AI Capstone Project 1: E-commerce

Project Task: Week 1

Class Imbalance Problem:

1. **Perform an EDA on the dataset.**

   a) See what a positive, negative, and neutral review looks like

   b) Check the class count for each class. It's a class imbalance problem.

2. **Convert the reviews in Tf-Idf score.**
3. **Run multinomial Naive Bayes classifier. Everything will be classified as positive because of the class imbalance.**

Project Task: Week 2

Tackling Class Imbalance Problem:

Oversampling or undersampling can be used to tackle the class imbalance problem.
In case of class imbalance criteria, use the following metrices for evaluating model performance: precision, recall, F1-score, AUC-ROC curve. Use F1-Score as the evaluation criteria for this project.
Use Tree-based classifiers like Random Forest and XGBoost.

Note: Tree-based classifiers work on two ideologies namely, Bagging or Boosting and have fine-tuning parameter which takes care of the imbalanced class.

Project Task: Week 3

Model Selection:

Apply multi-class SVM's and neural nets.
Use possible ensemble techniques like: XGboost + oversampled_multinomial_NB.
Assign a score to the sentence sentiment (engineer a feature called sentiment score). Use this engineered feature in the model and check for improvements. Draw insights on the same.

Project Task: Week 4

Applying LSTM:

Use LSTM for the previous problem (use parameters of LSTM like top-word, embedding-length, Dropout, epochs, number of layers, etc.)

Hint: Another variation of LSTM, GRU (Gated Recurrent Units) can be tried as well.

2. Compare the accuracy of neural nets with traditional ML based algorithms.

3. Find the best setting of LSTM (Neural Net) and GRU that can best classify the reviews as positive, negative, and neutral.

Hint: Use techniques like Grid Search, Cross-Validation and Random Search

Optional Tasks: Week 4

Topic Modeling:

1. **Cluster similar reviews. Note: Some reviews may talk about the device as a gift-option. Other reviews may be about product looks and some may**

   highlight about its battery and performance. Try naming the clusters.

2. **Perform Topic Modeling Hint: Use scikit-learn provided Latent Dirchlette Allocation (LDA) and Non-Negative Matrix Factorization (NMF).**

In [ ]:

```
import warnings
warnings.filterwarnings('ignore')
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import re
import string
import nltk
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.dummy import DummyClassifier
from sklearn.metrics import precision_score, recall_score, confusion_matrix
from sklearn.metrics import f1_score, roc_auc_score, roc_curve

print("Setup Complete")
```

```
Setup Complete
```

In [ ]:

```
%tensorflow_version 2.x
import tensorflow as tf
print(tf.__version__)
```

```
2.2.0
```

In [ ]:

```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from wordcloud import WordCloud

from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
```

```
from sklearn.naive_bayes import BernoulliNB, MultinomialNB
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn import metrics
from sklearn.metrics import roc_auc_score, accuracy_score
from sklearn.pipeline import Pipeline

from bs4 import BeautifulSoup
import re
import nltk
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from nltk.stem import SnowballStemmer, WordNetLemmatizer
from nltk import sent_tokenize, word_tokenize, pos_tag

import logging
from gensim.models import word2vec
from gensim.models import Word2Vec
from gensim.models.keyedvectors import KeyedVectors

from keras.preprocessing import sequence
from keras.utils import np_utils
from keras.models import Sequential
from keras.layers.core import Dense, Dropout, Activation, Lambda
from keras.layers.embeddings import Embedding
from keras.layers.recurrent import LSTM, SimpleRNN, GRU
from keras.preprocessing.text import Tokenizer
from collections import defaultdict
from keras.layers.convolutional import Convolution1D
from keras import backend as K
from keras.layers.embeddings import Embedding
from keras.callbacks import EarlyStopping

print("Setup Complete")
```

```
Setup Complete
```

```
Using TensorFlow backend.
```

In [ ]:

```
from google.colab import files

uploaded = files.upload()

for fn in uploaded.keys():
  print('User uploaded file "{name}" with length {length} bytes'.format(
      name=fn, length=len(uploaded[fn])))
```

Choose File **No file selected**

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving test_data.csv to test_data.csv
Saving test_data_hidden.csv to test_data_hidden.csv
Saving train_data.csv to train_data.csv
User uploaded file "test_data.csv" with length 519747 bytes
User uploaded file "test_data_hidden.csv" with length 528718 bytes
User uploaded file "train_data.csv" with length 2074490 bytes
```

In [ ]:

```
# Load the data
import pandas as pd
data = pd.read_csv('train_data.csv')
test = pd.read_csv('test_data.csv')
test_prediction = pd.read_csv('test_data_hidden.csv')
```

In [ ]:

```
data.head()
```

Out[ ]:

	name	brand	categories	primaryCategories	reviews.date	reviews.text	revi
0	All-New Fire HD 8 Tablet, 8" HD Display, Wi-Fi...	Amazon	Electronics,iPad & Tablets,All Tablets,Fire Ta...	Electronics	2016-12-26T00:00:00.000Z	Purchased on Black FridayPros - Great Price (e...	Powerf
1	Amazon - Echo Plus w/ Built-In Hub - Silver	Amazon	Amazon Echo,Smart Home,Networking,Home & Tools...	Electronics,Hardware	2018-01-17T00:00:00.000Z	I purchased two Amazon in Echo Plus and two do...	Amazon E AW
2	Amazon Echo Show Alexa-enabled Bluetooth Speak...	Amazon	Amazon Echo,Virtual Assistant Speakers,Electro...	Electronics,Hardware	2017-12-20T00:00:00.000Z	Just an average Alexa option. Does show a few ...	
3	Fire HD 10 Tablet, 10.1 HD Display, Wi-Fi, 16 ...	Amazon	eBook Readers,Fire Tablets,Electronics Feature...	Office Supplies,Electronics	2017-08-04T00:00:00.000Z	very good product. Exactly what I wanted, and ...	G
4	Brand New Amazon Kindle Fire 16gb 7" Ips Displ...	Amazon	Computers/Tablets & Networking,Tablets & eBook...	Electronics	2017-01-23T00:00:00.000Z	This is the 3rd one I've purchased. I've bough...	Very

In [ ]:

```
Positive = data[data['sentiment']== "Positive"].iloc[:,[5,6,7]]
Neutral = data[data['sentiment']== "Neutral"].iloc[:,[5,6,7]]
Negative = data[data['sentiment']== "Negative"].iloc[:,[5,6,7]]
```

In [ ]:

```
Positive['sentiment'].value_counts()
```

Out[ ]:

```
Positive    3749
Name: sentiment, dtype: int64
```

In [ ]:

```
Neutral['sentiment'].value_counts()
```

Out[ ]:

```
Neutral    158
Name: sentiment, dtype: int64
```

In [ ]:

```
Negative['sentiment'].value_counts()
```

Out[ ]:

```
Negative    93
Name: sentiment, dtype: int64
```

Converting the Reviews as Tf-Idf Score

In [ ]:

```
# Keeping only those Features that we need for further exploring.
data1 = data [["sentiment","reviews.text"]]
```

In [ ]:

```
data1.head()
```

Out[ ]:

	sentiment	reviews.text
0	Positive	Purchased on Black FridayPros - Great Price (e...
1	Positive	I purchased two Amazon in Echo Plus and two do...
2	Neutral	Just an average Alexa option. Does show a few ...
3	Positive	very good product. Exactly what I wanted, and ...
4	Positive	This is the 3rd one I've purchased. I've bough...

In [ ]:

```
# Resetting the Index.
data1.index = pd.Series(list(range(data1.shape[0])))
```

In [ ]:

```
print('Shape : ',data1.shape)
data1.head()
```

```
Shape :  (4000, 2)
```

Out[ ]:

	sentiment	reviews.text
0	Positive	Purchased on Black FridayPros - Great Price (e...
1	Positive	I purchased two Amazon in Echo Plus and two do...
2	Neutral	Just an average Alexa option. Does show a few ...
3	Positive	very good product. Exactly what I wanted, and ...
4	Positive	This is the 3rd one I've purchased. I've bough...

Now we Create Preprocessing Function & Applying it on the Data

In [ ]:

```
from nltk.tokenize import RegexpTokenizer
from nltk.corpus import stopwords
import nltk
from nltk.corpus import wordnet
from nltk.stem import WordNetLemmatizer
nltk.download('wordnet')
#Download Stopwords
nltk.download('stopwords')

wordnet_lemmatizer = WordNetLemmatizer()
tokenizer = RegexpTokenizer(r'[a-z]+')
stop_words = set(stopwords.words('english'))

def preprocess(document):
    document = document.lower() # Convert to lowercase
    words = tokenizer.tokenize(document) # Tokenize
    words = [w for w in words if not w in stop_words] # Removing stopwords
    # Lemmatizing
    for pos in [wordnet.NOUN, wordnet.VERB, wordnet.ADJ, wordnet.ADV]:
        words = [wordnet_lemmatizer.lemmatize(x, pos) for x in words]
    return " ".join(words)

print("Setup Complete")
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Unzipping corpora/wordnet.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
Setup Complete
```

In [ ]:

```
data1['Processed_Review'] = data1['reviews.text'].apply(preprocess)

data1.head()
```

Out[ ]:

	sentiment	reviews.text	Processed_Review
0	Positive	Purchased on Black FridayPros - Great Price (e...	purchase black fridaypros great price even sal...
1	Positive	I purchased two Amazon in Echo Plus and two do...	purchase two amazon echo plus two dot plus fou...
2	Neutral	Just an average Alexa option. Does show a few ...	average alexa option show thing screen still l...
3	Positive	very good product. Exactly what I wanted, and ...	good product exactly want good price
4	Positive	This is the 3rd one I've purchased. I've bough...	rd one purchase buy one niece case compare one...

In [ ]:

```
data2 = data1 [["sentiment","Processed_Review"]]
data2.head()
```

Out[ ]:

	sentiment	Processed_Review
0	Positive	purchase black fridaypros great price even sal...
1	Positive	purchase two amazon echo plus two dot plus fou...
2	Neutral	average alexa option show thing screen still l...
3	Positive	good product exactly want good price
4	Positive	rd one purchase buy one niece case compare one...

Creating TF-IDF Matrix & Multinomial Naive Bayes Classifier

In [ ]:

```
def textPreprocessing(data2):
    #Remove Punctuation Logic
    import string
    removePunctuation = [char for char in data2 if char not in string.punctuation]
    #Join Chars to form sentences
    sentenceWithoutPunctuations = ''.join(removePunctuation)
    words = sentenceWithoutPunctuations.split()
    #StopwordRemoval
    from nltk.corpus import stopwords
    removeStopwords = [word for word in words if word.lower() not in stopwords.words('english')]

    return removeStopwords
```

In [ ]:

```
data2.groupby('sentiment').describe()
```

Out[ ]:

	Processed_Review			
	count	unique	top	freq
sentiment				
Negative	93	78	proprietary apps daughter like could install b...	3
Neutral	158	145	average alexa option show thing screen still l...	2
Positive	3749	3372	give grandkids age christmas love	4

In [ ]:

```
#Text preprocessing
data2['Processed_Review'].head(2).apply(textPreprocessing)
```

Out[ ]:

```
0    [purchase, black, fridaypros, great, price, ev...
1    [purchase, two, amazon, echo, plus, two, dot, ...
Name: Processed_Review, dtype: object
```

Sklearn Package CountVectorizer (For creating Bag of Words)

In [ ]:

```
from sklearn.feature_extraction.text import CountVectorizer
bow = CountVectorizer(analyzer=textPreprocessing).fit(data2['Processed_Review'])
```

In [ ]:

```
len(bow.vocabulary_)
```

Out[ ]:

```
3407
```

In [ ]:

```
reviews_bow = bow.transform(data2['Processed_Review'])
```

Tf-Idf

In [ ]:

```
from sklearn.feature_extraction.text import TfidfTransformer
tfidfData = TfidfTransformer().fit(reviews_bow)
tfidfDataFinal = tfidfData.transform(reviews_bow)
```

In [ ]:

```
tfidfDataFinal.shape
```

Out[ ]:

```
(4000, 3407)
```

Now the data is all set for Model Building!

- **Model Training - NaiveBayes Algorithm**
- **String Data Handling - MultinomialNB**

In [ ]:

```
from sklearn.naive_bayes import MultinomialNB
model = MultinomialNB().fit(tfidfDataFinal,data2['sentiment'])
```

In [ ]:

```
model
```

Out[ ]:

```
MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)
```

In [ ]:

In [ ]:

```
inputData = "very bad dont like it at all it sucks"
l1 = textPreprocessing(inputData)
l2 = bow.transform(l1)
l3 = tfidfData.transform(l2)
prediction = model.predict(l3[0])
prediction
```

Out[ ]:

```
array(['Positive'], dtype='<U8')
```

Insight

After running Multinomial Naive Bayes Classifier Everything is classified as positive because of the class imbalance as seen above.

Tackling Class Imbalance Problem:

In [ ]:

```
#Creating independent and Dependent Features
columns = data2.columns.tolist()
# Filtering the columns to remove data we do not want
columns = [c for c in columns if c not in ["sentiment"]]
# Store the variable we are predicting
target = "sentiment"
# Defining a random state
state = np.random.RandomState(42)
X = data2[columns]
Y = data2[target]
# Printing the shapes of X & Y
print(X.shape)
print(Y.shape)
```

```
(4000, 1)
(4000,)
```

In [ ]:

```
print(data2.sentiment.value_counts())
```

```
Positive    3749
Neutral      158
Negative      93
Name: sentiment, dtype: int64
```

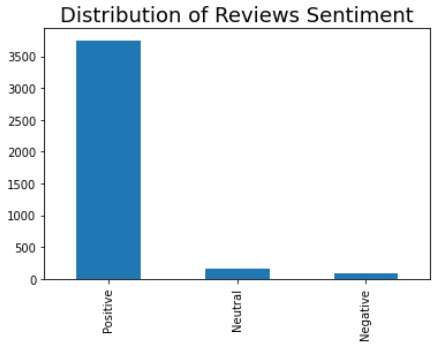
In [ ]:

```
# Using Matplotlib to show distribution of reviews sentiment in the dataset
print(data1.sentiment.value_counts())
data1['sentiment'].value_counts().plot(kind='bar')
plt.title("Distribution of Reviews Sentiment", size=18)
```

```
Positive    3749
Neutral      158
Negative      93
Name: sentiment, dtype: int64
```

Out[ ]:

```
Text(0.5, 1.0, 'Distribution of Reviews Sentiment')
```



In [ ]:

```
print(Positive.shape,Neutral.shape,Negative.shape)
```

```
(3749, 3) (158, 3) (93, 3)
```

Applying Over Sampling

In [ ]:

```
# RandomOverSampler to handle imbalanced data
from imblearn.over_sampling import RandomOverSampler
ros = RandomOverSampler(random_state=0)
X_res,Y_res=ros.fit_sample(X,Y)
```

In [ ]:

```
from collections import Counter
print(sorted(Counter(Y_res).items()))
```

```
[('Negative', 3749), ('Neutral', 3749), ('Positive', 3749)]
```

In [ ]:

```
X_res.shape,Y_res.shape
```

Out[ ]:

```
((11247, 1), (11247,))
```

In [ ]:

```
#Checking out both old & new data
```

```
print('Original dataset shape {}'.format(Counter(Y)))
print('Resampled dataset shape {}'.format(Counter(Y_res)))
```

```
Original dataset shape Counter({'Positive': 3749, 'Neutral': 158, 'Negative': 93})
Resampled dataset shape Counter({'Positive': 3749, 'Neutral': 3749, 'Negative': 3749})
```

In [ ]:

```
#Creating X output to dataframe
X1=pd.DataFrame(X_res,columns=['Processed_Review'])
```

In [ ]:

```
#Creating Y output to dataframe for merging
Y1=pd.DataFrame(Y_res,columns=['sentiment'])
```

In [ ]:

```
#Merging the X & Y output to Final data
Final_data=pd.concat([X1,Y1],axis=1)
Final_data.head()
```

Out[ ]:

	Processed_Review	sentiment
0	purchase black fridaypros great price even sal...	Positive
1	purchase two amazon echo plus two dot plus fou...	Positive
2	average alexa option show thing screen still l...	Neutral
3	good product exactly want good price	Positive
4	rd one purchase buy one niece case compare one...	Positive

In [ ]:

```
Final_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11247 entries, 0 to 11246
Data columns (total 2 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   Processed_Review  11247 non-null  object
 1   sentiment         11247 non-null  object
dtypes: object(2)
memory usage: 175.9+ KB
```

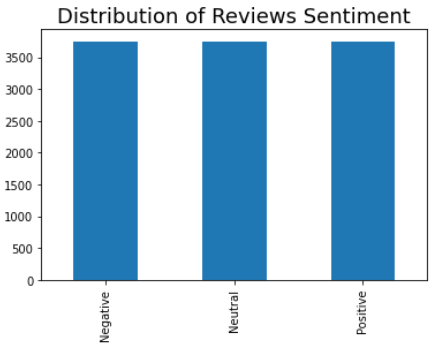
In [ ]:

```
# Using Matplotlib to plot the final data & show distribution of reviews sentiment in the dataset
print(Final_data.sentiment.value_counts())
Final_data['sentiment'].value_counts().plot(kind='bar')
plt.title("Distribution of Reviews Sentiment", size=18)
```

```
Negative    3749
Neutral     3749
Positive    3749
Name: sentiment, dtype: int64
```

Out[ ]:

```
Text(0.5, 1.0, 'Distribution of Reviews Sentiment')
```



In [ ]:

```
df = Final_data.sample(frac=0.1, random_state=0)
# Dropping missing values
df.dropna(inplace=True)
df.head()
```

Out[ ]:

	Processed_Review	sentiment
8805	buy think would great read book play game howe...	Neutral
9736	good tablet kid lot appts download game	Neutral
125	item work expect great product	Positive
10143	great beginner like child limit use many apps ...	Neutral
10937	buy kindle past time one come defective port b...	Neutral

Train & Test Split Data

In [ ]:

```
# Splitting data into training set and validation
X_train, X_test, y_train, y_test = train_test_split(df['Processed_Review'], df['sentiment'], \
                                                    test_size=0.1, random_state=0)

print('Load %d training examples and %d validation examples. \n' %(X_train.shape[0],X_test.shape[0]))
print('Show a review in the training set : \n', X_train.iloc[10])
```

```
Load 1012 training examples and 113 validation examples.

Show a review in the training set :
 daughter love easy navigate hard break
```

Bag of Words

- **The aim of this project is to classify the reviews into positive, neutral and negative sentiment.**

  There are two main steps which are involved:

  1. **We need to find word embedding to convert text into a numerical representation.**
  2. **We fit the numerical representations of text to machine learning algorithms or deep learning architectures.**

- **One common approach of word embedding is frequency based embedding viz Bag of Words (BoW) model. BoW model learns a vocabulary list from a given corpus and represents each document based on some counting methods of words. Here, we will explore the model performance using BoW with supervised learning algorithms.**

  Here's the workflow of this part.

  1. **Preprocess raw reviews to cleaned reviews.**
  2. **Create BoW using CountVectorizer/Tfidfvectorizer in Sklearn.**
  3. **Transform review text to numerical representations (feature vectors).**
  4. **Fit feature vectors to supervised learning algorithm (Naive Bayes, Logistic regression, etc.).**
  5. **Improve the model performance by GridSearch.**

Text Preprocessing

The following text preprocessing methods are implemented to convert raw reviews to cleaned review making it easier to do feature extraction in the next step.

1. **Remove non-character such as digits and symbols.**
2. **Convert to lower case.**
3. **Remove stop words such as "the" and "and" if needed.**
4. **Convert to root words by stemming if needed.**

In [ ]:

```
def cleanText(raw_text, remove_stopwords=False, stemming=False, split_text=False, \
              ):
    '''
    Convert a raw review to a cleaned review
    '''
    text = BeautifulSoup(raw_text, 'lxml').get_text()  #remove html
    letters_only = re.sub("[^a-zA-Z]", " ", text)  # remove non-character
    words = letters_only.lower().split() # convert to lower case

    if remove_stopwords: # remove stopword
        stops = set(stopwords.words("english"))
        words = [w for w in words if not w in stops]

    if stemming==True: # stemming
#         stemmer = PorterStemmer()
        stemmer = SnowballStemmer('english')
        words = [stemmer.stem(w) for w in words]

    if split_text==True:  # split text
        return (words)

    return( " ".join(words))
```

In [ ]:

```
# Preprocess text data in training set and validation set
X_train_cleaned = []
X_test_cleaned = []

for d in X_train:
    X_train_cleaned.append(cleanText(d))
print('Show a cleaned review in the training set : \n',  X_train_cleaned[10])

for d in X_test:
    X_test_cleaned.append(cleanText(d))
```

```
Show a cleaned review in the training set :
 daughter love easy navigate hard break
```

In [ ]:

```
# Preprocess text data in training set and validation set
X_train_cleaned = []
X_test_cleaned = []

for d in X_train:
    X_train_cleaned.append(cleanText(d))
print('Show a cleaned review in the training set : \n',  X_train_cleaned[10])

for d in X_test:
    X_test_cleaned.append(cleanText(d))
```

```
Show a cleaned review in the training set :
 daughter love easy navigate hard break
```

CountVectorizer with Multinomial Naive Bayes (Benchmark Model)

- **Now as we have cleaned all reviews the next step is converting the reviews into numerical representations for a machine learning algorithm.**
- **We will use CountVectorizer which implements both tokenization and occurrence counting in a single class provided by the Sklearn library. The output is a sparse matrix representation of the document.**

In [ ]:

```
# Fit and transform the training data to a document-term matrix using CountVectorizer
countVect = CountVectorizer()
X_train_countVect = countVect.fit_transform(X_train_cleaned)
print("Number of features : %d \n" %len(countVect.get_feature_names())) #6378
print("Show some feature names : \n", countVect.get_feature_names()[::1000])


# Train MultinomialNB classifier
mnb = MultinomialNB()
mnb.fit(X_train_countVect, y_train)
```

```
Number of features : 1511

Show some feature names :
 ['ability', 'playtime']
```

Out[ ]:

