+1) \times \log(4/4) = 0$

TF IDF summer doc3 = $\log(15+1) \times \log(4/4) = 0$

TF IDF summer doc4 = $\log(80+1) \times \log(4/4) = 0$

TF IDF walk doc1 = $\log(3+1) \times \log(4/4) = 0$

TF IDF walk doc2 = $\log(6+1) \times \log(4/4) = 0$

TF IDF walk doc3 = $\log(32+1) \times \log(4/4) = 0$

TF IDF walk doc4 = $\log(20+1) \times \log(4/4) = 0$

	DOC1	DOC2	DOC3	DOC4
cup	0	0.2429	0.2371	0.0752
summer	0	0	0	0
coffee	0.4259	0.3849	0	0
walk	0	0	0	0
vacation	0	0	0	0

doc similarity = cos. similarity = $\dfrac{\vec{x} \cdot \vec{y}}{\|x\| \times \|y\|}$

cos[doc1, doc2] = $\dfrac{0.1639929}{0.4259 \times 0.455136}$ + ~~0.3849~~

$$\cos(doc1, doc2) = \frac{0.163929}{\sqrt{0.4239^2} \times \sqrt{0.2429^2 + 0.3849^2}} = 0.8456$$

$$\cos(doc1, doc3) = \frac{0}{\sqrt{0.4239^2} \times \sqrt{0.2371^2}} = 0$$

$\cos(DOC1, doc4) = 0$. Dot product is zero

$$\cos(doc2, doc3) = \frac{0.057592}{0.455136 \times 0.2371} = 0.533687$$

$$\cos(doc2\ doc4) = \frac{0.018266}{0.455136 \times 0.0752} = 0.533687$$

$$\cos(doc3, doc4) = \frac{0.01783}{0.2371 \times 0.0752} = 1$$

most similar
doc3 doc4
doc1 doc2
doc2 doc3
doc2 doc4
least similar
doc1 doc3
doc1 doc4

2a

	keyboard	webcam	sheep	less	racing	count(w)
Whiteboard	1434	512	8	42	3	1999
computer	5290	2351	19	9	42	7711
mouse	6102	3425	352	4251	532	14662
horse	1	53	531	2542	3513	6640
Vacation	14	53	34	5	9	115
count(context)	12841	6394	944	6849	4099	31127

2b

	keyboard	webcam	sheep	less	racing	P(w)
Whiteboard	0.0461	0.01644	0.0003	0.0013	0.0001	0.0642
computer	0.1699	0.0755	0.0006	0.0003	0.0013	0.2477
mouse	0.1960	0.1100	0.0113	0.1366	0.0171	0.4710
horse	0.0003	0.0017	0.01705	0.0817	0.1129	0.2133
Vacation	0.0004	0.0017	0.0011	0.0010	0.0002	0.0037
P(context	0.4125	0.2054	0.0303	0.2200	0.1316	1

2c

	keyboard	webcam	sheep	less	racing
whiteboard	0.7991	0.3181	0	0	0
computer	0.7336	0.5723	0	0	0
mouse	0.0126	0.18527	0	0.3986	0
horse	0	0	0	0.7875	1.988
Vacation	0	1.1643	3.2945	0.2969	0

2D

whiteboard = [0.7991, 0.3181, 0, 0, 0]
computer = [0.7336, 0.5723, 0, 0, 0]
mouse = [0.0126, 0.18527, 0, 0.3986, 0]
horse = [0, 0, 0, 0.7875, 1.988]
vacation = [0, 1.1643, 3.2945, 0.2969, 0]

~~whiteboard - mouse~~ =

cosine similarity = $\frac{\vec{x} \cdot \vec{y}}{\|x\| * \|y\|}$

whiteboard computer = $\frac{0.768268}{0.86086 \times 0.930428} = 0.960037$

whiteboard mouse = $\frac{0.069003}{0.860086 \times 0.439734} = 0.182447$

horse - vacation = $\frac{0.233809}{2.138294 \times 3.506775} = 0.031181$

most similar
whiteboard computer
least similar
whiteboard mouse
horse vacation.

(3)

```
[ ]  #question a
     def find_hypernym():
         house = wn.synset('house.n.01')
         hypernyms = house.hypernyms()

         if hypernyms:
             print("Hypernyms of 'house:")
             for hypernym in hypernyms:
                 # Split the hypernym name by periods and keep only the first part
                 hypernym_name = hypernym.name().split('.')[0]
                 print(hypernym_name)
         else:
             print("No hypernyms found for 'house.n.01'.")

     find_hypernym()
```

```
Hypernyms of 'house:
building
dwelling
```

```
    from nltk.corpus import wordnet as wn

    # Get the synsets for the words "mouse" and "horse"
    mouse_synsets = wn.synsets('mouse')
    horse_synsets = wn.synsets('horse')
    vacation_synsets = wn.synsets('vacation')

    # Get the first synset for each word
    mouse_synset = mouse_synsets[0]
    horse_synset = horse_synsets[0]
    vacation_synset = vacation_synsets[0]

    # Calculate the path similarity between the first senses of "mouse" and "horse"
    mouse_horse_similarity = mouse_synset.path_similarity(horse_synset)
    print("Path similarity between 'mouse' and 'horse':", mouse_horse_similarity)

    # Calculate the path similarity between the first senses of "horse" and "vacatio
    horse_vacation_similarity = horse_synset.path_similarity(vacation_synset)
    print("Path similarity between 'horse' and 'vacation':", horse_vacation_similari
```

```python
#question c
import numpy as np

# Load Glove word vectors
def load_glove_vectors(file_path):
    word_vectors = {}
    with open(file_path, 'r', encoding='utf-8') as f:
        for line in f:
            values = line.split()
            word = values[0]
            vector = np.array(values[1:], dtype='float32')
            word_vectors[word] = vector
    return word_vectors

# Compute cosine similarity between two words
def cosine_similarity(word1, word2, word_vectors):
    if word1 not in word_vectors or word2 not in word_vectors:
        return None

    vec1 = word_vectors[word1]
    vec2 = word_vectors[word2]

    dot_product = np.dot(vec1, vec2)
    norm1 = np.linalg.norm(vec1)
    norm2 = np.linalg.norm(vec2)

    similarity = dot_product / (norm1 * norm2)
    return similarity

# Example usage
glove_file_path = '/content/drive/MyDrive/nlp/hw5/glove.6B/glove.6B.50d.txt'
word_vectors = load_glove_vectors(glove_file_path)
```

```
[ ] #question d
    word1 = 'mouse'
    word2 = 'horse'
    similarity = cosine_similarity(word1, word2, word_vectors)
    print(f"Similarity between '{word1}' and '{word2}': {similarity:.4f}")
```

```
Similarity between 'mouse' and 'horse': 0.4356
```

```
    word1 = 'horse'
    word2 = 'vacation'
    similarity = cosine_similarity(word1, word2, word_vectors)
    print(f"Similarity between '{word1}' and '{word2}': {similarity:.4f}")
```

```
Similarity between 'horse' and 'vacation': 0.3102
```