```
MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)
```

In [ ]:

```
def modelEvaluation(predictions):
    '''
    Print model evaluation to predicted result
    '''
    print ("\nAccuracy on validation set: {:.4f}".format(accuracy_score(y_test, predictions)))
    #print("\nAUC score : {:.4f}".format(roc_auc_score(y_test, predictions)))
    print("\nClassification report : \n", metrics.classification_report(y_test, predictions))
    print("\nConfusion Matrix : \n", metrics.confusion_matrix(y_test, predictions))
```

In [ ]:

```
# Evaluate the model on validaton set
predictions = mnb.predict(countVect.transform(X_test_cleaned))
modelEvaluation(predictions)
```

```
Accuracy on validation set: 0.8938

Classification report :
              precision    recall  f1-score   support

    Negative       0.93      0.95      0.94        39
     Neutral       0.85      0.90      0.88        39
    Positive       0.91      0.83      0.87        35

    accuracy                           0.89       113
   macro avg       0.89      0.89      0.89       113
weighted avg       0.89      0.89      0.89       113


Confusion Matrix :
 [[37  0  2]
 [ 3 35  1]
 [ 0  6 29]]
```

TfidfVectorizer with Logistic Regression

- **Some words might appear quite frequently but have a very less or negligible meaningful information about the sentiment for a particular review. Instead of using occurance counting we will use tf-idf transform to scale down the impact of frequently appearing words in a given corpus.**
- **In sklearn library we will use TfidfVectorizer which implements both tokenization and tf-idf weighted counting in a single class.**

In [ ]:

```
# Fitting and transforming the training data to a document-term matrix using TfidfVectorizer
tfidf = TfidfVectorizer(min_df=5) #minimum document frequency of 5
X_train_tfidf = tfidf.fit_transform(X_train)
print("Number of features : %d \n" %len(tfidf.get_feature_names())) #1722
print("Show some feature names : \n", tfidf.get_feature_names()[::1000])

# Logistic Regression
lr = LogisticRegression()
lr.fit(X_train_tfidf, y_train)
```

```
Number of features : 691

Show some feature names :
 ['able']
```

Out[ ]:

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

In [ ]:

```
# Have a look at the top 10 features with the smallest and largest coefficients
feature_names = np.array(tfidf.get_feature_names())
sorted_coef_index = lr.coef_[0].argsort()
print('\nTop 10 features with the smallest coefficients :\n{}\n'.format(feature_names[sorted_coef_index[:10]]))
print('Top 10 features with the largest coefficients : \n{}'.format(feature_names[sorted_coef_index[:-11:-1]]))
```

```
Top 10 features with the smallest coefficients :
['love' 'easy' 'great' 'play' 'read' 'alexa' 'kid' 'price' 'well' 'enjoy']

Top 10 features with the largest coefficients :
['return' 'update' 'bad' 'know' 'terrible' 'th' 'poor' 'try' 'minute'
 'youtube']
```

In [ ]:

```
# Evaluating on the validaton set
predictions = lr.predict(tfidf.transform(X_test_cleaned))
modelEvaluation(predictions)
```

```
Accuracy on validation set: 0.9292

Classification report :
              precision    recall  f1-score   support

    Negative       0.93      1.00      0.96        39
     Neutral       0.88      0.92      0.90        39
    Positive       1.00      0.86      0.92        35

    accuracy                           0.93       113
   macro avg       0.94      0.93      0.93       113
weighted avg       0.93      0.93      0.93       113


Confusion Matrix :
 [[39  0  0]
 [ 3 36  0]
```

```
[ 0  5 30]]
```

TfidfVectorizer with Linear SVM by using SGD

Some words might appear quite frequently but have a very less or negligible meaningful information about the sentiment for a particular review. Instead of using occurance counting we will use tf-idf transform to scale down the impact of frequently appearing words in a given corpus.

In sklearn library we will use TfidfVectorizer which implements both tokenization and tf-idf weighted counting in a single class.

In [ ]:

```
# Fitting and transforming the training data to a document-term matrix using TfidfVectorizer
tfidf = TfidfVectorizer(min_df=5) #minimum document frequency of 5
X_train_tfidf = tfidf.fit_transform(X_train)
print("Number of features : %d \n" %len(tfidf.get_feature_names())) #1722
print("Show some feature names : \n", tfidf.get_feature_names()[::1000])

# SVM
from sklearn.linear_model import SGDClassifier
clf = SGDClassifier(loss="hinge", penalty="l2")
clf.fit(X_train_tfidf, y_train)
```

```
Number of features : 691

Show some feature names :
 ['able']
```

Out[ ]:

```
SGDClassifier(alpha=0.0001, average=False, class_weight=None,
              early_stopping=False, epsilon=0.1, eta0=0.0, fit_intercept=True,
              l1_ratio=0.15, learning_rate='optimal', loss='hinge',
              max_iter=1000, n_iter_no_change=5, n_jobs=None, penalty='l2',
              power_t=0.5, random_state=None, shuffle=True, tol=0.001,
              validation_fraction=0.1, verbose=0, warm_start=False)
```

In [ ]:

```
# Have a look at the top 10 features with the smallest and largest coefficients
feature_names = np.array(tfidf.get_feature_names())
sorted_coef_index = clf.coef_[0].argsort()
print('\nTop 10 features with the smallest coefficients :\n{}\n'.format(feature_names[sorted_coef_index[:10]]))
print('Top 10 features with the largest coefficients : \n{}'.format(feature_names[sorted_coef_index[:-11:-1]]))
```

```
Top 10 features with the smallest coefficients :
['overall' 'play' 'well' 'sometimes' 'love' 'easy' 'great' 'age'
 'definitely' 'control']

Top 10 features with the largest coefficients :
['terrible' 'return' 'exchange' 'poor' 'minute' 'update' 'th' 'question'
 'never' 'customer']
```

In [ ]:

```
# Evaluating on the validaton set
predictions = clf.predict(tfidf.transform(X_test_cleaned))
modelEvaluation(predictions)
```

```
Accuracy on validation set: 0.9204

Classification report :
              precision    recall  f1-score   support

    Negative       0.95      1.00      0.97        39
     Neutral       0.84      0.97      0.90        39
    Positive       1.00      0.77      0.87        35

    accuracy                           0.92       113
   macro avg       0.93      0.92      0.92       113
weighted avg       0.93      0.92      0.92       113


Confusion Matrix :
 [[39  0  0]
 [ 1 38  0]
 [ 1  7 27]]
```

Using XGBoost Classifier

Some words might appear quite frequently but have a very less or negligible meaningful information about the sentiment for a particular review. Instead of using occurance counting we will use tf-idf transform to scale down the impact of frequently appearing words in a given corpus.

In sklearn library we will use TfidfVectorizer which implements both tokenization and tf-idf weighted counting in a single class.

In [ ]:

```
# XGBoost
from xgboost import XGBClassifier
```

In [ ]:

```
# Fitting and transforming the training data to a document-term matrix using TfidfVectorizer
tfidf = TfidfVectorizer(min_df=5) #minimum document frequency of 5
X_train_tfidf = tfidf.fit_transform(X_train)
print("Number of features : %d \n" %len(tfidf.get_feature_names())) #1722
print("Show some feature names : \n", tfidf.get_feature_names()[::1000])

# XGBoost Classifier
xgb = XGBClassifier()
xgb.fit(X_train_tfidf, y_train)
```

```
Number of features : 691

Show some feature names :
 ['able']
```

Out[ ]:

```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0,
              learning_rate=0.1, max_delta_step=0, max_depth=3,
              min_child_weight=1, missing=None, n_estimators=100, n_jobs=1,
              nthread=None, objective='multi:softprob', random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
              silent=None, subsample=1, verbosity=1)
```

In [ ]:

```
# Look at the top 10 features with smallest and the largest coefficients
feature_names = np.array(tfidf.get_feature_names())
# sorted_coef_index = xgb.coef_[0].argsort()
print('\nTop 10 features with smallest coefficients :\n{}\n'.format(feature_names[sorted_coef_index[:10]]))
print('Top 10 features with largest coefficients : \n{}'.format(feature_names[sorted_coef_index[:-11:-1]]))
```

```
Top 10 features with smallest coefficients :
['overall' 'play' 'well' 'sometimes' 'love' 'easy' 'great' 'age'
 'definitely' 'control']

Top 10 features with largest coefficients :
['terrible' 'return' 'exchange' 'poor' 'minute' 'update' 'th' 'question'
 'never' 'customer']
```

In [ ]:

```
# Evaluating on the validaton set
predictions = xgb.predict(tfidf.transform(X_test_cleaned))
modelEvaluation(predictions)
```

```
Accuracy on validation set: 0.9027

Classification report :
              precision    recall  f1-score   support

    Negative       0.90      0.97      0.94        39
     Neutral       0.86      0.95      0.90        39
    Positive       0.96      0.77      0.86        35

    accuracy                           0.90       113
   macro avg       0.91      0.90      0.90       113
weighted avg       0.91      0.90      0.90       113


Confusion Matrix :
 [[38  1  0]
 [ 1 37  1]
 [ 3  5 27]]
```

Pipeline and GridSearch

We will build a pipeline in the Sklearn library to streamline the workflow and use GridSearch on the pipeline model to implement hyperparameter tuning for both the vectorizer and classifier at once.

In [ ]:

```
# Building a pipeline
estimators = [("tfidf", TfidfVectorizer()), ("lr", LogisticRegression())]
model = Pipeline(estimators)


# Grid search
params = {"lr__C":[0.1, 1, 10], #regularization param of logistic regression
          "tfidf__min_df": [1, 3], #min count of words
          "tfidf__max_features": [1000, None], #max features
          "tfidf__ngram_range": [(1,1), (1,2)], #1-grams or 2-grams
          "tfidf__stop_words": [None, "english"]} #use stopwords or don't

grid = GridSearchCV(estimator=model, param_grid=params, scoring="accuracy", n_jobs=-1)
grid.fit(X_train_cleaned, y_train)
print("The best paramenter set is : \n", grid.best_params_)


# Evaluate on the validaton set
predictions = grid.predict(X_test_cleaned)
modelEvaluation(predictions)
```

```
The best paramenter set is :
 {'lr__C': 10, 'tfidf__max_features': None, 'tfidf__min_df': 1, 'tfidf__ngram_range': (1, 2), 'tfidf__stop_words': None}

Accuracy on validation set: 0.9381

Classification report :
              precision    recall  f1-score   support

    Negative       0.97      0.97      0.97        39
     Neutral       0.90      0.95      0.92        39
    Positive       0.94      0.89      0.91        35

    accuracy                           0.94       113
   macro avg       0.94      0.94      0.94       113
weighted avg       0.94      0.94      0.94       113


Confusion Matrix :
 [[38  0  1]
 [ 1 37  1]
 [ 0  4 31]]
```

Word2Vec

Another common approach of word embedding is the prediction based embedding like Word2Vec model. Briefly, Word2Vec is a combination of two techniques: Continuous Bag of Words (CBoW) and Skip-Gram model. Both are Shallow Neural Networks which learn weights for the word vector representations.

Here, we will train Word2Vec model to create our own word vector representation using gensim library. Then we will fit the feature vectors of the reviews to the Random Forest Classifier. Here's the workflow of this part: -

1. **Parse review text to sentences (Word2Vec model takes a list of sentences as inputs).**
2. **Create vocabulary list using Word2Vec model.**
3. **Transform each review into numerical representation by computing average feature vectors of words therein**
4. **Fit the average feature vectors to Random Forest Classifier**

Parsing Review into Sentences

Word2Vec model takes a list of sentences as inputs and outputs word vector representations for words in the vocabulary list created. Before we train the Word2Vec model, we have to parse reviews in the training set into sentences.

In [ ]:

```
import nltk
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
```

Out[ ]:

```
True
```

In [ ]:

```
# Splitting review text into parsed sentences using NLTK's punkt tokenizer

tokenizer = nltk.data.load('tokenizers/punkt/english.pickle')

def parseSent(review, tokenizer, remove_stopwords=False):
    '''
    Parse text into sentences
    '''
    raw_sentences = tokenizer.tokenize(review.strip())
    sentences = []
    for raw_sentence in raw_sentences:
        if len(raw_sentence) > 0:
            sentences.append(cleanText(raw_sentence, remove_stopwords, split_text=True))
    return sentences


# Parsing each review in the training set into sentences
sentences = []
for review in X_train_cleaned:
    sentences += parseSent(review, tokenizer)

print('%d parsed sentence in the training set\n'  %len(sentences))
print('Show a parsed sentence in the training set : \n',  sentences[10])
```

```
1012 parsed sentence in the training set

Show a parsed sentence in the training set :
 ['daughter', 'love', 'easy', 'navigate', 'hard', 'break']
```

Creating Vocabulary List using Word2Vec Model

As we have a set of cleaned and parsed sentences from the training data, we will train our own word vector representations by specifying the embedding dimension (= length of feature vector).

In [ ]:

```
# Fitting parsed sentences to Word2Vec model

num_features = 300  #embedding dimension
min_word_count = 10
num_workers = 4
context = 10
downsampling = 1e-3

print("Training Word2Vec model ...\n")
w2v = Word2Vec(sentences, workers=num_workers, size=num_features, min_count = min_word_count,\
                 window = context, sample = downsampling)
w2v.init_sims(replace=True)
w2v.save("w2v_300features_10minwordcounts_10context") #save trained word2vec model

print("Number of words in the vocabulary list : %d \n" %len(w2v.wv.index2word)) #4016
print("Show first 10 words in the vocabulary list  vocabulary list: \n", w2v.wv.index2word[0:10])
```

```
Training Word2Vec model ...

Number of words in the vocabulary list : 416

Show first 10 words in the vocabulary list  vocabulary list:
 ['buy', 'tablet', 'use', 'good', 'great', 'work', 'get', 'one', 'amazon', 'kindle']
```

Averaging Feature Vectors

Now we have created a vocabulary list of words with each word having a word representation(ie. feature vector of dim 300).

To find a numerical representation for a review, we run through each word in a review text. For words appear in the vocabulary list, we compute the average feature vectors of all those words. The average feature vector is the numerical represenation of the review.

In [ ]:

```
# Transfroming the training data into feature vectors

def makeFeatureVec(review, model, num_features):
    '''
    Transform a review to a feature vector by averaging feature vectors of words
    appeared in that review and in the vocabulary list created
    '''
    featureVec = np.zeros((num_features,),dtype="float32")
    nwords = 0.
    index2word_set = set(model.wv.index2word) #index2word is the vocabulary list of the Word2Vec model
    isZeroVec = True
    for word in review:
        if word in index2word_set:
            nwords = nwords + 1.
            featureVec = np.add(featureVec, model[word])
            isZeroVec = False
    if isZeroVec == False:
        featureVec = np.divide(featureVec, nwords)
    return featureVec


def getAvgFeatureVecs(reviews, model, num_features):
    '''
    Transform all reviews to feature vectors using makeFeatureVec()
    '''
    counter = 0
    reviewFeatureVecs = np.zeros((len(reviews),num_features),dtype="float32")
    for review in reviews:
        reviewFeatureVecs[counter] = makeFeatureVec(review, model,num_features)
        counter = counter + 1
    return reviewFeatureVecs
```

In [ ]:

In [ ]:

```
# Getting feature vectors for training set
X_train_cleaned = []
for review in X_train:
    X_train_cleaned.append(cleanText(review, remove_stopwords=True, split_text=True))
trainVector = getAvgFeatureVecs(X_train_cleaned, w2v, num_features)
print("Training set : %d feature vectors with %d dimensions" %trainVector.shape)


# Getting feature vectors for validation set
X_test_cleaned = []
for review in X_test:
    X_test_cleaned.append(cleanText(review, remove_stopwords=True, split_text=True))
testVector = getAvgFeatureVecs(X_test_cleaned, w2v, num_features)
print("Validation set : %d feature vectors with %d dimensions" %testVector.shape)
```

```
Training set : 1012 feature vectors with 300 dimensions
Validation set : 113 feature vectors with 300 dimensions
```

In [ ]:

```
# Getting feature vectors for training set
trainVector = getAvgFeatureVecs(X_train, w2v, num_features)
print("Training set : %d feature vectors with %d dimensions" %trainVector.shape)

# Getting feature vectors for validation set
testVector = getAvgFeatureVecs(X_test, w2v, num_features)
print("Validation set : %d feature vectors with %d dimensions" %testVector.shape)
```

```
Training set : 1012 feature vectors with 300 dimensions
Validation set : 113 feature vectors with 300 dimensions
```

Random Forest Classifer

We will now train the Random Forest Classifier using feature vectors of reviews in the training set.

In [ ]:

```
# Random Forest Classifier
rf = RandomForestClassifier(n_estimators=100)
rf.fit(trainVector, y_train)
predictions = rf.predict(testVector)
modelEvaluation(predictions)
```

```
Accuracy on validation set: 0.4779

Classification report :
              precision    recall  f1-score   support

    Negative       0.51      0.69      0.59        39
     Neutral       0.47      0.41      0.44        39
    Positive       0.42      0.31      0.36        35

    accuracy                           0.48       113
   macro avg       0.47      0.47      0.46       113
weighted avg       0.47      0.48      0.47       113


Confusion Matrix :
 [[27  6  6]
 [14 16  9]
 [12 12 11]]
```

Applying LSTM

Long Short Term Memory(LSTM) Networks are a special kind of the Recurrent Neural Networks(RNN) capable of learning long-term dependencies. LSTM can be very useful in text mining problems as it involves dependencies in the sentences which can be caught in the "memory" of the LSTM. Here, we will train a simple LSTM and LSTM with Word2Vec embedding for classifying the reviews into positive and negative sentiments using Keras library.

Simple LSTM

We need to preprocess the text data to 2D tensor before we begin fitting it into a simple LSTM. Firstly we will tokenize the corpus by considering only top words (top_words = 20000) and transforming reviews to numerical sequences using the trained tokenizer. Lastly we will make it sure that all the numerical sequences have the same length (maxlen=100) for modelling by truncating the long reviews and padding shorter reviews having zero values.

For constructing a simple LSTM, we will use embedding class in Keras to building up the first layer. This embedding layer converts numerical sequence of words into a word embedding. We should also note that the embedding class provides a convenient way to map discrete words into a continuous vector space but it doesn't take the semantic similarity of the words into account. The next layer is the LSTM layer with 128 memory units. Finally, we will use a dense output layer with a single neuron and a sigmoid activation function to make 0 or 1 prediction for the two classes (positive sentiment and negative sentiment). As it is a binary classification problem log loss is used as the loss function(binary_crossentropy in Keras). ADAM optimization algorithm will be used.

Here's the workflow in this part: -

1. **Prepare X_train and X_test to 2D tensor.**
2. **Train a simple LSTM (embedding layer => LSTM layer => dense layer).**
3. **Compile and fit the model using log loss function and ADAM optimizer.**

In [ ]:

```
df = Final_data.sample(frac=0.1, random_state=0)

# Drop missing values
df.dropna(inplace=True)

# Convert the sentiments
df.sentiment.replace(('Positive','Negative','Neutral'),(1,0,2),inplace=True)
df.head()
```

Out[ ]:

	Processed_Review	sentiment
8805	buy think would great read book play game howe...	**2**
9736	good tablet kid lot appts download game	**2**
125	item work expect great product	**1**
10143	great beginner like child limit use many apps ...	**2**
10937	buy kindle past time one come defective port b...	**2**

In [ ]:

```
# Splitting data into training set and validation
X_train, X_test, y_train, y_test = train_test_split(df['Processed_Review'], df['sentiment'], \
                                                    test_size=0.1, random_state=1)
```

In [ ]:

```
top_words = 20000
maxlen = 100
batch_size = 32
nb_classes = 3
nb_epoch = 3

# Vectorize X_train and X_test to 2D tensor
tokenizer = Tokenizer(nb_words=top_words) #Considering only top 20000 words in the corpus
tokenizer.fit_on_texts(X_train)
# tokenizer.word_index #access word-to-index dictionary of trained tokenizer

sequences_train = tokenizer.texts_to_sequences(X_train)
sequences_test = tokenizer.texts_to_sequences(X_test)

X_train_seq = sequence.pad_sequences(sequences_train, maxlen=maxlen)
X_test_seq = sequence.pad_sequences(sequences_test, maxlen=maxlen)


# One-Hot Encoding of y_train and y_test
y_train_seq = np_utils.to_categorical(y_train, nb_classes)
y_test_seq = np_utils.to_categorical(y_test, nb_classes)

print('X_train shape:', X_train_seq.shape) #(27799, 100)
print('X_test shape:', X_test_seq.shape) #(3089, 100)
print('y_train shape:', y_train_seq.shape) #(27799, 2)
print('y_test shape:', y_test_seq.shape) #(3089, 2)
```

```
X_train shape: (1012, 100)
X_test shape: (113, 100)
y_train shape: (1012, 3)
y_test shape: (113, 3)
```

In [ ]:

```
# Constructing a Simple LSTM
model1 = Sequential()
model1.add(Embedding(top_words, 128, dropout=0.2))
model1.add(LSTM(128, dropout_W=0.2, dropout_U=0.2))
model1.add(Dense(nb_classes))
model1.add(Activation('softmax'))
model1.summary()

# Compiling LSTM
model1.compile(loss='binary_crossentropy',
               optimizer='adam',
               metrics=['accuracy'])

model1.fit(X_train_seq, y_train_seq, batch_size=batch_size, nb_epoch=nb_epoch, verbose=1)

# Model Evaluation
score = model1.evaluate(X_test_seq, y_test_seq, batch_size=batch_size)
print('Test loss : {:.4f}'.format(score[0]))
print('Test accuracy : {:.4f}'.format(score[1]))
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, None, 128)         2560000
_________________________________________________________________
lstm_1 (LSTM)                (None, 128)               131584
_________________________________________________________________
dense_1 (Dense)              (None, 3)                 387
_________________________________________________________________
activation_1 (Activation)    (None, 3)                 0
=================================================================
Total params: 2,691,971
Trainable params: 2,691,971
Non-trainable params: 0
_________________________________________________________________
Epoch 1/3
1012/1012 [==============================] - 6s 6ms/step - loss: 0.6137 - accuracy: 0.6739
Epoch 2/3
1012/1012 [==============================] - 5s 5ms/step - loss: 0.4028 - accuracy: 0.8211
Epoch 3/3
1012/1012 [==============================] - 5s 5ms/step - loss: 0.1886 - accuracy: 0.9440
113/113 [==============================] - 0s 2ms/step
Test loss : 0.2017
Test accuracy : 0.9204
```

In [ ]:

```
# Getting weight matrix of the embedding layer
model1.layers[0].get_weights()[0] # weight matrix of the embedding layer, word-by-dim matrix
print("Size of weight matrix in the embedding layer : ", \
      model1.layers[0].get_weights()[0].shape) #(20000, 128)

# Getting weight matrix of the hidden layer
print("Size of weight matrix in the hidden layer : ", \
      model1.layers[1].get_weights()[0].shape) #(128, 512)  weight dim of LSTM - w

# Getting weight matrix of the output layer
print("Size of weight matrix in the output layer : ", \
      model1.layers[2].get_weights()[0].shape) #(128, 2) weight dim of dense layer
```

```
Size of weight matrix in the embedding layer :  (20000, 128)
Size of weight matrix in the hidden layer :  (128, 512)
Size of weight matrix in the output layer :  (128, 3)
```

LSTM with Word2Vec Embedding

In the simple LSTM model constructed above, the embedding class in Keras comes in handy for converting the numerical sequence of words into a word embedding but it doesn't take the semantic similarity of the words into account. The model assigns random weights to the embedding layer and learn the embeddings by minimizing the global error of the network.

Instead of using random weights we will use pretrained word embeddings for initializing the weight of an embedding layer. Here, we will use the Word2Vec embedding trained in Part 4 for intializing the weights of embedding layer in LSTM.

Initializing the weights of embedding layer in LSTM:

1. **Load pretrained word embedding model.**
2. **Construct embedding layer using embedding matrix as weights.**
3. **Train a LSTM with Word2Vec embedding (embedding layer => LSTM layer => dense layer).**
4. **Compile and fit the model using log loss function and ADAM optimizer.**

In [ ]:

```
# Loading pretrained Word2Vec model
w2v = Word2Vec.load("w2v_300features_10minwordcounts_10context")

# Getting Word2Vec embedding matrix
embedding_matrix = w2v.wv.syn0  # embedding matrix, type = numpy.ndarray
print("Shape of embedding matrix : ", embedding_matrix.shape) #(4016, 300) = (vocabulary size, embedding dimension)
# w2v.wv.syn0[0] #feature vector of the first word in the vocabulary list
```

```
Shape of embedding matrix :  (416, 300)
```

In [ ]:

```
top_words = embedding_matrix.shape[0] #4016
maxlen = 100
batch_size = 32
nb_classes = 3
nb_epoch = 3


# Vectorizing X_train and X_test to 2D tensor
tokenizer = Tokenizer(nb_words=top_words) #Considering only top 20000 words in the corpus
tokenizer.fit_on_texts(X_train)
# tokenizer.word_index #access word-to-index dictionary of trained tokenizer

sequences_train = tokenizer.texts_to_sequences(X_train)
sequences_test = tokenizer.texts_to_sequences(X_test)

X_train_seq = sequence.pad_sequences(sequences_train, maxlen=maxlen)
X_test_seq = sequence.pad_sequences(sequences_test, maxlen=maxlen)


# One-Hot Encoding of y_train and y_test
y_train_seq = np_utils.to_categorical(y_train, nb_classes)
y_test_seq = np_utils.to_categorical(y_test, nb_classes)

print('X_train shape:', X_train_seq.shape) #(27799, 100)
print('X_test shape:', X_test_seq.shape) #(3089, 100)
print('y_train shape:', y_train_seq.shape) #(27799, 2)
print('y_test shape:', y_test_seq.shape) #(3089, 2)
```

```
X_train shape: (1012, 100)
X_test shape: (113, 100)
y_train shape: (1012, 3)
y_test shape: (113, 3)
```

In [ ]:

```
# Constructing Word2Vec embedding layer
embedding_layer = Embedding(embedding_matrix.shape[0], #4016
                            embedding_matrix.shape[1], #300
                            weights=[embedding_matrix])


# Constructing LSTM with Word2Vec embedding
model2 = Sequential()
model2.add(embedding_layer)
model2.add(LSTM(128, dropout_W=0.2, dropout_U=0.2))
model2.add(Dense(nb_classes))
model2.add(Activation('softmax'))
model2.summary()

# Compiling model
model2.compile(loss='binary_crossentropy',
               optimizer='adam',
               metrics=['accuracy'])

model2.fit(X_train_seq, y_train_seq, batch_size=batch_size, nb_epoch=nb_epoch, verbose=1)


# Model evaluation
score = model2.evaluate(X_test_seq, y_test_seq, batch_size=batch_size)
print('Test loss : {:.4f}'.format(score[0]))
print('Test accuracy : {:.4f}'.format(score[1]))
```

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_2 (Embedding)      (None, None, 300)         124800
_________________________________________________________________
lstm_2 (LSTM)                (None, 128)               219648
_________________________________________________________________
dense_2 (Dense)              (None, 3)                 387
_________________________________________________________________
activation_2 (Activation)    (None, 3)                 0
=================================================================
Total params: 344,835
Trainable params: 344,835
Non-trainable params: 0
_________________________________________________________________
Epoch 1/3
1012/1012 [==============================] - 8s 8ms/step - loss: 0.6216 - accuracy: 0.6696
Epoch 2/3
1012/1012 [==============================] - 8s 8ms/step - loss: 0.4778 - accuracy: 0.7862
Epoch 3/3
1012/1012 [==============================] - 8s 8ms/step - loss: 0.2823 - accuracy: 0.8880
113/113 [==============================] - 0s 2ms/step
Test loss : 0.2717
Test accuracy : 0.9027
```

In [ ]:

```
# Getting weight matrix of the embedding layer
print("Size of weight matrix in the embedding layer : ", \
      model2.layers[0].get_weights()[0].shape) #(20000, 128)
```

```
# Getting weight matrix of the hidden layer
print("Size of weight matrix in the hidden layer : ", \
      model2.layers[1].get_weights()[0].shape) #(128, 512)  weight dim of LSTM - w

# Getting weight matrix of the output layer
print("Size of weight matrix in the output layer : ", \
      model2.layers[2].get_weights()[0].shape) #(128, 2) weight dim of dense layer
```

```
Size of weight matrix in the embedding layer :  (416, 300)
Size of weight matrix in the hidden layer :  (300, 512)
Size of weight matrix in the output layer :  (128, 3)
```

Optional Tasks: Topic Modelling

Latent Dirichlet Allocation(LDA)

In [ ]:

```
import nltk
nltk.download('wordnet')

doc_complete = data2["Processed_Review"].tolist()
doc_clean = [cleanText(doc).split() for doc in doc_complete]
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

In [ ]:

```
import gensim
from gensim import corpora
```

In [ ]:

```
dictionary = corpora.Dictionary(doc_clean)
print(dictionary)
```

```
Dictionary(3415 unique tokens: ['able', 'access', 'accomplish', 'ad', 'add']...)
```

In [ ]:

```
doc_term_matrix = [dictionary.doc2bow(doc) for doc in doc_clean]
doc_term_matrix
```

Out[ ]:

```
[[(0, 1),
  (1, 1),
  (2, 1),
  (3, 1),
  (4, 2),
  (5, 1),
  (6, 3),
  (7, 1),
  (8, 1),
  (9, 1),
  (10, 1),
  (11, 1),
  (12, 1),
  (13, 1),
  (14, 1),
  (15, 1),
  (16, 2),
  (17, 1),
  (18, 1),
  (19, 1),
  (20, 1),
  (21, 1),
  (22, 1),
  (23, 1),
  (24, 1),
  (25, 1),
  (26, 1),
  (27, 1),
  (28, 1),
  (29, 1),
  (30, 1),
  (31, 1),
  (32, 1),
  (33, 2),
  (34, 1)],
 [(6, 1),
  (26, 2),
  (34, 1),
  (35, 4),
  (36, 1),
  (37, 2),
  (38, 1),
  (39, 1),
  (40, 2),
  (41, 1),
  (42, 1),
  (43, 1),
  (44, 1),
  (45, 1),
  (46, 1),
  (47, 1),
  (48, 1),
  (49, 1),
  (50, 1),
  (51, 1),
  (52, 1),
  (53, 1),
  (54, 1),
  (55, 1),
  (56, 1),
  (57, 1),
  (58, 1),
  (59, 1),
  (60, 1),
  (61, 1),
  (62, 1),
  (63, 1),
```

```
 (64, 1),
 (65, 1),
 (66, 1),
 (67, 1),
 (68, 1),
 (69, 2),
 (70, 1),
 (71, 1),
 (72, 1),
 (73, 1),
 (74, 2),
 (75, 1),
 (76, 1),
 (77, 1),
 (78, 2),
 (79, 1),
 (80, 1),
 (81, 1),
 (82, 1),
 (83, 1)],
[(35, 1), (84, 1), (85, 1), (86, 1), (87, 1), (88, 1), (89, 1), (90, 1)],
[(24, 1), (80, 1), (91, 1), (92, 2), (93, 1)],
[(20, 3),
 (26, 1),
 (33, 1),
 (94, 1),
 (95, 1),
 (96, 1),
 (97, 1),
 (98, 1),
 (99, 1),
 (100, 1),
 (101, 1),
 (102, 1),
 (103, 1)],
[(16, 1),
 (93, 1),
 (104, 1),
 (105, 1),
 (106, 1),
 (107, 1),
 (108, 1),
 (109, 1)],
[(7, 2),
 (16, 1),
 (19, 1),
 (21, 1),
 (24, 3),
 (33, 2),
 (37, 1),
 (92, 1),
 (94, 1),
 (103, 1),
 (110, 1),
 (111, 1),
 (112, 1),
 (113, 1),
 (114, 2),
 (115, 1),
 (116, 1),
 (117, 1),
 (118, 1),
 (119, 1),
 (120, 1),
 (121, 1),
 (122, 1),
 (123, 1),
 (124, 1),
 (125, 1),
 (126, 2),
 (127, 1),
 (128, 1),
 (129, 1),
 (130, 1),
 (131, 1),
 (132, 1),
 (133, 1),
 (134, 1),
 (135, 1),
 (136, 1),
 (137, 2),
 (138, 1),
 (139, 1),
 (140, 1),
 (141, 1),
 (142, 1),
 (143, 1),
 (144, 1),
 (145, 1),
 (146, 1),
 (147, 1),
 (148, 1),
 (149, 1),
 (150, 1),
 (151, 1),
 (152, 1),
 (153, 2)],
[(26, 1),
 (92, 2),
 (114, 1),
 (128, 1),
 (154, 1),
 (155, 1),
 (156, 1),
 (157, 1),
 (158, 1),
 (159, 1),
 (160, 1),
 (161, 1),
 (162, 1)],
[(16, 2),
 (19, 1),
 (24, 1),
 (77, 1),
 (80, 1),
 (86, 1)
```

```
 (88, 1),
 (94, 1),
 (134, 1),
 (163, 1),
 (164, 1),
 (165, 1),
 (166, 1),
 (167, 1),
 (168, 1)],
[(169, 1), (170, 1), (171, 1), (172, 1), (173, 1), (174, 1)],
[(20, 1),
 (33, 1),
 (36, 1),
 (95, 1),
 (99, 1),
 (103, 1),
 (110, 1),
 (139, 1),
 (153, 1),
 (160, 1),
 (164, 1),
 (175, 1),
 (176, 1),
 (177, 1),
 (178, 1),
 (179, 1),
 (180, 1),
 (181, 1),
 (182, 1),
 (183, 1)],
[(7, 1),
 (16, 1),
 (26, 1),
 (93, 1),
 (120, 1),
 (128, 1),
 (133, 2),
 (142, 1),
 (184, 1),
 (185, 1),
 (186, 1),
 (187, 1),
 (188, 1),
 (189, 1)],
[(26, 1), (78, 1), (179, 1), (190, 1), (191, 1), (192, 1), (193, 1)],
[(49, 1), (143, 1), (179, 1), (181, 1), (193, 1), (194, 1)],
[(53, 1), (94, 1), (176, 1), (181, 1), (195, 1), (196, 1)],
[(16, 1),
 (26, 1),
 (104, 1),
 (179, 1),
 (197, 1),
 (198, 1),
 (199, 1),
 (200, 1),
 (201, 1),
 (202, 1),
 (203, 1)],
[(47, 1),
 (64, 1),
 (92, 1),
 (204, 1),
 (205, 1),
 (206, 1),
 (207, 1),
 (208, 1),
 (209, 1)],
[(92, 2), (139, 1), (193, 1), (210, 1), (211, 1)],
[(15, 1),
 (24, 1),
 (29, 1),
 (55, 1),
 (56, 1),
 (62, 1),
 (80, 1),
 (103, 1),
 (105, 1),
 (120, 1),
 (164, 1),
 (212, 1),
 (213, 1),
 (214, 1),
 (215, 1),
 (216, 1),
 (217, 1),
 (218, 1),
 (219, 1)],
[(18, 1),
 (20, 1),
 (35, 1),
 (36, 1),
 (39, 1),
 (51, 1),
 (66, 1),
 (86, 1),
 (87, 1),
 (90, 1),
 (155, 1),
 (164, 2),
 (179, 2),
 (198, 1),
 (216, 1),
 (219, 1),
 (220, 1),
 (221, 1),
 (222, 1),
 (223, 1),
 (224, 2),
 (225, 1),
 (226, 1),
 (227, 1),
 (228, 1),
 (229, 1),
 (230, 1),
 (231, 1),
 (232, 1)],
```

```
 (232, 1),
 (233, 1),
 (234, 1),
 (235, 1),
 (236, 1),
 (237, 1),
 (238, 1),
 (239, 1),
 (240, 1),
 (241, 1),
 (242, 1),
 (243, 2),
 (244, 1)],
[(35, 1), (120, 1), (175, 1), (245, 1), (246, 1)],
[(16, 2), (33, 1), (179, 1), (227, 1), (247, 1)],
[(6, 1), (20, 1), (174, 1), (248, 1), (249, 1), (250, 1), (251, 1), (252, 1)],
[(136, 1),
 (253, 1),
 (254, 1),
 (255, 1),
 (256, 1),
 (257, 1),
 (258, 1),
 (259, 1),
 (260, 1),
 (261, 1)],
[(6, 1),
 (15, 1),
 (65, 1),
 (80, 1),
 (163, 1),
 (168, 1),
 (211, 1),
 (252, 1),
 (262, 1),
 (263, 1),
 (264, 1),
 (265, 1),
 (266, 1),
 (267, 1)],
[(16, 1),
 (24, 1),
 (36, 1),
 (79, 1),
 (92, 1),
 (143, 1),
 (167, 1),
 (192, 1),
 (197, 1),
 (217, 1),
 (234, 1),
 (268, 1)],
[(16, 1), (105, 1), (193, 1), (197, 1), (269, 1), (270, 1)],
[(33, 1), (79, 1), (80, 1), (83, 2), (94, 1), (197, 1), (257, 1), (271, 1)],
[(62, 2),
 (110, 1),
 (133, 1),
 (165, 1),
 (168, 1),
 (179, 1),
 (272, 1),
 (273, 1),
 (274, 1),
 (275, 1),
 (276, 1),
 (277, 1),
 (278, 1)],
[(92, 1), (120, 1), (279, 1), (280, 1), (281, 1), (282, 1), (283, 1)],
[(6, 1),
 (35, 1),
 (63, 1),
 (83, 2),
 (88, 1),
 (90, 1),
 (104, 1),
 (109, 1),
 (136, 1),
 (164, 1),
 (179, 1),
 (184, 1),
 (263, 1),
 (284, 1),
 (285, 1),
 (286, 1),
 (287, 1),
 (288, 1),
 (289, 1),
 (290, 1),
 (291, 1),
 (292, 1),
 (293, 1),
 (294, 1),
 (295, 1),
 (296, 1)],
[(22, 1),
 (52, 1),
 (65, 1),
 (79, 2),
 (90, 1),
 (172, 1),
 (202, 1),
 (297, 1),
 (298, 1),
 (299, 1),
 (300, 1),
 (301, 1)],
[(7, 1), (16, 1), (79, 1), (197, 2), (201, 1), (217, 1), (302, 1)],
[(16, 1),
 (19, 2),
 (20, 1),
 (33, 1),
 (111, 1),
 (114, 1),
 (128, 1),
 (164, 1),
 (165, 1),
```

```
 (243, 1),
 (303, 1),
 (304, 1),
 (305, 1)],
[(7, 1),
 (19, 2),
 (33, 1),
 (51, 1),
 (62, 1),
 (80, 1),
 (83, 1),
 (92, 1),
 (165, 1),
 (194, 1),
 (255, 1),
 (287, 1),
 (306, 1),
 (307, 1),
 (308, 1),
 (309, 1),
 (310, 1),
 (311, 1),
 (312, 1),
 (313, 1)],
[(47, 1), (88, 1), (92, 1), (207, 1), (211, 1), (314, 1), (315, 1)],
[(16, 1), (79, 1), (92, 1), (143, 1), (197, 2), (201, 1), (217, 1), (316, 1)],
[(16, 1),
 (20, 1),
 (33, 2),
 (79, 1),
 (92, 1),
 (165, 1),
 (317, 1),
 (318, 1),
 (319, 1),
 (320, 1),
 (321, 1),
 (322, 1),
 (323, 1),
 (324, 1)],
[(6, 1),
 (16, 2),
 (43, 1),
 (47, 2),
 (62, 2),
 (81, 1),
 (88, 2),
 (90, 2),
 (223, 1),
 (325, 1),
 (326, 1),
 (327, 1),
 (328, 1),
 (329, 1),
 (330, 1),
 (331, 1),
 (332, 1),
 (333, 1),
 (334, 1),
 (335, 1)],
[(19, 1),
 (69, 1),
 (92, 1),
 (94, 1),
 (103, 1),
 (207, 1),
 (336, 1),
 (337, 1),
 (338, 1)],
[(5, 1),
 (6, 1),
 (16, 2),
 (20, 1),
 (33, 1),
 (43, 1),
 (62, 1),
 (79, 1),
 (88, 1),
 (90, 1),
 (113, 1),
 (128, 1),
 (152, 1),
 (211, 1),
 (252, 1),
 (258, 1),
 (339, 1),
 (340, 1),
 (341, 1),
 (342, 1),
 (343, 1),
 (344, 1),
 (345, 1),
 (346, 1),
 (347, 1)],
[(7, 1), (33, 1), (92, 1), (177, 1), (183, 1), (300, 1), (348, 1), (349, 1)],
[(79, 1), (263, 1), (350, 1), (351, 1), (352, 1)],
[(244, 1), (333, 1), (353, 1), (354, 1), (355, 1), (356, 1)],
[(16, 1),
 (19, 1),
 (35, 2),
 (36, 1),
 (47, 5),
 (62, 1),
 (80, 1),
 (140, 1),
 (161, 1),
 (211, 1),
 (258, 1),
 (325, 1),
 (357, 1),
 (358, 1),
 (359, 1),
 (360, 1),
 (361, 1),
 (362, 1),
```

```
 (363, 1),
 (364, 1),
 (365, 1),
 (366, 1),
 (367, 3),
 (368, 1)],
[(16, 1), (93, 1), (216, 1), (302, 1), (316, 1)],
[(6, 1),
 (67, 1),
 (79, 1),
 (87, 1),
 (90, 1),
 (94, 1),
 (125, 2),
 (143, 1),
 (150, 2),
 (155, 1),
 (176, 1),
 (201, 3),
 (234, 1),
 (252, 1),
 (262, 1),
 (369, 1),
 (370, 1),
 (371, 1),
 (372, 1),
 (373, 1),
 (374, 1),
 (375, 1),
 (376, 1),
 (377, 1),
 (378, 1),
 (379, 1)],
[(33, 1),
 (61, 2),
 (79, 1),
 (149, 1),
 (380, 1),
 (381, 1),
 (382, 1),
 (383, 1),
 (384, 1)],
[(6, 1),
 (49, 1),
 (102, 1),
 (114, 1),
 (231, 2),
 (318, 1),
 (343, 1),
 (385, 2),
 (386, 1),
 (387, 1),
 (388, 1),
 (389, 1)],
[(0, 1),
 (5, 1),
 (8, 1),
 (16, 1),
 (19, 2),
 (29, 1),
 (35, 1),
 (47, 1),
 (66, 1),
 (77, 1),
 (127, 1),
 (159, 1),
 (179, 1),
 (188, 1),
 (201, 1),
 (207, 1),
 (216, 1),
 (230, 1),
 (238, 3),
 (306, 1),
 (315, 1),
 (316, 1),
 (372, 1),
 (380, 1),
 (390, 1),
 (391, 1),
 (392, 1),
 (393, 1),
 (394, 1),
 (395, 1),
 (396, 1),
 (397, 1),
 (398, 1),
 (399, 1),
 (400, 1),
 (401, 1),
 (402, 1),
 (403, 1),
 (404, 1)],
[(19, 1),
 (33, 1),
 (49, 1),
 (77, 1),
 (92, 1),
 (94, 1),
 (139, 1),
 (179, 1),
 (181, 1),
 (192, 1),
 (296, 1),
 (405, 1),
 (406, 1),
 (407, 1),
 (408, 1),
 (409, 1)],
[(22, 1),
 (65, 1),
 (179, 1),
 (185, 1),
 (201, 1),
 (292, 1),
```

```
 (325, 1),
 (396, 1),
 (410, 1),
 (411, 1),
 (412, 1)],
[(24, 1),
 (47, 2),
 (64, 1),
 (69, 1),
 (87, 1),
 (90, 1),
 (185, 1),
 (197, 1),
 (413, 1),
 (414, 1),
 (415, 1),
 (416, 1)],
[(62, 1), (79, 1), (95, 1), (168, 1), (197, 1), (211, 1), (256, 1), (417, 1)],
[(16, 1), (418, 1), (419, 1), (420, 1), (421, 1), (422, 1), (423, 1)],
[(5, 2), (8, 1), (24, 1), (205, 1), (287, 1), (395, 1), (424, 1)],
[(191, 1),
 (230, 1),
 (234, 1),
 (425, 1),
 (426, 1),
 (427, 1),
 (428, 1),
 (429, 1),
 (430, 1)],
[(49, 1),
 (133, 1),
 (139, 1),
 (153, 1),
 (193, 1),
 (197, 1),
 (211, 1),
 (255, 1)],
[(35, 1), (79, 1), (88, 1), (250, 1), (431, 1)],
[(1, 1),
 (22, 1),
 (34, 1),
 (36, 1),
 (62, 1),
 (63, 1),
 (87, 2),
 (88, 1),
 (103, 1),
 (106, 1),
 (133, 1),
 (164, 1),
 (165, 1),
 (176, 1),
 (200, 1),
 (201, 1),
 (243, 1),
 (247, 1),
 (249, 1),
 (293, 1),
 (300, 1),
 (329, 1),
 (401, 1),
 (432, 1),
 (433, 1),
 (434, 1),
 (435, 1),
 (436, 1),
 (437, 1),
 (438, 1),
 (439, 1),
 (440, 1),
 (441, 1),
 (442, 2),
 (443, 1),
 (444, 1),
 (445, 1),
 (446, 1)],
[(24, 2),
 (33, 1),
 (262, 1),
 (447, 1),
 (448, 1),
 (449, 1),
 (450, 1),
 (451, 1),
 (452, 1)],
[(33, 1), (201, 1), (453, 1), (454, 1), (455, 1), (456, 1)],
[(52, 1), (62, 1), (94, 1), (243, 1), (250, 1), (457, 1), (458, 1)],
[(33, 1), (49, 1), (127, 1), (179, 1), (193, 1), (434, 1)],
[(7, 1),
 (33, 1),
 (92, 1),
 (109, 1),
 (127, 1),
 (133, 1),
 (164, 1),
 (197, 1),
 (230, 1),
 (277, 1)],
[(127, 1),
 (164, 1),
 (185, 1),
 (197, 2),
 (205, 1),
 (281, 1),
 (318, 1),
 (459, 1),
 (460, 1)],
[(48, 1),
 (86, 1),
 (114, 1),
 (164, 1),
 (192, 1),
 (268, 1),
 (387, 1),
 (461, 1)],
[(25, 1), (65, 1), (93, 2), (211, 1), (217, 1), (429, 1), (462, 1), (463, 1)]
```

```
[(33, 1), (65, 1), (92, 2), (211, 1), (217, 1), (429, 1), (462, 1), (463, 1)],
[(16, 1),
 (26, 1),
 (47, 1),
 (81, 1),
 (179, 1),
 (352, 1),
 (367, 1),
 (464, 1),
 (465, 1),
 (466, 1),
 (467, 1),
 (468, 1),
 (469, 1)],
[(21, 1),
 (33, 1),
 (92, 1),
 (128, 1),
 (164, 1),
 (342, 1),
 (429, 1),
 (470, 1),
 (471, 1)],
[(34, 1), (48, 1), (268, 1), (285, 1), (387, 1), (419, 1), (472, 1)],
[(16, 1),
 (22, 1),
 (79, 1),
 (103, 1),
 (114, 1),
 (128, 1),
 (152, 1),
 (343, 1),
 (346, 1)],
[(16, 3),
 (35, 1),
 (36, 1),
 (47, 1),
 (64, 1),
 (65, 1),
 (71, 1),
 (79, 1),
 (88, 1),
 (185, 1),
 (217, 2),
 (234, 1),
 (264, 1),
 (296, 1),
 (325, 1),
 (331, 1),
 (357, 1),
 (396, 1),
 (429, 1),
 (473, 1),
 (474, 1),
 (475, 1),
 (476, 1),
 (477, 1),
 (478, 1)],
[(20, 2),
 (26, 1),
 (36, 1),
 (62, 1),
 (77, 1),
 (79, 1),
 (83, 1),
 (277, 1),
 (296, 1),
 (300, 1),
 (341, 1),
 (479, 1)],
[(89, 1), (103, 1), (160, 1), (480, 1), (481, 1), (482, 1), (483, 1)],
[(20, 1), (35, 1), (80, 1), (159, 1), (263, 1), (329, 1), (484, 1)],
[(0, 1),
 (6, 1),
 (13, 1),
 (16, 1),
 (24, 1),
 (37, 2),
 (52, 1),
 (64, 2),
 (77, 1),
 (79, 1),
 (80, 1),
 (83, 1),
 (86, 1),
 (89, 2),
 (105, 1),
 (134, 1),
 (141, 1),
 (150, 1),
 (230, 1),
 (232, 1),
 (235, 1),
 (295, 1),
 (318, 2),
 (343, 2),
 (353, 1),
 (385, 2),
 (426, 1),
 (485, 1),
 (486, 1),
 (487, 2),
 (488, 1),
 (489, 1),
 (490, 1),
 (491, 2),
 (492, 1),
 (493, 1),
 (494, 1),
 (495, 1),
 (496, 1),
 (497, 1),
 (498, 1),
 (499, 1),
 (500, 1),
 (501, 1),
```

```
 (501, 1),
 (502, 1),
 (503, 1),
 (504, 1),
 (505, 1),
 (506, 1),
 (507, 1),
 (508, 3),
 (509, 1),
 (510, 2)],
[(7, 1), (20, 1), (33, 1), (94, 1), (179, 1), (511, 1), (512, 1), (513, 1)],
[(33, 1), (80, 1), (94, 1), (139, 1), (306, 1), (514, 1), (515, 1)],
[(26, 1),
 (37, 1),
 (94, 1),
 (106, 1),
 (136, 1),
 (155, 1),
 (173, 1),
 (193, 1),
 (435, 2),
 (516, 1),
 (517, 1),
 (518, 1),
 (519, 1),
 (520, 1)],
[(16, 1), (86, 1), (179, 1), (194, 1), (521, 1), (522, 1)],
[(49, 1),
 (52, 1),
 (114, 2),
 (136, 1),
 (152, 1),
 (193, 1),
 (248, 1),
 (264, 1),
 (343, 1),
 (357, 1),
 (397, 1),
 (444, 1),
 (523, 1),
 (524, 1)],
[(16, 1), (33, 1), (94, 1), (331, 1), (525, 1), (526, 1)],
[(26, 1),
 (45, 1),
 (49, 1),
 (62, 1),
 (136, 1),
 (153, 1),
 (171, 1),
 (181, 1),
 (194, 1),
 (258, 1),
 (368, 1),
 (388, 1),
 (435, 1),
 (527, 1),
 (528, 1),
 (529, 1),
 (530, 1),
 (531, 1),
 (532, 1)],
[(39, 1),
 (47, 1),
 (64, 1),
 (79, 1),
 (88, 1),
 (179, 1),
 (197, 1),
 (286, 1),
 (517, 1),
 (533, 1),
 (534, 1),
 (535, 1)],
[(20, 1), (24, 1), (79, 1), (87, 1), (388, 1), (491, 1), (536, 1), (537, 1)],
[(79, 1),
 (94, 1),
 (139, 1),
 (153, 1),
 (179, 1),
 (197, 1),
 (435, 1),
 (538, 1)],
[(61, 1), (89, 1), (125, 1)],
[(12, 1),
 (16, 2),
 (64, 1),
 (114, 1),
 (193, 1),
 (230, 1),
 (302, 1),
 (306, 1),
 (318, 1),
 (343, 1),
 (385, 1),
 (539, 1),
 (540, 1),
 (541, 1),
 (542, 1),
 (543, 1),
 (544, 1),
 (545, 1)],
[(61, 1), (79, 1), (93, 1), (103, 1), (197, 2), (268, 1), (546, 1)],
[(79, 1), (92, 2), (93, 2), (143, 1), (197, 1)],
[(133, 1),
 (179, 1),
 (225, 1),
 (233, 1),
 (244, 1),
 (263, 1),
 (547, 1),
 (548, 1)],
[(16, 1),
 (36, 1),
 (48, 1),
 (71, 1),
 (97, 1),
```

```
 (316, 1),
 (419, 1),
 (458, 1),
 (474, 1),
 (549, 1),
 (550, 1)],
[(161, 1),
 (179, 1),
 (197, 1),
 (231, 1),
 (239, 1),
 (367, 1),
 (374, 2),
 (551, 1),
 (552, 1)],
[(83, 1),
 (193, 1),
 (207, 1),
 (238, 1),
 (255, 1),
 (315, 1),
 (354, 1),
 (434, 1),
 (517, 1),
 (553, 1),
 (554, 1)],
[(11, 1),
 (93, 1),
 (150, 1),
 (201, 1),
 (425, 1),
 (555, 1),
 (556, 1),
 (557, 1)],
[(16, 1), (139, 1), (153, 1), (160, 1), (329, 1), (558, 1)],
[(6, 1),
 (7, 1),
 (12, 1),
 (19, 1),
 (22, 1),
 (35, 2),
 (39, 2),
 (46, 1),
 (47, 4),
 (65, 1),
 (77, 1),
 (79, 1),
 (80, 1),
 (81, 1),
 (83, 1),
 (85, 1),
 (88, 2),
 (90, 2),
 (93, 1),
 (94, 1),
 (136, 1),
 (170, 1),
 (179, 1),
 (185, 1),
 (209, 1),
 (243, 5),
 (251, 1),
 (262, 2),
 (283, 2),
 (289, 1),
 (292, 1),
 (296, 1),
 (319, 1),
 (329, 1),
 (349, 1),
 (432, 1),
 (435, 1),
 (511, 1),
 (549, 1),
 (559, 1),
 (560, 1),
 (561, 1),
 (562, 1),
 (563, 1),
 (564, 1),
 (565, 1),
 (566, 1),
 (567, 1),
 (568, 1),
 (569, 1),
 (570, 1),
 (571, 1),
 (572, 1),
 (573, 1),
 (574, 1),
 (575, 1),
 (576, 1),
 (577, 1),
 (578, 1),
 (579, 1),
 (580, 1),
 (581, 1),
 (582, 1),
 (583, 2)],
[(16, 1),
 (33, 1),
 (83, 1),
 (207, 1),
 (241, 1),
 (510, 1),
 (511, 1),
 (540, 1),
 (584, 1),
 (585, 1),
 (586, 1)],
[(16, 1),
 (26, 1),
 (121, 1),
 (179, 1),
 (192, 1),
```

```
 (256, 1),
 (492, 1),
 (587, 1)],
[(16, 1),
 (37, 1),
 (94, 2),
 (174, 1),
 (179, 1),
 (306, 1),
 (329, 1),
 (588, 1),
 (589, 1),
 (590, 1)],
[(26, 1), (78, 1), (179, 1), (190, 1), (191, 1), (192, 1), (193, 1)],
[(46, 1), (87, 1), (92, 1), (141, 1), (168, 1), (211, 1), (217, 1), (591, 1)],
[(37, 1),
 (79, 1),
 (94, 1),
 (99, 1),
 (197, 1),
 (201, 1),
 (329, 1),
 (406, 1),
 (588, 1),
 (592, 1),
 (593, 1)],
[(6, 1),
 (7, 1),
 (33, 1),
 (92, 1),
 (99, 1),
 (211, 1),
 (303, 1),
 (334, 1),
 (373, 1),
 (434, 1),
 (485, 1),
 (594, 1),
 (595, 1)],
[(16, 1), (26, 1), (47, 1), (88, 1), (99, 1), (179, 1), (406, 1)],
[(19, 1),
 (86, 1),
 (92, 1),
 (120, 1),
 (125, 1),
 (139, 1),
 (152, 1),
 (153, 1),
 (164, 1),
 (179, 1),
 (531, 1),
 (596, 1)],
[(87, 1),
 (91, 1),
 (104, 1),
 (114, 1),
 (127, 1),
 (185, 1),
 (193, 1),
 (197, 1),
 (256, 1),
 (263, 1),
 (343, 1),
 (595, 1),
 (597, 1)],
[(24, 1),
 (63, 1),
 (244, 1),
 (268, 1),
 (387, 1),
 (458, 1),
 (492, 1),
 (598, 1)],
[(83, 1), (122, 1), (205, 1), (207, 1), (268, 1), (387, 1), (599, 1)],
[(64, 1),
 (87, 1),
 (117, 1),
 (185, 1),
 (244, 1),
 (258, 1),
 (288, 1),
 (353, 1),
 (458, 1),
 (592, 1),
 (600, 1),
 (601, 1),
 (602, 1),
 (603, 1),
 (604, 1),
 (605, 1),
 (606, 1)],
[(16, 1),
 (19, 1),
 (26, 1),
 (35, 3),
 (47, 2),
 (77, 1),
 (98, 1),
 (143, 1),
 (161, 1),
 (201, 1),
 (211, 1),
 (217, 1),
 (262, 1),
 (363, 2),
 (367, 2),
 (519, 1),
 (549, 1),
 (607, 1),
 (608, 1),
 (609, 1),
 (610, 1),
 (611, 2),
 (612, 1),
 (613, 2),
```

```
 (614, 1),
 (615, 1),
 (616, 1)],
[(19, 2),
 (22, 1),
 (24, 1),
 (33, 3),
 (128, 1),
 (143, 1),
 (152, 1),
 (169, 1),
 (211, 2),
 (271, 1),
 (435, 1),
 (442, 1),
 (458, 1),
 (617, 1),
 (618, 1),
 (619, 1),
 (620, 1),
 (621, 1),
 (622, 1),
 (623, 1)],
[(6, 1),
 (19, 1),
 (20, 1),
 (24, 1),
 (53, 1),
 (69, 1),
 (112, 1),
 (133, 1),
 (139, 1),
 (194, 1),
 (252, 1),
 (276, 1),
 (296, 2),
 (306, 1),
 (409, 1),
 (515, 1),
 (603, 1),
 (624, 1),
 (625, 1),
 (626, 1)],
[(94, 1),
 (139, 1),
 (153, 1),
 (179, 1),
 (197, 1),
 (228, 1),
 (611, 1),
 (627, 1),
 (628, 1)],
[(19, 1),
 (62, 1),
 (120, 1),
 (142, 1),
 (176, 2),
 (180, 1),
 (197, 1),
 (370, 1),
 (629, 1),
 (630, 1)],
[(7, 1), (16, 1), (19, 1), (33, 1), (128, 1), (190, 1), (631, 1)],
[(79, 1),
 (94, 1),
 (197, 1),
 (296, 1),
 (435, 1),
 (632, 1),
 (633, 1),
 (634, 1)],
[(20, 1),
 (22, 1),
 (26, 1),
 (33, 1),
 (79, 1),
 (128, 1),
 (371, 1),
 (635, 1),
 (636, 1),
 (637, 1),
 (638, 1),
 (639, 1),
 (640, 1)],
[(20, 1),
 (37, 1),
 (46, 1),
 (87, 1),
 (164, 1),
 (231, 1),
 (240, 1),
 (325, 1),
 (373, 2),
 (641, 1)],
[(6, 1),
 (20, 1),
 (26, 2),
 (41, 1),
 (49, 1),
 (51, 1),
 (62, 1),
 (76, 1),
 (80, 1),
 (83, 1),
 (179, 1),
 (367, 1),
 (479, 1),
 (587, 1),
 (642, 1),
 (643, 1),
 (644, 1),
 (645, 1)],
[(19, 1),
 (20, 1),
 (52, 1),
 (94, 5)
```

```
 (94, 5),
 (103, 1),
 (115, 1),
 (121, 1),
 (139, 1),
 (153, 1),
 (164, 1),
 (168, 1),
 (181, 1),
 (183, 2),
 (193, 1),
 (272, 1),
 (307, 1),
 (374, 1),
 (380, 1),
 (646, 1),
 (647, 1),
 (648, 1),
 (649, 1),
 (650, 1),
 (651, 1),
 (652, 1),
 (653, 1)],
[(0, 1),
 (52, 2),
 (105, 2),
 (159, 1),
 (302, 1),
 (403, 1),
 (426, 1),
 (613, 1),
 (654, 1),
 (655, 1),
 (656, 1),
 (657, 1),
 (658, 1),
 (659, 1)],
[(24, 1),
 (26, 1),
 (33, 1),
 (37, 1),
 (53, 1),
 (62, 1),
 (168, 1),
 (211, 1),
 (539, 1),
 (660, 1),
 (661, 1),
 (662, 1)],
[(79, 1), (94, 1), (179, 1), (193, 1), (197, 1), (636, 1)],
[(16, 1), (93, 1), (211, 1), (247, 1), (314, 1)],
[(120, 1),
 (121, 1),
 (133, 1),
 (142, 1),
 (179, 1),
 (185, 1),
 (186, 1),
 (197, 1),
 (331, 1)],
[(41, 1), (164, 1), (192, 1), (302, 1), (518, 1), (663, 1), (664, 1)],
[(0, 1),
 (57, 1),
 (105, 2),
 (159, 1),
 (172, 1),
 (175, 1),
 (177, 1),
 (179, 1),
 (264, 1),
 (299, 1),
 (357, 1),
 (435, 1),
 (665, 1),
 (666, 1),
 (667, 1)],
[(13, 1),
 (20, 1),
 (33, 2),
 (34, 1),
 (83, 1),
 (86, 1),
 (92, 1),
 (207, 1),
 (215, 1),
 (231, 1),
 (315, 1),
 (345, 1),
 (391, 1),
 (408, 1),
 (668, 1),
 (669, 1)],
[(94, 1),
 (133, 1),
 (165, 1),
 (169, 1),
 (181, 1),
 (197, 1),
 (230, 1),
 (309, 1),
 (670, 1),
 (671, 1),
 (672, 1),
 (673, 1)],
[(0, 1),
 (36, 1),
 (39, 1),
 (41, 1),
 (62, 1),
 (92, 1),
 (161, 1),
 (164, 1),
 (185, 1),
 (217, 1),
 (243, 1),
 (268, 1),
```

```
 (200, 1),
 (663, 1)],
[(16, 1),
 (20, 1),
 (179, 1),
 (193, 1),
 (211, 1),
 (276, 1),
 (296, 1),
 (469, 1),
 (472, 1)],
[(37, 1), (49, 1), (94, 2), (193, 1), (205, 1), (249, 1)],
[(13, 1),
 (16, 1),
 (33, 2),
 (64, 1),
 (92, 1),
 (139, 2),
 (368, 1),
 (373, 1),
 (674, 1)],
[(37, 1),
 (47, 1),
 (51, 1),
 (66, 1),
 (81, 1),
 (88, 1),
 (179, 1),
 (328, 1),
 (406, 1),
 (675, 1),
 (676, 1),
 (677, 1),
 (678, 1)],
[(22, 1),
 (35, 1),
 (109, 1),
 (185, 1),
 (216, 1),
 (435, 1),
 (535, 1),
 (679, 1)],
[(16, 1),
 (22, 2),
 (95, 1),
 (113, 1),
 (120, 1),
 (128, 1),
 (139, 1),
 (153, 1),
 (164, 1),
 (179, 1),
 (185, 1),
 (197, 1),
 (237, 1),
 (247, 1),
 (374, 2),
 (420, 1),
 (496, 1),
 (512, 1),
 (647, 1),
 (650, 1),
 (651, 1),
 (671, 2),
 (680, 1),
 (681, 1),
 (682, 1),
 (683, 1),
 (684, 1),
 (685, 1),
 (686, 1)],
[(16, 1),
 (47, 1),
 (133, 1),
 (155, 1),
 (179, 1),
 (247, 1),
 (687, 1),
 (688, 1)],
[(22, 1), (61, 1), (89, 1), (418, 1), (420, 1), (642, 1), (689, 1), (690, 1)],
[(5, 1),
 (35, 2),
 (64, 1),
 (94, 2),
 (168, 1),
 (211, 1),
 (247, 1),
 (343, 1),
 (387, 1),
 (691, 1),
 (692, 1),
 (693, 1),
 (694, 1)],
[(16, 1),
 (35, 1),
 (39, 1),
 (47, 1),
 (61, 1),
 (88, 1),
 (90, 1),
 (104, 1),
 (168, 1),
 (184, 1),
 (197, 1),
 (211, 1),
 (217, 1),
 (243, 2),
 (292, 1),
 (348, 1),
 (351, 1),
 (695, 1),
 (696, 1),
 (697, 1),
 (698, 1)],
[(7, 1),
 (20, 1),
```

```
 (33, 1),
 (85, 1),
 (133, 1),
 (165, 1),
 (268, 1),
 (381, 1),
 (699, 1)],
[(26, 1),
 (33, 1),
 (36, 1),
 (94, 1),
 (95, 1),
 (100, 1),
 (101, 1),
 (133, 1),
 (139, 1),
 (153, 1),
 (179, 1),
 (197, 1),
 (201, 1),
 (349, 1),
 (700, 1)],
[(6, 1), (33, 1), (168, 1), (202, 1), (211, 1), (243, 1), (701, 1), (702, 1)],
[(95, 1),
 (113, 1),
 (120, 1),
 (139, 1),
 (142, 1),
 (153, 1),
 (157, 1),
 (164, 2),
 (249, 1),
 (442, 1),
 (458, 2),
 (619, 1),
 (641, 1),
 (703, 1),
 (704, 1),
 (705, 1),
 (706, 1)],
[(0, 1),
 (33, 1),
 (92, 1),
 (94, 1),
 (98, 1),
 (128, 1),
 (139, 1),
 (153, 1),
 (164, 1),
 (179, 1),
 (268, 1),
 (304, 1),
 (707, 1)],
[(20, 1),
 (37, 1),
 (89, 1),
 (94, 1),
 (103, 1),
 (129, 1),
 (133, 2),
 (138, 2),
 (143, 1),
 (147, 1),
 (150, 1),
 (177, 2),
 (191, 1),
 (193, 1),
 (255, 1),
 (288, 1),
 (373, 1),
 (374, 1),
 (683, 1),
 (708, 1),
 (709, 1),
 (710, 1),
 (711, 1),
 (712, 1),
 (713, 1)],
[(33, 1),
 (92, 1),
 (165, 1),
 (304, 1),
 (343, 1),
 (387, 1),
 (714, 1),
 (715, 1),
 (716, 1)],
[(11, 1),
 (36, 2),
 (52, 1),
 (62, 1),
 (87, 1),
 (92, 1),
 (96, 1),
 (164, 1),
 (193, 1),
 (197, 3),
 (258, 1),
 (281, 1),
 (300, 1),
 (316, 1),
 (343, 1),
 (367, 1),
 (387, 2),
 (531, 1),
 (595, 1),
 (609, 1),
 (626, 1),
 (664, 1),
 (717, 1),
 (718, 1),
 (719, 1),
 (720, 2),
 (721, 1)],
[(0, 1),
```

```
    (1, 1),
    (6, 2),
    (16, 1),
    (18, 1),
    (20, 1),
    (26, 2),
    (37, 1),
    (79, 1),
    (83, 1),
    (88, 2),
    (92, 1),
    (93, 2),
    (94, 1),
    (104, 1),
    (109, 1),
    (155, 1),
    (164, 1),
    (165, 1),
    (168, 1),
    (171, 1),
    (193, 1),
    (216, 1),
    (240, 1),
    (262, 4),
    (286, 1),
    (304, 1),
    (307, 1),
    (319, 1),
    (343, 2),
    (380, 1),
    (472, 1),
    (540, 2),
    (607, 1),
    (608, 1),
    (626, 1),
    (629, 2),
    (641, 1),
    (667, 1),
    (703, 2),
    (706, 2),
    (722, 1),
    (723, 1),
    (724, 1),
    (725, 1),
    (726, 1),
    (727, 2),
    (728, 1),
    (729, 1),
    (730, 1),
    (731, 1),
    (732, 1),
    (733, 1),
    (734, 1),
    (735, 1),
    (736, 1),
    (737, 1),
    (738, 1),
    (739, 2),
    (740, 1),
    (741, 1),
    (742, 1)],
   [(6, 2),
    (20, 1),
    (52, 1),
    (74, 1),
    (89, 1),
    (149, 1),
    (164, 1),
    (201, 1),
    (216, 1),
    (230, 1),
    (259, 1),
    (262, 1),
    (397, 1),
    (442, 2),
    (472, 1),
    (494, 1),
    (704, 1),
    (729, 1),
    (743, 1),
    (744, 3),
    (745, 1),
    (746, 1),
    (747, 1),
    (748, 1),
    (749, 1),
    (750, 1),
    (751, 1)],
   [(19, 1),
    (62, 1),
    (64, 1),
    (79, 1),
    (94, 1),
    (146, 1),
    (203, 1),
    (296, 1),
    (399, 1),
    (434, 1),
    (535, 1),
    (583, 1),
    (703, 1),
    (706, 1),
    (752, 1),
    (753, 1)],
   [(22, 1),
    (65, 1),
    (94, 1),
    (139, 1),
    (153, 1),
    (179, 1),
    (300, 1),
    (325, 1),
    (411, 1),
    (420, 1),
    (754, 1)],
```

```
[(47, 1), (69, 1), (262, 1), (300, 1), (302, 1), (730, 1), (755, 1)],
[(16, 1), (53, 1), (61, 1), (133, 1), (329, 1), (343, 1), (514, 1)],
[(16, 1), (33, 1), (133, 1), (179, 1), (195, 1)],
[(155, 1),
 (164, 1),
 (179, 1),
 (191, 1),
 (230, 1),
 (237, 1),
 (388, 1),
 (549, 1)],
[(7, 1),
 (20, 1),
 (61, 1),
 (79, 1),
 (94, 1),
 (198, 1),
 (211, 1),
 (480, 1),
 (522, 1),
 (622, 1),
 (625, 1),
 (671, 1),
 (756, 1),
 (757, 1),
 (758, 1),
 (759, 1)],
[(7, 1),
 (22, 2),
 (61, 1),
 (62, 1),
 (94, 1),
 (128, 1),
 (139, 1),
 (153, 1),
 (164, 1),
 (192, 1),
 (203, 1),
 (514, 1),
 (760, 1)],
[(16, 1), (39, 1), (133, 1), (203, 1), (211, 1), (243, 2), (678, 1)],
[(49, 1), (79, 1), (94, 1), (139, 1), (179, 1), (197, 1), (515, 1), (761, 1)],
[(6, 2),
 (16, 1),
 (39, 1),
 (47, 1),
 (62, 1),
 (88, 1),
 (93, 1),
 (141, 1),
 (162, 1),
 (243, 1),
 (716, 1),
 (752, 1),
 (762, 1)],
[(92, 1), (197, 1), (262, 1), (343, 1), (651, 1), (763, 1)],
[(62, 1), (197, 1), (343, 1), (500, 1), (764, 1)],
[(16, 1), (35, 1), (172, 1), (262, 1), (477, 2), (765, 1), (766, 1)],
[(19, 1),
 (79, 3),
 (104, 1),
 (105, 2),
 (114, 1),
 (122, 1),
 (155, 1),
 (179, 2),
 (211, 1),
 (343, 1),
 (399, 1),
 (492, 1),
 (620, 1),
 (648, 1),
 (731, 1),
 (767, 1),
 (768, 1),
 (769, 1),
 (770, 1),
 (771, 1),
 (772, 1),
 (773, 1),
 (774, 2),
 (775, 1),
 (776, 1),
 (777, 1)],
[(34, 1),
 (36, 1),
 (178, 1),
 (231, 1),
 (238, 1),
 (262, 1),
 (270, 1),
 (304, 1),
 (373, 1),
 (778, 1)],
[(6, 1),
 (20, 1),
 (26, 1),
 (33, 1),
 (94, 1),
 (115, 1),
 (179, 1),
 (215, 1),
 (288, 1),
 (397, 1),
 (554, 1),
 (635, 1),
 (779, 1),
 (780, 1)],
[(5, 3),
 (6, 3),
 (24, 1),
 (29, 1),
 (33, 1),
 (87, 1),
 (90, 1),
 (93, 1)
```

```
 (92, 1),
 (98, 1),
 (114, 1),
 (128, 1),
 (164, 2),
 (185, 1),
 (188, 1),
 (197, 1),
 (237, 1),
 (262, 1),
 (287, 1),
 (318, 1),
 (385, 1),
 (406, 2),
 (429, 1),
 (470, 1),
 (506, 1),
 (781, 1),
 (782, 1),
 (783, 1)],
[(24, 1), (33, 1), (62, 1), (207, 1), (300, 1), (388, 1), (458, 2)],
[(8, 1),
 (24, 1),
 (33, 1),
 (41, 1),
 (92, 1),
 (94, 1),
 (169, 1),
 (192, 1),
 (395, 1)],
[(62, 1),
 (79, 1),
 (121, 1),
 (139, 1),
 (153, 1),
 (197, 1),
 (354, 1),
 (387, 1)],
[(0, 1),
 (1, 1),
 (16, 1),
 (35, 2),
 (36, 2),
 (65, 1),
 (79, 1),
 (81, 1),
 (105, 1),
 (120, 1),
 (268, 1),
 (296, 1),
 (339, 1),
 (403, 1),
 (477, 1),
 (719, 1),
 (730, 1)],
[(16, 1), (94, 1), (193, 1), (508, 1), (647, 1)],
[(36, 1),
 (45, 1),
 (47, 2),
 (51, 1),
 (57, 1),
 (69, 1),
 (74, 1),
 (81, 1),
 (105, 1),
 (141, 1),
 (209, 1),
 (211, 1),
 (216, 1),
 (231, 1),
 (263, 1),
 (335, 1),
 (492, 1),
 (595, 1),
 (676, 1),
 (691, 1),
 (702, 1),
 (784, 1),
 (785, 1),
 (786, 1),
 (787, 1),
 (788, 1)],
[(61, 1), (79, 1), (167, 1), (192, 1), (651, 1)],
[(16, 1), (211, 1)],
[(6, 1),
 (26, 1),
 (41, 1),
 (153, 1),
 (179, 1),
 (185, 1),
 (192, 1),
 (789, 1),
 (790, 1)],
[(20, 1), (121, 1), (153, 1), (164, 1), (179, 1), (643, 1)],
[(0, 1),
 (71, 1),
 (92, 1),
 (93, 1),
 (161, 1),
 (179, 1),
 (299, 1),
 (343, 1),
 (429, 1),
 (791, 1)],
[(1, 1),
 (16, 1),
 (62, 1),
 (77, 1),
 (85, 1),
 (90, 1),
 (103, 1),
 (176, 1),
 (213, 1),
 (792, 1),
 (793, 1),
 (794, 1)],
```

(795, 1)],
[(16, 1),
(92, 2),
(105, 1),
(113, 1),
(155, 1),
(169, 1),
(193, 1),
(295, 1),
(318, 1),
(343, 1),
(385, 1),
(617, 1),
(662, 1),
(796, 1),
(797, 1)],
[(26, 1), (121, 1), (139, 1), (153, 1), (179, 1), (272, 1)],
[(6, 1),
(15, 2),
(20, 1),
(22, 1),
(26, 1),
(33, 1),
(34, 1),
(62, 2),
(77, 1),
(79, 2),
(83, 1),
(92, 1),
(93, 3),
(96, 1),
(128, 1),
(144, 1),
(154, 1),
(160, 1),
(164, 1),
(173, 1),
(174, 1),
(176, 1),
(197, 1),
(201, 1),
(238, 1),
(308, 1),
(369, 1),
(370, 1),
(429, 1),
(472, 1),
(523, 1),
(611, 1),
(798, 1),
(799, 1),
(800, 1),
(801, 1)],
[(37, 1),
(107, 1),
(114, 1),
(185, 1),
(193, 1),
(343, 2),
(388, 1),
(435, 1),
(458, 1),
(491, 1),
(547, 1),
(777, 1),
(802, 1)],
[(6, 1),
(47, 2),
(66, 1),
(69, 1),
(79, 1),
(80, 1),
(81, 1),
(93, 1),
(105, 4),
(120, 3),
(159, 1),
(168, 1),
(179, 1),
(192, 1),
(223, 1),
(263, 1),
(268, 1),
(335, 1),
(359, 1),
(405, 1),
(406, 1),
(419, 1),
(420, 1),
(752, 1),
(803, 1),
(804, 1),
(805, 1),
(806, 1)],
[(33, 1), (92, 1), (262, 1), (316, 1), (780, 1), (807, 1), (808, 1)],
[(24, 1),
(62, 1),
(168, 1),
(191, 1),
(211, 1),
(388, 1),
(614, 1),
(809, 1)],
[(47, 1), (69, 2), (197, 1), (201, 1), (306, 1), (314, 1)],
[(16, 1),
(35, 1),
(64, 1),
(88, 1),
(93, 1),
(185, 1),
(222, 1),
(243, 1),
(319, 1),
(406, 1),
(411, 1),

```
 (526, 1),
 (793, 1),
 (810, 1)],
[(49, 1), (179, 1), (193, 1), (195, 1), (354, 1), (554, 1)],
[(76, 1),
 (88, 1),
 (102, 1),
 (128, 1),
 (133, 1),
 (162, 1),
 (168, 1),
 (179, 1),
 (193, 2),
 (196, 1),
 (200, 1),
 (237, 1),
 (295, 1),
 (306, 1),
 (343, 1),
 (662, 1),
 (773, 1),
 (785, 1),
 (811, 1)],
[(6, 1), (16, 1), (103, 1), (133, 1), (164, 1), (258, 1), (349, 1), (812, 1)],
[(114, 1), (435, 1), (633, 1), (813, 1), (814, 1)],
[(11, 1), (16, 1), (24, 1), (35, 1), (88, 1), (92, 1), (93, 1)],
[(33, 1),
 (87, 1),
 (94, 1),
 (103, 1),
 (168, 1),
 (194, 1),
 (419, 1),
 (434, 1),
 (478, 1),
 (498, 1),
 (591, 1),
 (628, 1),
 (815, 1),
 (816, 1)],
[(37, 1),
 (64, 1),
 (87, 1),
 (92, 2),
 (105, 1),
 (141, 1),
 (143, 1),
 (193, 1),
 (216, 1),
 (343, 1),
 (388, 1),
 (406, 1),
 (540, 1),
 (601, 1),
 (674, 1),
 (711, 1),
 (806, 1),
 (817, 1),
 (818, 1),
 (819, 1),
 (820, 1),
 (821, 2),
 (822, 1),
 (823, 1)],
[(79, 1),
 (92, 1),
 (105, 2),
 (107, 1),
 (255, 1),
 (300, 1),
 (343, 1),
 (387, 1),
 (624, 1),
 (773, 1),
 (824, 1),
 (825, 1),
 (826, 1)],
[(26, 1),
 (46, 1),
 (47, 2),
 (92, 1),
 (164, 1),
 (181, 1),
 (209, 1),
 (217, 2),
 (249, 1),
 (470, 1),
 (492, 1),
 (572, 1),
 (827, 1),
 (828, 1)],
[(33, 1),
 (62, 1),
 (79, 1),
 (83, 1),
 (197, 1),
 (211, 1),
 (314, 1),
 (825, 1),
 (829, 1)],
[(33, 1), (458, 2), (558, 1), (830, 1)],
[(62, 1),
 (77, 1),
 (126, 1),
 (164, 1),
 (192, 1),
 (740, 1),
 (790, 1),
 (831, 1),
 (832, 1)],
[(37, 1),
 (49, 1),
 (75, 1),
 (103, 1),
 (191, 1),
```

```
 (193, 1),
 (300, 1),
 (374, 1),
 (475, 1),
 (496, 1),
 (820, 1),
 (833, 1)],
[(26, 1), (48, 1), (83, 1), (179, 1), (207, 1), (461, 1)],
[(19, 1),
 (22, 1),
 (33, 2),
 (79, 1),
 (87, 1),
 (92, 1),
 (128, 2),
 (168, 2),
 (169, 1),
 (211, 1),
 (231, 1),
 (243, 1),
 (287, 1),
 (316, 1),
 (331, 1),
 (373, 1),
 (387, 4),
 (434, 1),
 (733, 1),
 (765, 1),
 (834, 1),
 (835, 1),
 (836, 1),
 (837, 1)],
[(35, 1), (133, 1), (179, 2), (219, 1), (265, 1), (296, 1), (478, 1)],
[(11, 1),
 (16, 1),
 (39, 1),
 (83, 2),
 (92, 1),
 (211, 1),
 (222, 1),
 (238, 1),
 (298, 1),
 (607, 1),
 (838, 1),
 (839, 1)],
[(0, 1),
 (87, 1),
 (179, 1),
 (238, 1),
 (243, 1),
 (263, 1),
 (373, 1),
 (457, 1),
 (702, 1),
 (840, 1)],
[(6, 1),
 (16, 1),
 (22, 1),
 (62, 1),
 (65, 1),
 (83, 1),
 (109, 1),
 (155, 1),
 (216, 1),
 (217, 1),
 (367, 1),
 (498, 1),
 (841, 1)],
[(7, 1),
 (13, 1),
 (16, 1),
 (41, 1),
 (94, 1),
 (105, 1),
 (136, 1),
 (139, 1),
 (169, 1),
 (179, 1),
 (181, 1),
 (296, 1),
 (300, 1),
 (628, 1),
 (774, 1),
 (842, 1),
 (843, 1),
 (844, 1)],
[(22, 1),
 (94, 1),
 (128, 1),
 (167, 1),
 (268, 2),
 (524, 1),
 (562, 1),
 (845, 1)],
[(16, 1),
 (79, 1),
 (83, 1),
 (87, 1),
 (93, 1),
 (197, 1),
 (207, 1),
 (388, 1),
 (391, 1)],
[(26, 1),
 (56, 1),
 (68, 1),
 (105, 1),
 (329, 1),
 (580, 1),
 (846, 1),
 (847, 1),
 (848, 1),
 (849, 1)],
[(62, 1),
 (79, 1),
```

```
 (92, 1),
 (97, 1),
 (139, 1),
 (179, 1),
 (193, 1),
 (211, 1),
 (227, 1),
 (850, 1),
 (851, 1),
 (852, 1),
 (853, 1)],
[(3, 1),
 (6, 1),
 (20, 1),
 (21, 1),
 (62, 1),
 (87, 1),
 (92, 1),
 (94, 1),
 (113, 1),
 (127, 1),
 (139, 1),
 (153, 1),
 (163, 1),
 (164, 1),
 (314, 1),
 (387, 1),
 (641, 1),
 (671, 1),
 (854, 1),
 (855, 1),
 (856, 1),
 (857, 1),
 (858, 1)],
[(72, 1),
 (79, 1),
 (88, 1),
 (90, 1),
 (93, 1),
 (121, 1),
 (172, 1),
 (238, 1),
 (249, 1),
 (263, 1),
 (268, 1),
 (325, 1),
 (396, 1),
 (397, 1),
 (465, 1),
 (859, 1),
 (860, 1),
 (861, 1),
 (862, 1),
 (863, 1)],
[(51, 1), (133, 1), (192, 1), (300, 1), (329, 1), (864, 1), (865, 1)],
[(22, 1),
 (49, 3),
 (74, 1),
 (98, 1),
 (114, 3),
 (117, 1),
 (128, 1),
 (179, 2),
 (193, 4),
 (237, 1),
 (262, 1),
 (264, 1),
 (343, 1),
 (535, 1),
 (731, 1),
 (866, 1),
 (867, 1)],
[(16, 1), (93, 1), (154, 1), (203, 1), (343, 1), (868, 1)],
[(79, 1), (94, 1), (152, 1), (177, 1), (178, 1), (346, 1), (869, 1)],
[(16, 1),
 (19, 1),
 (24, 1),
 (26, 1),
 (33, 1),
 (111, 1),
 (336, 1),
 (643, 1),
 (870, 1)],
[(35, 1), (37, 1), (47, 1), (48, 1), (88, 1), (136, 1), (179, 1)],
[(16, 1),
 (24, 1),
 (79, 1),
 (87, 2),
 (94, 1),
 (121, 1),
 (179, 1),
 (197, 1),
 (434, 1),
 (478, 1),
 (535, 1),
 (661, 2),
 (871, 1)],
[(93, 1),
 (121, 1),
 (179, 1),
 (207, 1),
 (272, 1),
 (315, 1),
 (800, 1),
 (872, 1)],
[(16, 1), (48, 1), (141, 1), (179, 1), (191, 1), (262, 2), (446, 1)],
[(19, 1), (26, 1), (79, 1), (197, 1), (268, 1), (387, 1), (399, 1), (663, 1)],
[(7, 1),
 (33, 1),
 (49, 1),
 (62, 1),
 (193, 1),
 (194, 1),
 (340, 1),
 (342, 1),
 (342, 1)
```

```
 (345, 1),
 (873, 1)],
[(4, 1),
 (26, 1),
 (41, 1),
 (46, 1),
 (47, 1),
 (56, 1),
 (79, 1),
 (105, 1),
 (177, 1),
 (179, 1),
 (192, 1),
 (296, 1),
 (691, 1),
 (777, 1),
 (806, 1),
 (874, 1)],
[(16, 2),
 (24, 1),
 (33, 1),
 (94, 1),
 (133, 1),
 (177, 1),
 (178, 1),
 (496, 1),
 (875, 1),
 (876, 1)],
[(13, 1),
 (16, 1),
 (20, 1),
 (26, 1),
 (47, 2),
 (62, 1),
 (71, 1),
 (79, 1),
 (81, 1),
 (87, 1),
 (88, 1),
 (136, 1),
 (226, 1),
 (229, 1),
 (238, 1),
 (283, 1),
 (292, 1),
 (316, 1),
 (329, 1),
 (357, 1),
 (387, 1),
 (393, 1),
 (427, 1),
 (434, 1),
 (841, 1),
 (877, 1),
 (878, 1),
 (879, 1),
 (880, 1),
 (881, 1),
 (882, 1)],
[(51, 1),
 (179, 1),
 (743, 1),
 (774, 1),
 (883, 1),
 (884, 1),
 (885, 1),
 (886, 1)],
[(33, 1),
 (79, 1),
 (94, 1),
 (211, 1),
 (510, 2),
 (538, 1),
 (887, 1),
 (888, 1),
 (889, 1),
 (890, 1),
 (891, 1)],
[(93, 1), (176, 1), (179, 1), (247, 1), (603, 1), (892, 1)],
[(11, 1),
 (22, 1),
 (64, 1),
 (65, 1),
 (139, 1),
 (222, 1),
 (359, 1),
 (411, 1),
 (421, 1),
 (515, 1),
 (750, 1),
 (777, 1),
 (788, 1),
 (893, 1)],
[(37, 1),
 (70, 1),
 (94, 1),
 (103, 1),
 (150, 1),
 (164, 1),
 (173, 1),
 (237, 1),
 (388, 1),
 (536, 1),
 (562, 1),
 (799, 1),
 (894, 1),
 (895, 1),
 (896, 1)],
[(20, 1), (26, 1), (160, 2), (263, 1), (540, 1), (634, 1)],
[(1, 1),
 (16, 2),
 (22, 1),
 (29, 1),
 (33, 1),
 (34, 1),
 (51, 1),
```

```
 (91, 1),
 (79, 1),
 (92, 1),
 (143, 1),
 (150, 1),
 (164, 1),
 (173, 1),
 (185, 1),
 (211, 1),
 (252, 1),
 (318, 1),
 (427, 1),
 (470, 1),
 (489, 1),
 (525, 1),
 (527, 1),
 (592, 1),
 (613, 1),
 (765, 1),
 (788, 1),
 (897, 1),
 (898, 1),
 (899, 1),
 (900, 1),
 (901, 1)],
[(16, 1),
 (35, 1),
 (47, 1),
 (52, 1),
 (65, 1),
 (69, 1),
 (179, 1),
 (217, 1),
 (292, 1),
 (306, 1),
 (902, 1)],
[(6, 1),
 (11, 1),
 (16, 1),
 (56, 1),
 (68, 2),
 (79, 1),
 (113, 1),
 (197, 1),
 (201, 1),
 (211, 1),
 (281, 1),
 (640, 1),
 (666, 1),
 (691, 2),
 (903, 1),
 (904, 1),
 (905, 1),
 (906, 1),
 (907, 1)],
[(37, 1),
 (94, 1),
 (164, 2),
 (177, 1),
 (211, 1),
 (238, 1),
 (263, 1),
 (304, 1),
 (667, 1),
 (724, 1),
 (799, 1),
 (908, 1),
 (909, 1)],
[(16, 1), (24, 2), (83, 1), (92, 2), (94, 1), (211, 1), (409, 1)],
[(47, 1), (79, 1), (136, 1), (179, 1), (197, 1), (201, 1)],
[(20, 1),
 (29, 1),
 (62, 1),
 (92, 1),
 (94, 1),
 (125, 1),
 (155, 1),
 (167, 1),
 (179, 1),
 (197, 1),
 (388, 1),
 (390, 1),
 (401, 1),
 (485, 1),
 (718, 1),
 (821, 1),
 (825, 1),
 (826, 1),
 (910, 1),
 (911, 1)],
[(16, 1),
 (26, 1),
 (35, 2),
 (47, 1),
 (69, 1),
 (197, 1),
 (263, 2),
 (281, 1),
 (299, 1),
 (702, 2),
 (912, 1)],
[(41, 1), (94, 1), (121, 1), (179, 1), (256, 1), (405, 1), (479, 1)],
[(13, 1),
 (25, 1),
 (49, 1),
 (87, 1),
 (139, 1),
 (141, 1),
 (193, 1),
 (258, 1),
 (388, 2)],
[(16, 1), (33, 1), (79, 1), (197, 1), (207, 1), (586, 1)],
[(16, 1),
 (24, 1),
 (79, 1),
 (94, 1),
```

```
 (112, 1),
 (197, 1),
 (591, 1),
 (664, 1),
 (765, 1)],
[(92, 2), (176, 1), (262, 1), (587, 1)],
[(94, 1), (114, 1), (140, 1), (343, 1), (603, 1), (913, 1), (914, 1)],
[(33, 1),
 (37, 1),
 (62, 1),
 (79, 1),
 (114, 1),
 (303, 1),
 (343, 1),
 (352, 1),
 (591, 1),
 (915, 1)],
[(16, 1),
 (20, 1),
 (33, 1),
 (64, 1),
 (80, 1),
 (92, 1),
 (108, 1),
 (139, 1),
 (179, 1),
 (296, 1)],
[(16, 2),
 (22, 1),
 (23, 1),
 (24, 1),
 (33, 2),
 (87, 1),
 (152, 1),
 (164, 1),
 (197, 1),
 (331, 1),
 (346, 1),
 (381, 1),
 (410, 1),
 (526, 1),
 (562, 1),
 (583, 1),
 (591, 1),
 (664, 1),
 (845, 1),
 (916, 1),
 (917, 1),
 (918, 1),
 (919, 1)],
[(6, 1),
 (33, 2),
 (49, 1),
 (79, 1),
 (83, 1),
 (92, 1),
 (181, 1),
 (492, 1),
 (537, 1),
 (550, 1),
 (844, 1),
 (920, 1),
 (921, 1),
 (922, 1)],
[(16, 1),
 (79, 2),
 (92, 1),
 (93, 1),
 (114, 1),
 (128, 1),
 (133, 2),
 (169, 1),
 (197, 1),
 (200, 1),
 (209, 1),
 (798, 1),
 (875, 1),
 (923, 1),
 (924, 1),
 (925, 1),
 (926, 1)],
[(20, 1),
 (35, 1),
 (47, 1),
 (64, 1),
 (94, 1),
 (105, 1),
 (164, 1),
 (179, 1),
 (263, 1),
 (325, 1),
 (595, 1),
 (653, 1)],
[(11, 1),
 (52, 1),
 (64, 1),
 (94, 2),
 (121, 1),
 (139, 1),
 (153, 1),
 (179, 2),
 (419, 1)],
[(20, 1),
 (70, 1),
 (90, 1),
 (92, 1),
 (94, 1),
 (125, 1),
 (139, 1),
 (153, 1),
 (164, 1),
 (179, 1),
 (380, 1),
 (420, 1),
 (512, 1)],
```

```
[(16, 1),
 (35, 1),
 (114, 2),
 (161, 1),
 (193, 1),
 (217, 1),
 (264, 1),
 (358, 1),
 (406, 1),
 (511, 1),
 (897, 1)],
[(7, 1),
 (16, 1),
 (33, 1),
 (79, 1),
 (133, 1),
 (168, 1),
 (176, 1),
 (197, 1),
 (211, 1),
 (249, 1),
 (387, 1),
 (812, 1),
 (813, 1),
 (927, 1),
 (928, 1)],
[(16, 1),
 (24, 1),
 (61, 1),
 (79, 3),
 (133, 1),
 (150, 1),
 (164, 1),
 (196, 1),
 (211, 1),
 (288, 1),
 (389, 1),
 (458, 1),
 (477, 1),
 (510, 1),
 (813, 1),
 (831, 1),
 (929, 1)],
[(0, 1),
 (62, 1),
 (80, 1),
 (83, 1),
 (87, 1),
 (90, 1),
 (140, 1),
 (150, 1),
 (185, 1),
 (238, 2),
 (343, 1),
 (930, 1),
 (931, 1),
 (932, 2),
 (933, 1)],
[(41, 1),
 (51, 1),
 (53, 1),
 (83, 1),
 (87, 1),
 (92, 1),
 (99, 1),
 (165, 1),
 (168, 1),
 (179, 1),
 (192, 1),
 (248, 1),
 (358, 1),
 (374, 1),
 (388, 1),
 (534, 1),
 (587, 1),
 (615, 1),
 (643, 1),
 (934, 1),
 (935, 1)],
[(0, 1),
 (22, 1),
 (33, 2),
 (92, 1),
 (95, 1),
 (139, 1),
 (146, 1),
 (153, 1),
 (168, 1),
 (179, 1),
 (195, 1),
 (544, 1),
 (629, 1),
 (936, 1),
 (937, 1),
 (938, 1)],
[(16, 1),
 (22, 1),
 (24, 1),
 (112, 1),
 (114, 1),
 (128, 1),
 (139, 1),
 (153, 1),
 (160, 1),
 (164, 1),
 (343, 1)],
[(80, 1), (91, 1), (197, 1), (224, 1), (283, 1)],
[(11, 1),
 (24, 1),
 (96, 1),
 (131, 1),
 (143, 1),
 (161, 1),
 (164, 1),
 (283, 1),
```

```
 (388, 1),
 (939, 1),
 (940, 1),
 (941, 1),
 (942, 1),
 (943, 1),
 (944, 1)],
[(24, 1),
 (52, 1),
 (62, 1),
 (92, 1),
 (93, 1),
 (262, 2),
 (343, 1),
 (435, 1),
 (448, 1),
 (603, 1)],
[(103, 1),
 (128, 1),
 (133, 1),
 (179, 2),
 (193, 1),
 (349, 1),
 (512, 1),
 (513, 1)],
[(52, 1),
 (62, 1),
 (79, 1),
 (80, 1),
 (86, 1),
 (93, 1),
 (94, 1),
 (165, 2),
 (171, 1),
 (197, 1),
 (249, 1),
 (309, 2),
 (345, 1),
 (388, 1),
 (514, 1),
 (660, 1),
 (664, 1),
 (670, 1),
 (742, 1),
 (945, 1),
 (946, 1),
 (947, 1),
 (948, 1)],
[(26, 1), (160, 1), (628, 1), (710, 1), (740, 1)],
[(16, 1), (85, 1), (176, 1), (201, 1), (742, 1), (949, 1)],
[(20, 1),
 (37, 1),
 (49, 1),
 (80, 1),
 (94, 2),
 (193, 1),
 (216, 1),
 (653, 1),
 (788, 1)],
[(1, 1),
 (15, 2),
 (16, 1),
 (22, 2),
 (24, 1),
 (33, 1),
 (90, 2),
 (92, 1),
 (95, 1),
 (109, 1),
 (133, 2),
 (155, 2),
 (173, 1),
 (183, 1),
 (200, 2),
 (203, 1),
 (347, 1),
 (369, 1),
 (777, 1),
 (950, 1),
 (951, 1),
 (952, 1)],
[(6, 1),
 (49, 1),
 (79, 1),
 (133, 1),
 (168, 1),
 (197, 1),
 (212, 1),
 (268, 1),
 (387, 2),
 (802, 1),
 (953, 1),
 (954, 1)],
[(6, 1),
 (8, 1),
 (16, 1),
 (29, 1),
 (33, 1),
 (252, 1),
 (258, 1),
 (277, 1),
 (381, 1),
 (395, 1),
 (418, 1),
 (434, 1),
 (603, 1),
 (847, 1),
 (900, 1),
 (955, 1)],
[(6, 1),
 (35, 2),
 (46, 1),
 (47, 2),
 (64, 1),
 (78, 1),
 (88, 1)
```

```
 (88, 1),
 (127, 1),
 (153, 1),
 (179, 1),
 (302, 1),
 (439, 1),
 (479, 1),
 (956, 1),
 (957, 1)],
[(16, 1), (167, 1), (333, 1), (958, 1), (959, 1)],
[(33, 1), (121, 1), (168, 1), (211, 1), (248, 1), (734, 1)],
[(133, 1),
 (169, 1),
 (227, 1),
 (231, 1),
 (239, 1),
 (262, 1),
 (458, 1),
 (765, 1)],
[(33, 1), (139, 1), (157, 1), (300, 1), (960, 1)],
[(19, 1), (24, 1), (87, 1), (231, 1), (237, 1), (255, 1), (306, 1)],
[(20, 1),
 (29, 1),
 (33, 1),
 (37, 2),
 (45, 1),
 (49, 1),
 (79, 1),
 (92, 1),
 (94, 1),
 (95, 1),
 (155, 1),
 (164, 1),
 (169, 1),
 (225, 1),
 (230, 1),
 (240, 1),
 (316, 1),
 (338, 1),
 (435, 1),
 (479, 1),
 (651, 1),
 (703, 1),
 (765, 1),
 (961, 1),
 (962, 1),
 (963, 1)],
[(0, 1),
 (6, 1),
 (13, 1),
 (16, 1),
 (24, 1),
 (37, 2),
 (52, 1),
 (64, 2),
 (77, 1),
 (79, 1),
 (80, 1),
 (83, 1),
 (86, 1),
 (89, 2),
 (105, 1),
 (134, 1),
 (141, 1),
 (150, 1),
 (230, 1),
 (232, 1),
 (235, 1),
 (295, 1),
 (318, 2),
 (343, 2),
 (353, 1),
 (385, 2),
 (426, 1),
 (485, 1),
 (486, 1),
 (487, 2),
 (488, 1),
 (489, 1),
 (490, 1),
 (491, 2),
 (492, 1),
 (493, 1),
 (494, 1),
 (495, 1),
 (496, 1),
 (497, 1),
 (498, 1),
 (499, 1),
 (500, 1),
 (501, 1),
 (502, 1),
 (503, 1),
 (504, 1),
 (505, 1),
 (506, 1),
 (507, 1),
 (508, 3),
 (509, 1),
 (510, 2)],
[(6, 1),
 (26, 1),
 (33, 1),
 (62, 1),
 (94, 1),
 (128, 1),
 (133, 2),
 (168, 1),
 (192, 1),
 (325, 1),
 (344, 1),
 (434, 1),
 (703, 3),
 (964, 1),
 (965, 2),
 (966, 1),
```

```
 (968, 1),
 (967, 1)],
[(16, 1), (93, 1), (133, 1), (354, 1), (494, 1), (968, 1)],
[(6, 1),
 (7, 1),
 (16, 2),
 (33, 3),
 (52, 2),
 (103, 1),
 (150, 2),
 (164, 2),
 (168, 1),
 (184, 1),
 (201, 1),
 (211, 2),
 (216, 1),
 (281, 1),
 (342, 4),
 (369, 1),
 (482, 1),
 (496, 1),
 (525, 1),
 (539, 1),
 (727, 1),
 (729, 1),
 (739, 1),
 (757, 1),
 (806, 1),
 (924, 4),
 (954, 1),
 (969, 1),
 (970, 1),
 (971, 1),
 (972, 1)],
[(79, 1),
 (153, 1),
 (168, 1),
 (179, 2),
 (191, 1),
 (193, 1),
 (219, 1),
 (226, 1),
 (711, 1),
 (771, 1)],
[(16, 1),
 (33, 1),
 (36, 1),
 (79, 1),
 (83, 1),
 (110, 1),
 (117, 1),
 (139, 1),
 (168, 1),
 (179, 1),
 (180, 1),
 (195, 1),
 (207, 1),
 (391, 1),
 (515, 1),
 (762, 1),
 (765, 1),
 (973, 1)],
[(5, 1),
 (13, 1),
 (19, 1),
 (24, 1),
 (64, 2),
 (90, 1),
 (139, 1),
 (164, 1),
 (193, 1),
 (306, 1),
 (333, 1),
 (772, 1)],
[(26, 1), (79, 1), (179, 1), (197, 1), (645, 1)],
[(22, 1), (92, 1), (93, 1), (176, 1), (510, 1), (671, 1), (780, 1)],
[(79, 1),
 (179, 1),
 (192, 1),
 (197, 1),
 (249, 1),
 (494, 1),
 (636, 1),
 (974, 1)],
[(16, 2), (24, 1), (93, 1), (94, 1), (121, 1), (179, 1)],
[(19, 1),
 (35, 2),
 (71, 1),
 (80, 1),
 (92, 1),
 (93, 1),
 (133, 1),
 (155, 1),
 (222, 1),
 (231, 3),
 (286, 1),
 (329, 1),
 (474, 1),
 (485, 1),
 (549, 1),
 (975, 1),
 (976, 1),
 (977, 1),
 (978, 1)],
[(16, 1),
 (22, 1),
 (35, 2),
 (46, 2),
 (51, 1),
 (65, 1),
 (66, 1),
 (71, 1),
 (92, 1),
 (159, 1),
 (161, 1),
 (164, 1),
```

```
 (192, 1),
 (262, 1),
 (325, 1),
 (343, 1),
 (435, 1),
 (608, 1),
 (634, 1)],
[(33, 1), (57, 1), (87, 1), (179, 1), (255, 1), (270, 1), (343, 1), (434, 1)],
[(41, 1),
 (79, 1),
 (94, 1),
 (98, 1),
 (128, 1),
 (160, 1),
 (192, 1),
 (197, 1),
 (300, 1)],
[(47, 1), (79, 1), (88, 1), (90, 1), (179, 1), (300, 1), (420, 1)],
[(16, 1), (20, 1), (94, 1), (164, 1), (166, 1), (211, 1), (227, 1)],
[(11, 1),
 (20, 2),
 (49, 1),
 (52, 1),
 (64, 1),
 (87, 1),
 (93, 1),
 (94, 2),
 (127, 1),
 (133, 1),
 (139, 1),
 (153, 1),
 (157, 1),
 (179, 2),
 (181, 1),
 (193, 1),
 (446, 1),
 (553, 1),
 (956, 1),
 (979, 1)],
[(16, 1), (35, 1), (48, 1), (90, 1), (211, 1), (300, 1), (420, 1)],
[(4, 1),
 (45, 1),
 (87, 1),
 (143, 1),
 (211, 1),
 (224, 1),
 (248, 1),
 (387, 1),
 (409, 1),
 (734, 1),
 (773, 1),
 (791, 1),
 (956, 1),
 (980, 1),
 (981, 1)],
[(19, 1),
 (22, 1),
 (24, 1),
 (29, 1),
 (37, 1),
 (94, 2),
 (164, 1),
 (188, 1),
 (200, 1),
 (203, 1),
 (262, 1),
 (287, 1),
 (409, 1),
 (410, 1),
 (577, 1),
 (689, 2),
 (982, 1),
 (983, 1),
 (984, 1),
 (985, 1),
 (986, 1),
 (987, 1),
 (988, 1)],
[(92, 1), (103, 1)],
[(0, 3),
 (8, 2),
 (11, 1),
 (19, 4),
 (20, 4),
 (21, 1),
 (24, 1),
 (33, 1),
 (34, 1),
 (45, 6),
 (52, 4),
 (62, 1),
 (64, 1),
 (74, 1),
 (77, 1),
 (78, 1),
 (79, 1),
 (83, 9),
 (85, 2),
 (87, 1),
 (89, 3),
 (90, 2),
 (94, 4),
 (98, 1),
 (103, 3),
 (104, 1),
 (105, 3),
 (114, 8),
 (117, 1),
 (127, 1),
 (131, 1),
 (154, 3),
 (164, 1),
 (168, 1),
 (185, 3),
 (191, 2),
```

(193, 13),
(197, 3),
(201, 3),
(211, 1),
(216, 1),
(221, 2),
(225, 2),
(238, 1),
(239, 1),
(244, 1),
(248, 1),
(249, 5),
(256, 3),
(293, 1),
(299, 1),
(306, 2),
(310, 1),
(316, 1),
(318, 1),
(319, 1),
(324, 1),
(325, 1),
(336, 1),
(343, 12),
(344, 1),
(353, 1),
(374, 14),
(385, 4),
(387, 1),
(388, 1),
(400, 1),
(411, 1),
(420, 1),
(426, 1),
(458, 1),
(479, 2),
(487, 7),
(490, 2),
(492, 1),
(494, 1),
(496, 4),
(499, 1),
(500, 1),
(510, 1),
(516, 1),
(536, 1),
(553, 1),
(587, 4),
(591, 1),
(592, 1),
(595, 3),
(608, 1),
(615, 5),
(616, 2),
(643, 5),
(658, 1),
(659, 5),
(667, 1),
(717, 1),
(720, 2),
(732, 1),
(734, 4),
(747, 3),
(759, 1),
(772, 3),
(793, 1),
(802, 1),
(804, 1),
(806, 1),
(816, 1),
(825, 1),
(833, 1),
(848, 2),
(866, 2),
(916, 1),
(932, 1),
(950, 1),
(986, 1),
(989, 1),
(990, 4),
(991, 1),
(992, 1),
(993, 1),
(994, 1),
(995, 1),
(996, 1),
(997, 1),
(998, 2),
(999, 1),
(1000, 1),
(1001, 1),
(1002, 1),
(1003, 1),
(1004, 1),
(1005, 1),
(1006, 1),
(1007, 1),
(1008, 1),
(1009, 1),
(1010, 1),
(1011, 1),
(1012, 1),
(1013, 9),
(1014, 1),
(1015, 1),
(1016, 1),
(1017, 1),
(1018, 1),
(1019, 1),
(1020, 1),
(1021, 1),
(1022, 3),
(1023, 3),
(1024, 1),
(1025, 2),

```
 (1026, 3),
 (1027, 1),
 (1028, 1),
 (1029, 1),
 (1030, 1),
 (1031, 3),
 (1032, 2),
 (1033, 2)],
[(5, 1),
 (49, 1),
 (79, 1),
 (136, 1),
 (179, 1),
 (193, 1),
 (197, 1),
 (318, 1),
 (385, 1)],
[(16, 1),
 (24, 1),
 (33, 1),
 (94, 2),
 (136, 1),
 (139, 1),
 (179, 1),
 (515, 1),
 (636, 1)],
[(16, 1),
 (49, 1),
 (79, 1),
 (167, 1),
 (179, 1),
 (192, 1),
 (193, 1),
 (194, 1),
 (197, 1),
 (388, 1),
 (458, 1),
 (914, 1),
 (1034, 1)],
[(7, 1), (19, 2), (99, 1)],
[(16, 1),
 (35, 1),
 (37, 1),
 (47, 1),
 (69, 1),
 (79, 1),
 (103, 1),
 (250, 1),
 (406, 1),
 (651, 1),
 (793, 1),
 (848, 1),
 (975, 1),
 (1021, 1),
 (1035, 1)],
[(64, 1), (78, 1), (94, 1), (133, 1), (179, 1), (192, 1)],
[(62, 1),
 (64, 1),
 (87, 1),
 (92, 1),
 (165, 1),
 (193, 1),
 (197, 1),
 (343, 1),
 (387, 1),
 (825, 1)],
[(26, 1), (131, 1), (143, 1), (384, 1), (660, 1)],
[(16, 1),
 (64, 1),
 (77, 1),
 (83, 1),
 (87, 1),
 (88, 1),
 (92, 1),
 (146, 1),
 (161, 1),
 (211, 1)],
[(24, 1), (100, 1), (112, 1), (164, 1), (179, 1), (272, 1)],
[(22, 1), (26, 1), (179, 1), (645, 1), (1036, 1)],
[(6, 1),
 (51, 2),
 (64, 1),
 (81, 1),
 (93, 1),
 (121, 1),
 (160, 2),
 (168, 1),
 (177, 1),
 (179, 1),
 (185, 1),
 (192, 1),
 (200, 1),
 (211, 1),
 (222, 1),
 (238, 1),
 (259, 1),
 (312, 2),
 (335, 1),
 (367, 2),
 (402, 1),
 (429, 1),
 (433, 1),
 (594, 1),
 (603, 1),
 (674, 1),
 (1029, 1),
 (1037, 1),
 (1038, 1),
 (1039, 1),
 (1040, 1),
 (1041, 1),
 (1042, 1),
 (1043, 1)],
[(6, 1),
 (35, 1),
 (40, 1)
```

```
 (49, 1),
 (56, 1),
 (76, 1),
 (79, 2),
 (93, 1),
 (103, 1),
 (105, 1),
 (127, 1),
 (179, 1),
 (197, 1),
 (235, 1),
 (312, 1),
 (440, 1),
 (492, 1),
 (691, 1),
 (1044, 1),
 (1045, 1),
 (1046, 1),
 (1047, 1)],
[(24, 1), (86, 1), (92, 1), (99, 1), (192, 1), (406, 1)],
[(7, 1), (62, 1), (92, 1), (309, 1), (945, 1), (1048, 1)],
[(16, 1),
 (19, 1),
 (93, 1),
 (238, 1),
 (399, 1),
 (435, 1),
 (535, 1),
 (549, 1),
 (906, 1),
 (1049, 1)],
[(16, 1),
 (47, 1),
 (62, 1),
 (80, 1),
 (105, 1),
 (168, 1),
 (211, 1),
 (262, 1),
 (286, 1),
 (306, 1),
 (387, 2),
 (419, 1)],
[(11, 1),
 (79, 1),
 (88, 1),
 (106, 1),
 (200, 1),
 (202, 1),
 (203, 1),
 (343, 1),
 (360, 1),
 (510, 1),
 (615, 1),
 (665, 1),
 (845, 1),
 (1050, 1),
 (1051, 1),
 (1052, 1),
 (1053, 1),
 (1054, 1),
 (1055, 1)],
[(83, 1), (92, 1), (93, 1), (207, 1), (892, 1), (1049, 1)],
[(16, 1),
 (79, 1),
 (90, 2),
 (104, 1),
 (116, 1),
 (306, 1),
 (435, 2),
 (1056, 1)],
[(6, 1), (314, 1), (511, 1), (630, 2), (900, 1), (1057, 1), (1058, 1)],
[(26, 1),
 (49, 2),
 (64, 2),
 (77, 1),
 (79, 3),
 (83, 1),
 (92, 1),
 (93, 1),
 (109, 1),
 (114, 1),
 (185, 1),
 (193, 2),
 (318, 1),
 (343, 2),
 (347, 1),
 (348, 1),
 (385, 1),
 (496, 1),
 (906, 2),
 (932, 1),
 (1059, 1),
 (1060, 1),
 (1061, 1)],
[(6, 1),
 (33, 1),
 (92, 1),
 (141, 1),
 (173, 1),
 (184, 1),
 (596, 1),
 (1062, 1),
 (1063, 1),
 (1064, 1)],
[(11, 2),
 (20, 1),
 (24, 1),
 (39, 1),
 (53, 1),
 (77, 1),
 (80, 1),
 (94, 2),
 (121, 1),
 (165, 3),
 (181, 1),
```

```
 (181, 1),
 (195, 1),
 (571, 1),
 (757, 1),
 (1065, 1)],
[(7, 1),
 (16, 1),
 (104, 1),
 (108, 1),
 (211, 1),
 (496, 1),
 (529, 1),
 (875, 1),
 (876, 2),
 (1066, 1)],
[(92, 1), (120, 1), (142, 1), (302, 1), (314, 1), (406, 1)],
[(35, 2),
 (51, 1),
 (88, 1),
 (216, 1),
 (457, 1),
 (791, 1),
 (1067, 1),
 (1068, 1),
 (1069, 1),
 (1070, 1),
 (1071, 1)],
[(1, 1),
 (6, 1),
 (15, 1),
 (16, 1),
 (22, 1),
 (33, 2),
 (34, 1),
 (121, 1),
 (179, 1),
 (211, 1),
 (344, 1),
 (510, 1),
 (559, 1),
 (624, 1),
 (901, 1),
 (1072, 1),
 (1073, 1)],
[(16, 1),
 (52, 1),
 (114, 1),
 (197, 1),
 (252, 1),
 (286, 1),
 (774, 1),
 (779, 1),
 (924, 1),
 (1074, 1)],
[(16, 1),
 (47, 3),
 (69, 1),
 (88, 1),
 (93, 1),
 (179, 1),
 (211, 1),
 (787, 1),
 (1075, 1),
 (1076, 1),
 (1077, 1)],
[(20, 1),
 (37, 1),
 (94, 1),
 (105, 1),
 (193, 1),
 (434, 1),
 (469, 1),
 (777, 1),
 (1078, 1)],
[(33, 1), (128, 1), (231, 1), (249, 1), (1052, 1), (1055, 1)],
[(33, 1), (92, 1), (121, 1), (195, 1), (1079, 1)],
[(127, 1),
 (179, 1),
 (193, 1),
 (270, 1),
 (302, 1),
 (458, 1),
 (775, 1),
 (1080, 1)],
[(133, 1), (179, 1), (247, 1), (302, 1)],
[(4, 1),
 (20, 2),
 (43, 1),
 (79, 1),
 (127, 1),
 (136, 1),
 (179, 1),
 (191, 1),
 (193, 1),
 (197, 1),
 (198, 1),
 (268, 1),
 (324, 1),
 (406, 1),
 (478, 1),
 (1081, 1),
 (1082, 1),
 (1083, 1),
 (1084, 1)],
[(22, 1),
 (34, 1),
 (35, 2),
 (36, 1),
 (46, 1),
 (47, 2),
 (52, 1),
 (61, 1),
 (62, 1),
 (65, 1),
 (71, 1),
 (80, 1),
```

```
 (88, 1),
 (154, 1),
 (197, 1),
 (263, 1),
 (292, 1),
 (305, 2),
 (316, 1),
 (349, 1),
 (411, 1),
 (418, 1),
 (549, 1),
 (583, 1),
 (587, 1),
 (690, 1),
 (740, 1),
 (841, 1),
 (890, 1),
 (982, 1),
 (984, 1),
 (987, 1),
 (1085, 1),
 (1086, 1)],
[(0, 1),
 (49, 1),
 (103, 1),
 (114, 1),
 (136, 1),
 (179, 2),
 (193, 1),
 (731, 1),
 (1087, 1),
 (1088, 1),
 (1089, 1)],
[(0, 1), (47, 1), (69, 1), (105, 1), (120, 1), (262, 1), (1090, 1)],
[(0, 1),
 (16, 1),
 (20, 1),
 (22, 1),
 (26, 1),
 (62, 1),
 (79, 1),
 (80, 1),
 (83, 1),
 (87, 1),
 (89, 1),
 (92, 1),
 (116, 1),
 (146, 1),
 (155, 1),
 (174, 1),
 (197, 1),
 (201, 2),
 (211, 1),
 (217, 2),
 (231, 1),
 (234, 3),
 (244, 1),
 (250, 1),
 (340, 1),
 (342, 1),
 (359, 1),
 (367, 1),
 (396, 1),
 (397, 1),
 (452, 1),
 (479, 1),
 (591, 1),
 (732, 2),
 (777, 1),
 (863, 1),
 (1091, 1),
 (1092, 1),
 (1093, 1),
 (1094, 1)],
[(47, 1),
 (71, 1),
 (88, 1),
 (105, 1),
 (120, 1),
 (243, 1),
 (260, 1),
 (291, 1),
 (292, 1),
 (296, 1),
 (477, 1),
 (478, 1),
 (1095, 1),
 (1096, 1)],
[(22, 1),
 (48, 1),
 (52, 1),
 (62, 1),
 (65, 1),
 (80, 1),
 (93, 1),
 (179, 1),
 (325, 1),
 (411, 1),
 (662, 1),
 (1025, 1),
 (1088, 1),
 (1097, 1),
 (1098, 1)],
[(16, 2), (24, 1), (33, 1), (83, 1), (207, 1), (315, 1), (388, 1)],
[(26, 1),
 (79, 1),
 (114, 1),
 (147, 1),
 (185, 1),
 (200, 1),
 (300, 1),
 (349, 1),
 (535, 1)],
[(5, 1),
 (16, 1),
```

```
 (24, 1),
 (37, 1),
 (52, 1),
 (58, 1),
 (90, 1),
 (106, 1),
 (121, 1),
 (143, 1),
 (179, 2),
 (197, 1),
 (201, 1),
 (203, 1),
 (220, 1),
 (300, 1),
 (316, 1),
 (317, 1),
 (342, 1),
 (343, 1),
 (354, 1),
 (373, 1),
 (406, 1),
 (410, 1),
 (554, 1),
 (772, 1),
 (834, 1),
 (914, 1),
 (1099, 1)],
[(26, 1), (139, 1), (153, 1), (179, 1), (636, 1)],
[(93, 1), (207, 1), (245, 1), (246, 1), (419, 1)],
[(7, 1),
 (19, 1),
 (22, 3),
 (33, 2),
 (83, 1),
 (90, 1),
 (94, 1),
 (108, 2),
 (109, 1),
 (114, 1),
 (128, 2),
 (141, 1),
 (168, 1),
 (172, 1),
 (179, 2),
 (230, 1),
 (295, 1),
 (343, 1),
 (374, 1),
 (510, 1),
 (658, 1),
 (679, 1),
 (875, 1),
 (876, 1),
 (902, 1),
 (1100, 1),
 (1101, 1),
 (1102, 1)],
[(62, 1),
 (105, 1),
 (107, 1),
 (197, 1),
 (487, 1),
 (992, 1),
 (1103, 1),
 (1104, 1),
 (1105, 1)],
[(51, 1),
 (80, 1),
 (87, 1),
 (103, 1),
 (120, 1),
 (164, 1),
 (179, 1),
 (240, 1),
 (241, 1),
 (496, 1),
 (587, 1),
 (1106, 1),
 (1107, 1),
 (1108, 2)],
[(16, 1), (35, 1), (66, 1), (71, 1), (161, 1), (164, 1), (268, 1), (1109, 1)],
[(152, 1), (263, 1), (346, 1), (458, 1), (558, 1), (1110, 1)],
[(6, 1),
 (11, 1),
 (33, 1),
 (103, 1),
 (127, 1),
 (139, 1),
 (153, 1),
 (197, 1),
 (458, 1),
 (554, 1),
 (611, 1)],
[(6, 1),
 (16, 2),
 (48, 1),
 (93, 1),
 (420, 1),
 (472, 1),
 (664, 1),
 (812, 1),
 (1111, 1)],
[(8, 1),
 (16, 2),
 (24, 1),
 (26, 1),
 (33, 1),
 (94, 1),
 (179, 1),
 (211, 1),
 (388, 1),
 (395, 1),
 (405, 1),
 (458, 1),
 (1112, 1)],
```

```
[(7, 1), (24, 1), (33, 1), (70, 1), (172, 1), (627, 1), (1113, 1)],
[(21, 1),
 (24, 2),
 (37, 1),
 (113, 1),
 (236, 1),
 (316, 2),
 (343, 1),
 (347, 1),
 (491, 1),
 (507, 1),
 (508, 1),
 (540, 1),
 (603, 1),
 (630, 1),
 (774, 1),
 (1114, 1),
 (1115, 1),
 (1116, 1),
 (1117, 1)],
[(20, 1),
 (41, 1),
 (74, 1),
 (79, 1),
 (90, 1),
 (121, 1),
 (164, 1),
 (177, 1),
 (179, 1),
 (192, 1),
 (263, 1),
 (396, 1),
 (806, 1),
 (1087, 1),
 (1118, 1),
 (1119, 1)],
[(26, 1), (79, 1), (93, 1), (192, 1), (777, 1)],
[(6, 1), (47, 1), (53, 1), (179, 1), (302, 1), (902, 1), (1120, 1)],
[(18, 1), (33, 1), (160, 1), (179, 1), (244, 1), (512, 1)],
[(6, 1),
 (19, 2),
 (33, 1),
 (79, 1),
 (144, 1),
 (185, 1),
 (211, 2),
 (262, 2),
 (532, 1),
 (535, 1),
 (989, 1),
 (1121, 1),
 (1122, 1)],
[(159, 2), (179, 1), (363, 1), (572, 1), (1037, 1)],
[(24, 1),
 (47, 1),
 (154, 1),
 (160, 1),
 (181, 2),
 (234, 1),
 (243, 1),
 (360, 1),
 (561, 1),
 (577, 1),
 (614, 1),
 (660, 1),
 (1091, 1),
 (1092, 1)],
[(150, 1),
 (173, 1),
 (193, 1),
 (197, 1),
 (343, 1),
 (433, 1),
 (489, 1),
 (649, 1),
 (922, 1),
 (1023, 1),
 (1033, 2),
 (1123, 2)],
[(16, 1), (33, 1), (94, 1), (331, 1), (525, 1), (526, 1)],
[(139, 1), (153, 1), (628, 1), (636, 1), (638, 1)],
[(20, 1),
 (22, 1),
 (26, 1),
 (33, 1),
 (79, 1),
 (128, 1),
 (371, 1),
 (635, 1),
 (636, 1),
 (637, 1),
 (638, 1),
 (639, 1),
 (640, 1)],
[(6, 1),
 (22, 1),
 (37, 1),
 (46, 1),
 (47, 1),
 (92, 2),
 (93, 2),
 (143, 1),
 (197, 1),
 (207, 1),
 (217, 1),
 (243, 1),
 (315, 1),
 (322, 1),
 (373, 1),
 (431, 1),
 (544, 1),
 (550, 1),
 (698, 1),
 (1124, 1)],
[(161, 1),
 (170, 1),
```

```
 (179, 1),
 (247, 1),
 (251, 1),
 (288, 1),
 (406, 1),
 (458, 1),
 (477, 1)],
[(26, 1),
 (179, 1),
 (192, 1),
 (247, 1),
 (1125, 1),
 (1126, 1),
 (1127, 1),
 (1128, 1)],
[(74, 1), (79, 1), (197, 1), (263, 1), (1129, 1), (1130, 1)],
[(74, 1),
 (79, 1),
 (117, 1),
 (164, 1),
 (710, 1),
 (720, 1),
 (1131, 1),
 (1132, 1),
 (1133, 1),
 (1134, 1)],
[(6, 1),
 (36, 1),
 (52, 1),
 (62, 1),
 (77, 2),
 (79, 1),
 (80, 1),
 (98, 2),
 (105, 1),
 (114, 3),
 (155, 1),
 (164, 1),
 (171, 1),
 (179, 2),
 (197, 1),
 (219, 1),
 (221, 1),
 (229, 1),
 (231, 1),
 (296, 1),
 (300, 1),
 (339, 1),
 (343, 2),
 (353, 1),
 (368, 1),
 (401, 1),
 (433, 1),
 (435, 1),
 (442, 1),
 (495, 1),
 (662, 1),
 (667, 1),
 (719, 1),
 (788, 1),
 (811, 3),
 (821, 1),
 (1135, 1),
 (1136, 1),
 (1137, 1),
 (1138, 1),
 (1139, 1),
 (1140, 1),
 (1141, 1)],
[(33, 1),
 (79, 1),
 (94, 1),
 (211, 1),
 (510, 2),
 (538, 1),
 (887, 1),
 (888, 1),
 (889, 1),
 (890, 1),
 (891, 1)],
[(20, 2),
 (52, 1),
 (53, 1),
 (78, 1),
 (80, 3),
 (83, 2),
 (91, 1),
 (94, 1),
 (111, 2),
 (121, 1),
 (163, 1),
 (164, 1),
 (193, 3),
 (215, 2),
 (216, 2),
 (339, 1),
 (343, 1),
 (406, 1),
 (458, 1),
 (539, 1),
 (667, 2),
 (678, 2),
 (734, 1),
 (757, 1),
 (1117, 1),
 (1142, 1),
 (1143, 1),
 (1144, 1),
 (1145, 1),
 (1146, 1),
 (1147, 1),
 (1148, 1)],
[(16, 1),
 (33, 1),
 (49, 1),
 (87, 1),
```

```
 (94, 1),
 (238, 1),
 (255, 2),
 (258, 1),
 (271, 1),
 (434, 1),
 (673, 1),
 (1023, 1),
 (1149, 1),
 (1150, 1)],
[(19, 1), (34, 1), (47, 1), (52, 1), (56, 1), (69, 1), (546, 1), (903, 1)],
[(49, 1),
 (87, 1),
 (108, 1),
 (165, 1),
 (181, 1),
 (195, 1),
 (232, 1),
 (364, 1),
 (553, 1),
 (648, 1),
 (820, 1),
 (1151, 1),
 (1152, 1)],
[(37, 1),
 (49, 1),
 (52, 1),
 (83, 1),
 (94, 2),
 (179, 1),
 (193, 1),
 (207, 1),
 (578, 1)],
[(6, 1),
 (16, 1),
 (20, 1),
 (22, 1),
 (26, 1),
 (47, 1),
 (65, 1),
 (69, 1),
 (79, 1),
 (88, 1),
 (105, 1),
 (106, 1),
 (120, 1),
 (127, 1),
 (197, 1),
 (203, 1),
 (211, 1),
 (243, 1),
 (262, 1),
 (281, 1),
 (341, 1),
 (406, 1),
 (419, 1),
 (479, 1),
 (740, 1),
 (805, 1),
 (1062, 1)],
[(7, 1), (62, 1), (77, 1), (93, 1), (133, 1), (146, 1), (168, 1)],
[(16, 1), (92, 1), (93, 1), (192, 1), (237, 1), (353, 1), (1126, 1)],
[(44, 1),
 (47, 1),
 (88, 1),
 (90, 1),
 (99, 1),
 (103, 1),
 (164, 1),
 (179, 1),
 (207, 1),
 (268, 1),
 (391, 1),
 (730, 1),
 (838, 1)],
[(24, 1),
 (37, 2),
 (78, 1),
 (94, 2),
 (133, 1),
 (139, 1),
 (153, 1),
 (179, 1),
 (192, 1),
 (207, 1),
 (294, 1),
 (678, 1)],
[(6, 1),
 (16, 1),
 (24, 1),
 (26, 2),
 (49, 1),
 (79, 1),
 (89, 1),
 (144, 1),
 (168, 2),
 (185, 1),
 (193, 1),
 (211, 2),
 (244, 1),
 (288, 1),
 (299, 1),
 (353, 1),
 (374, 3),
 (494, 1),
 (523, 1),
 (614, 1),
 (641, 1),
 (647, 1),
 (772, 1),
 (856, 1),
 (888, 1),
 (1009, 1),
 (1148, 1),
 (1153, 1),
```

(1154, 1),
(1155, 1),
(1156, 1),
(1157, 1),
(1158, 1)],
[(26, 1),
(49, 1),
(79, 1),
(85, 1),
(193, 1),
(268, 1),
(343, 1),
(344, 1),
(770, 1),
(776, 1)],
[(16, 1), (46, 1), (47, 1), (211, 1), (286, 1), (358, 1), (762, 1)],
[(24, 1), (37, 1), (126, 1), (230, 1), (373, 1), (1159, 1)],
[(19, 1),
(36, 1),
(37, 1),
(41, 1),
(47, 1),
(48, 1),
(65, 1),
(69, 1),
(80, 2),
(81, 1),
(89, 1),
(94, 2),
(105, 2),
(113, 1),
(146, 1),
(159, 2),
(179, 1),
(264, 1),
(325, 1),
(405, 1),
(468, 1),
(562, 1),
(579, 1),
(587, 1),
(595, 2),
(616, 2),
(665, 1),
(678, 1),
(788, 1),
(827, 1),
(903, 1),
(1160, 1),
(1161, 1),
(1162, 1),
(1163, 1),
(1164, 1),
(1165, 1),
(1166, 1),
(1167, 1),
(1168, 1),
(1169, 1),
(1170, 1)],
[(26, 1),
(35, 4),
(39, 1),
(52, 1),
(74, 1),
(90, 1),
(164, 2),
(179, 1),
(197, 1),
(201, 2),
(297, 1),
(325, 1),
(333, 1),
(411, 1),
(419, 1),
(814, 1),
(1163, 1),
(1166, 1),
(1171, 1),
(1172, 1),
(1173, 1),
(1174, 1),
(1175, 1)],
[(97, 1), (166, 1), (197, 1), (230, 1), (343, 1), (534, 1), (1176, 1)],
[(16, 1),
(20, 1),
(26, 1),
(47, 1),
(52, 1),
(69, 1),
(74, 1),
(164, 1),
(262, 1),
(300, 1),
(329, 1),
(607, 1),
(622, 1),
(1077, 1),
(1148, 1),
(1177, 2),
(1178, 1),
(1179, 1),
(1180, 1)],
[(6, 1),
(33, 1),
(34, 1),
(45, 1),
(49, 1),
(80, 1),
(85, 1),
(89, 1),
(133, 2),
(164, 1),
(165, 1),
(176, 1),
(179, 1),

```
 (201, 1),
 (268, 1),
 (296, 1),
 (302, 1),
 (496, 1),
 (662, 1),
 (785, 1),
 (979, 1),
 (1181, 1),
 (1182, 1),
 (1183, 1)],
[(5, 1),
 (6, 1),
 (65, 1),
 (79, 1),
 (93, 1),
 (164, 1),
 (197, 1),
 (201, 1),
 (207, 1),
 (252, 1),
 (287, 1),
 (734, 1)],
[(26, 1),
 (33, 1),
 (53, 1),
 (185, 1),
 (192, 1),
 (193, 2),
 (197, 2),
 (215, 1),
 (472, 1),
 (1184, 1),
 (1185, 1)],
[(6, 1),
 (7, 1),
 (49, 1),
 (138, 1),
 (164, 1),
 (168, 1),
 (171, 1),
 (194, 1),
 (197, 1),
 (201, 1),
 (211, 2),
 (342, 1),
 (343, 1),
 (511, 1),
 (667, 1),
 (798, 1),
 (1063, 1),
 (1186, 1),
 (1187, 1)],
[(20, 1),
 (70, 1),
 (101, 1),
 (179, 2),
 (230, 1),
 (304, 1),
 (406, 1),
 (629, 1),
 (800, 1),
 (1188, 1),
 (1189, 1)],
[(6, 1),
 (16, 1),
 (46, 1),
 (47, 2),
 (69, 1),
 (79, 1),
 (88, 2),
 (186, 1),
 (197, 1),
 (211, 1),
 (243, 1),
 (373, 2),
 (664, 1),
 (678, 1),
 (788, 1),
 (939, 1),
 (1190, 1),
 (1191, 1),
 (1192, 1)],
[(33, 1), (62, 1), (92, 1), (343, 1), (1193, 1)],
[(26, 1),
 (49, 1),
 (57, 1),
 (99, 1),
 (136, 1),
 (139, 1),
 (179, 3),
 (193, 1),
 (215, 1),
 (298, 1),
 (312, 1),
 (368, 1),
 (406, 1),
 (671, 1),
 (1194, 1)],
[(0, 1),
 (4, 1),
 (6, 1),
 (34, 1),
 (36, 2),
 (47, 2),
 (52, 1),
 (62, 1),
 (66, 1),
 (87, 2),
 (88, 1),
 (136, 1),
 (168, 1),
 (177, 1),
 (178, 1),
 (223, 1),
```

```
 (270, 1),
 (276, 1),
 (401, 1),
 (689, 1),
 (984, 1),
 (1195, 1),
 (1196, 1)],
[(16, 1),
 (193, 1),
 (249, 1),
 (262, 1),
 (288, 1),
 (314, 1),
 (343, 1),
 (587, 1),
 (806, 2),
 (1197, 1)],
[(6, 1),
 (24, 1),
 (26, 1),
 (37, 1),
 (47, 3),
 (69, 1),
 (92, 1),
 (94, 1),
 (161, 1),
 (216, 1),
 (250, 1),
 (268, 1),
 (489, 1),
 (698, 1),
 (1198, 1),
 (1199, 1),
 (1200, 1),
 (1201, 1)],
[(35, 2),
 (105, 1),
 (120, 1),
 (177, 1),
 (179, 1),
 (263, 1),
 (325, 2),
 (411, 1),
 (420, 1),
 (484, 1),
 (1202, 1),
 (1203, 1),
 (1204, 1)],
[(26, 1), (79, 1), (121, 1), (193, 1), (268, 1), (387, 1)],
[(26, 1), (47, 1), (93, 1), (179, 1), (192, 1), (367, 1)],
[(16, 1), (202, 1), (211, 1), (286, 1), (517, 1), (559, 1), (1205, 1)],
[(80, 1),
 (167, 1),
 (179, 1),
 (258, 1),
 (343, 1),
 (399, 1),
 (535, 1),
 (553, 1),
 (675, 1),
 (774, 1),
 (1206, 1)],
[(6, 1),
 (7, 1),
 (11, 1),
 (33, 1),
 (94, 1),
 (139, 1),
 (162, 1),
 (193, 1),
 (200, 1),
 (203, 1),
 (211, 1),
 (230, 1),
 (252, 1),
 (302, 2),
 (536, 1),
 (1048, 1)],
[(26, 1),
 (33, 1),
 (79, 1),
 (92, 1),
 (93, 1),
 (104, 1),
 (105, 1),
 (143, 1),
 (197, 1),
 (234, 1),
 (243, 1),
 (254, 1),
 (266, 1),
 (294, 1),
 (791, 1),
 (1207, 1),
 (1208, 1),
 (1209, 1),
 (1210, 1),
 (1211, 1)],
[(20, 1),
 (22, 3),
 (34, 1),
 (39, 1),
 (41, 1),
 (57, 2),
 (62, 1),
 (79, 2),
 (83, 1),
 (87, 1),
 (88, 1),
 (89, 3),
 (92, 1),
 (94, 2),
 (103, 2),
 (113, 1),
 (128, 3),
 (130, 2),
```

```
 (129, 2),
 (136, 1),
 (139, 4),
 (151, 1),
 (153, 2),
 (164, 1),
 (176, 1),
 (179, 2),
 (193, 5),
 (200, 1),
 (203, 3),
 (211, 1),
 (243, 1),
 (304, 1),
 (306, 1),
 (316, 1),
 (341, 2),
 (387, 1),
 (478, 1),
 (514, 1),
 (515, 2),
 (636, 3),
 (793, 1),
 (806, 2),
 (872, 1),
 (1020, 1),
 (1029, 3),
 (1212, 1),
 (1213, 1),
 (1214, 1),
 (1215, 1),
 (1216, 1),
 (1217, 1)],
[(33, 1),
 (49, 2),
 (65, 1),
 (75, 1),
 (80, 1),
 (103, 2),
 (193, 1),
 (217, 2),
 (221, 2),
 (264, 1),
 (294, 1),
 (406, 1),
 (462, 1),
 (638, 1),
 (739, 2),
 (777, 1),
 (906, 1),
 (1218, 1),
 (1219, 1),
 (1220, 1)],
[(22, 1), (64, 3), (357, 1), (420, 1), (689, 1), (788, 1), (1170, 2)],
[(6, 1), (33, 1), (49, 1), (133, 1), (179, 1), (302, 1)],
[(33, 1),
 (37, 1),
 (49, 1),
 (79, 1),
 (120, 1),
 (133, 1),
 (142, 1),
 (354, 2),
 (408, 1),
 (409, 1),
 (492, 1),
 (762, 1),
 (800, 1)],
[(20, 1),
 (22, 3),
 (34, 1),
 (39, 1),
 (41, 1),
 (57, 2),
 (62, 1),
 (79, 2),
 (83, 1),
 (87, 1),
 (88, 1),
 (89, 3),
 (92, 1),
 (94, 2),
 (103, 2),
 (113, 1),
 (128, 3),
 (129, 2),
 (136, 1),
 (139, 4),
 (151, 1),
 (153, 2),
 (164, 1),
 (176, 1),
 (179, 2),
 (193, 5),
 (200, 1),
 (203, 3),
 (211, 1),
 (243, 1),
 (304, 1),
 (306, 1),
 (316, 1),
 (341, 2),
 (387, 1),
 (478, 1),
 (514, 1),
 (515, 2),
 (636, 3),
 (793, 1),
 (806, 2),
 (872, 1),
 (1020, 1),
 (1029, 3),
 (1212, 1),
 (1213, 1),
 (1214, 1),
 (1215, 1),
```

```
 (1215, 1),
 (1216, 1),
 (1217, 1)],
[(48, 1), (92, 1), (190, 1), (192, 1), (603, 1)],
[(50, 1), (94, 1), (133, 1), (179, 1)],
[(6, 1),
 (8, 1),
 (36, 1),
 (79, 1),
 (80, 1),
 (98, 1),
 (104, 2),
 (105, 1),
 (106, 1),
 (107, 1),
 (113, 1),
 (114, 4),
 (121, 2),
 (122, 2),
 (150, 1),
 (168, 1),
 (179, 1),
 (184, 1),
 (185, 1),
 (197, 1),
 (262, 2),
 (276, 1),
 (302, 1),
 (318, 1),
 (343, 3),
 (349, 2),
 (385, 1),
 (397, 1),
 (480, 1),
 (499, 2),
 (553, 1),
 (643, 1),
 (686, 1),
 (731, 2),
 (765, 2),
 (774, 1),
 (821, 1),
 (866, 1),
 (1101, 1),
 (1148, 1),
 (1221, 1),
 (1222, 1),
 (1223, 1),
 (1224, 1),
 (1225, 1),
 (1226, 1),
 (1227, 1),
 (1228, 1),
 (1229, 2),
 (1230, 2),
 (1231, 1),
 (1232, 1)],
[(62, 1),
 (64, 1),
 (87, 3),
 (92, 1),
 (103, 2),
 (104, 1),
 (105, 1),
 (109, 1),
 (114, 1),
 (122, 1),
 (131, 1),
 (134, 2),
 (144, 1),
 (197, 1),
 (232, 1),
 (260, 1),
 (312, 1),
 (316, 1),
 (318, 1),
 (328, 1),
 (343, 1),
 (385, 1),
 (470, 1),
 (496, 1),
 (594, 1),
 (643, 1),
 (734, 1),
 (774, 1),
 (825, 1),
 (1233, 1),
 (1234, 1),
 (1235, 1)],
[(33, 1), (97, 1), (302, 1), (480, 1), (671, 1), (1136, 1)],
[(6, 2),
 (20, 1),
 (26, 1),
 (34, 1),
 (37, 1),
 (46, 1),
 (47, 7),
 (52, 1),
 (64, 1),
 (69, 2),
 (77, 2),
 (79, 1),
 (83, 1),
 (155, 1),
 (206, 1),
 (217, 3),
 (262, 3),
 (325, 1),
 (372, 1),
 (419, 1),
 (477, 1),
 (540, 1),
 (547, 1),
 (603, 1),
 (625, 1),
 (839, 1),
```

```
 (1023, 1),
 (1236, 1),
 (1237, 1),
 (1238, 1),
 (1239, 1),
 (1240, 1),
 (1241, 1)],
[(6, 2),
 (11, 1),
 (22, 1),
 (34, 1),
 (36, 1),
 (46, 3),
 (47, 7),
 (51, 1),
 (52, 1),
 (53, 1),
 (59, 1),
 (64, 1),
 (65, 2),
 (77, 1),
 (83, 1),
 (85, 1),
 (92, 4),
 (96, 2),
 (106, 1),
 (120, 1),
 (125, 1),
 (164, 3),
 (168, 1),
 (172, 1),
 (185, 1),
 (199, 1),
 (217, 2),
 (221, 1),
 (224, 1),
 (230, 1),
 (236, 1),
 (239, 1),
 (241, 1),
 (249, 1),
 (258, 2),
 (264, 1),
 (271, 1),
 (283, 1),
 (288, 1),
 (294, 1),
 (308, 1),
 (325, 3),
 (363, 3),
 (367, 6),
 (374, 1),
 (385, 1),
 (387, 1),
 (408, 1),
 (411, 2),
 (419, 1),
 (420, 1),
 (479, 1),
 (494, 1),
 (571, 1),
 (605, 2),
 (606, 1),
 (609, 1),
 (627, 1),
 (732, 1),
 (740, 1),
 (750, 1),
 (757, 1),
 (906, 5),
 (975, 1),
 (1062, 1),
 (1079, 1),
 (1108, 1),
 (1242, 3),
 (1243, 1),
 (1244, 1),
 (1245, 1),
 (1246, 1),
 (1247, 1),
 (1248, 1),
 (1249, 1),
 (1250, 1),
 (1251, 1),
 (1252, 1),
 (1253, 1),
 (1254, 1),
 (1255, 1),
 (1256, 1),
 (1257, 1)],
[(6, 1),
 (16, 1),
 (33, 1),
 (65, 1),
 (114, 1),
 (173, 1),
 (197, 1),
 (200, 1),
 (536, 1),
 (914, 1),
 (1141, 1)],
[(20, 1),
 (93, 1),
 (139, 1),
 (268, 1),
 (418, 1),
 (768, 1),
 (1258, 1),
 (1259, 1)],
[(6, 1),
 (16, 1),
 (18, 1),
 (26, 1),
 (33, 1),
 (37, 1),
```

```
 (59, 1),
 (65, 1),
 (90, 1),
 (96, 1),
 (114, 2),
 (119, 1),
 (129, 1),
 (157, 1),
 (165, 2),
 (225, 1),
 (299, 1),
 (310, 1),
 (434, 1),
 (487, 1),
 (596, 1),
 (614, 1),
 (627, 1),
 (703, 1),
 (706, 1),
 (1064, 1),
 (1158, 1),
 (1159, 1),
 (1260, 1),
 (1261, 1),
 (1262, 1),
 (1263, 1)],
[(94, 1),
 (168, 1),
 (179, 1),
 (192, 1),
 (235, 1),
 (244, 1),
 (303, 1),
 (339, 1),
 (343, 1),
 (663, 1)],
[(41, 1),
 (52, 1),
 (74, 1),
 (90, 1),
 (94, 1),
 (168, 1),
 (192, 1),
 (211, 1),
 (230, 1),
 (238, 1),
 (296, 2),
 (329, 1),
 (479, 1),
 (627, 1),
 (667, 1),
 (710, 1),
 (734, 1),
 (739, 1),
 (1264, 1)],
[(16, 1),
 (18, 1),
 (33, 1),
 (79, 1),
 (169, 1),
 (795, 1),
 (989, 1),
 (1122, 1),
 (1265, 1),
 (1266, 1)],
[(16, 1), (19, 1), (730, 1), (1261, 1)],
[(16, 1), (19, 1), (79, 1), (192, 1), (200, 1), (410, 1), (458, 1)],
[(56, 1),
 (68, 1),
 (94, 1),
 (115, 1),
 (120, 1),
 (179, 1),
 (211, 1),
 (296, 1),
 (691, 1),
 (1047, 1),
 (1124, 1),
 (1261, 1),
 (1267, 1)],
[(33, 1),
 (59, 2),
 (141, 1),
 (160, 1),
 (168, 2),
 (184, 1),
 (442, 2),
 (518, 1),
 (539, 1),
 (737, 1),
 (834, 1),
 (900, 1),
 (964, 1),
 (1039, 1),
 (1058, 1),
 (1108, 1),
 (1247, 1),
 (1268, 1),
 (1269, 1),
 (1270, 1)],
[(33, 1),
 (49, 1),
 (133, 1),
 (193, 1),
 (194, 1),
 (316, 1),
 (458, 1),
 (1271, 1),
 (1272, 1)],
[(6, 2),
 (15, 1),
 (62, 1),
 (71, 1),
 (83, 1),
 (88, 2),
```

```
 (90, 3),
 (177, 1),
 (205, 1),
 (329, 1),
 (399, 1),
 (429, 1),
 (474, 1),
 (536, 1),
 (558, 1),
 (583, 1),
 (942, 1),
 (982, 1),
 (1273, 1),
 (1274, 1)],
[(7, 1),
 (16, 1),
 (92, 1),
 (94, 1),
 (121, 1),
 (133, 1),
 (192, 1),
 (512, 1),
 (1275, 1),
 (1276, 1)],
[(19, 1), (35, 1), (103, 1), (120, 1), (571, 1), (641, 1), (1277, 1)],
[(179, 1),
 (181, 1),
 (302, 1),
 (341, 1),
 (643, 1),
 (774, 1),
 (791, 1),
 (1278, 1)],
[(22, 1),
 (64, 1),
 (128, 1),
 (139, 1),
 (153, 1),
 (179, 1),
 (268, 1),
 (343, 1)],
[(16, 1),
 (26, 1),
 (48, 1),
 (53, 1),
 (74, 1),
 (79, 1),
 (94, 1),
 (114, 1),
 (192, 1),
 (197, 1),
 (343, 1),
 (419, 2),
 (458, 2),
 (660, 1),
 (731, 1),
 (785, 1),
 (1279, 1)],
[(6, 1),
 (35, 1),
 (179, 1),
 (262, 2),
 (286, 1),
 (405, 1),
 (477, 1),
 (1124, 1),
 (1253, 1),
 (1280, 1),
 (1281, 1)],
[(7, 1), (95, 1), (143, 1), (177, 1), (179, 1), (970, 1), (1282, 1)],
[(16, 1),
 (33, 1),
 (65, 1),
 (79, 1),
 (89, 1),
 (103, 1),
 (122, 1),
 (252, 1),
 (264, 1),
 (406, 1),
 (774, 1),
 (1283, 1)],
[(26, 1),
 (35, 1),
 (48, 2),
 (185, 1),
 (226, 1),
 (268, 1),
 (300, 1),
 (477, 1),
 (702, 1),
 (805, 1),
 (1259, 1)],
[(93, 1),
 (121, 1),
 (179, 1),
 (207, 1),
 (272, 1),
 (315, 1),
 (800, 1),
 (872, 1)],
[(22, 1),
 (26, 1),
 (33, 1),
 (36, 1),
 (79, 3),
 (114, 1),
 (122, 1),
 (128, 1),
 (136, 1),
 (138, 1),
 (139, 1),
 (146, 2),
 (153, 1),
 (197, 2),
 (221, 1),
```

```
 (221, 1),
 (238, 1),
 (435, 1),
 (532, 1),
 (571, 1),
 (774, 1),
 (790, 1),
 (1019, 1),
 (1284, 1),
 (1285, 1),
 (1286, 1)],
[(79, 2), (197, 1), (331, 1), (636, 1), (651, 1)],
[(22, 1),
 (65, 1),
 (105, 1),
 (179, 1),
 (224, 1),
 (262, 1),
 (319, 1),
 (620, 1),
 (1287, 1)],
[(16, 1),
 (41, 1),
 (47, 1),
 (52, 1),
 (62, 1),
 (65, 1),
 (71, 1),
 (77, 1),
 (88, 2),
 (136, 1),
 (177, 1),
 (179, 1),
 (217, 1),
 (219, 1),
 (221, 1),
 (234, 1),
 (258, 1),
 (277, 1),
 (283, 1),
 (296, 1),
 (492, 1),
 (591, 1),
 (610, 1),
 (614, 1),
 (743, 1)],
[(13, 1),
 (26, 1),
 (79, 1),
 (101, 1),
 (114, 1),
 (179, 1),
 (197, 1),
 (198, 1),
 (262, 1),
 (331, 1),
 (914, 1),
 (1288, 1)],
[(35, 1),
 (90, 1),
 (103, 1),
 (104, 1),
 (105, 1),
 (127, 1),
 (186, 1),
 (232, 1),
 (259, 1),
 (286, 1),
 (322, 1),
 (406, 1),
 (411, 1),
 (591, 1),
 (625, 1),
 (702, 1),
 (762, 1),
 (778, 1),
 (1029, 1),
 (1289, 1),
 (1290, 1),
 (1291, 1),
 (1292, 1),
 (1293, 2)],
[(6, 1),
 (16, 1),
 (22, 1),
 (25, 1),
 (33, 3),
 (37, 1),
 (63, 2),
 (92, 1),
 (94, 1),
 (95, 1),
 (97, 1),
 (103, 1),
 (128, 2),
 (133, 3),
 (139, 2),
 (150, 1),
 (153, 3),
 (157, 2),
 (172, 1),
 (177, 1),
 (194, 1),
 (213, 1),
 (231, 1),
 (255, 1),
 (258, 1),
 (262, 1),
 (358, 1),
 (373, 2),
 (406, 1),
 (446, 1),
 (480, 1),
 (559, 1),
 (611, 1),
 (623, 1),
```

```
 (685, 1),
 (740, 1),
 (770, 1),
 (776, 1),
 (793, 1),
 (906, 1),
 (917, 1),
 (938, 1),
 (1001, 1),
 (1029, 1),
 (1034, 1),
 (1261, 1),
 (1294, 1),
 (1295, 1),
 (1296, 1),
 (1297, 1),
 (1298, 2),
 (1299, 1),
 (1300, 1),
 (1301, 1),
 (1302, 1),
 (1303, 1),
 (1304, 1),
 (1305, 1),
 (1306, 1),
 (1307, 1),
 (1308, 1),
 (1309, 1),
 (1310, 1),
 (1311, 1),
 (1312, 1),
 (1313, 1)],
[(19, 1),
 (33, 1),
 (96, 1),
 (193, 2),
 (316, 1),
 (343, 2),
 (458, 1),
 (740, 2),
 (752, 2),
 (1230, 1)],
[(6, 1),
 (16, 1),
 (33, 1),
 (49, 1),
 (51, 1),
 (79, 2),
 (133, 2),
 (192, 1),
 (195, 1),
 (249, 1),
 (334, 1),
 (420, 1),
 (512, 1),
 (812, 1),
 (1272, 1),
 (1314, 1),
 (1315, 1),
 (1316, 1)],
[(33, 1), (164, 1), (179, 1), (636, 1), (705, 1)],
[(4, 1),
 (34, 1),
 (35, 1),
 (79, 1),
 (86, 1),
 (161, 2),
 (179, 1),
 (202, 1),
 (478, 1),
 (663, 1),
 (1163, 1),
 (1317, 1),
 (1318, 1)],
[(13, 1),
 (16, 1),
 (87, 1),
 (168, 1),
 (181, 1),
 (193, 1),
 (306, 1),
 (310, 1),
 (345, 1),
 (374, 1),
 (553, 1),
 (638, 1),
 (1293, 1),
 (1319, 1),
 (1320, 1)],
[(15, 1),
 (33, 1),
 (79, 1),
 (87, 1),
 (89, 1),
 (117, 1),
 (126, 1),
 (169, 1),
 (200, 1),
 (203, 1),
 (387, 1),
 (440, 2),
 (920, 1),
 (1321, 1),
 (1322, 2)],
[(16, 1),
 (46, 1),
 (78, 1),
 (103, 1),
 (143, 1),
 (161, 1),
 (213, 1),
 (238, 1),
 (259, 1),
 (302, 1),
 (630, 1),
```

```
 (820, 1)],
[(36, 1),
 (132, 1),
 (134, 1),
 (146, 1),
 (165, 2),
 (211, 1),
 (309, 1),
 (435, 1),
 (913, 1),
 (914, 1),
 (1323, 1)],
[(179, 1), (192, 1), (458, 1), (496, 1), (514, 1), (876, 1), (1324, 1)],
[(39, 1),
 (47, 1),
 (48, 1),
 (71, 1),
 (79, 1),
 (86, 1),
 (87, 1),
 (88, 1),
 (154, 1),
 (168, 1),
 (343, 1),
 (410, 1),
 (419, 1),
 (540, 1),
 (678, 1),
 (1321, 1),
 (1325, 1)],
[(62, 1), (79, 1), (106, 1), (114, 1), (231, 1), (262, 1), (343, 1)],
[(16, 1),
 (83, 1),
 (139, 1),
 (157, 1),
 (207, 1),
 (318, 1),
 (345, 1),
 (385, 1),
 (603, 1),
 (780, 1),
 (1326, 1)],
[(16, 1),
 (26, 1),
 (47, 1),
 (143, 1),
 (243, 2),
 (419, 1),
 (479, 1),
 (561, 1),
 (562, 1),
 (628, 1)],
[(33, 1),
 (49, 1),
 (53, 1),
 (62, 1),
 (104, 1),
 (114, 1),
 (162, 1),
 (193, 2),
 (215, 1),
 (538, 1),
 (731, 1),
 (815, 1),
 (1029, 1),
 (1327, 1),
 (1328, 1)],
[(0, 2),
 (6, 1),
 (16, 1),
 (22, 1),
 (35, 2),
 (36, 1),
 (39, 1),
 (47, 2),
 (48, 1),
 (52, 1),
 (65, 1),
 (81, 1),
 (88, 2),
 (93, 1),
 (141, 1),
 (179, 3),
 (184, 1),
 (185, 1),
 (197, 1),
 (243, 1),
 (318, 1),
 (325, 2),
 (419, 1)],
[(13, 1),
 (179, 1),
 (193, 1),
 (271, 1),
 (331, 1),
 (510, 1),
 (876, 1),
 (1023, 1),
 (1044, 1),
 (1221, 1),
 (1329, 1),
 (1330, 1)],
[(6, 1),
 (16, 2),
 (65, 1),
 (88, 1),
 (112, 1),
 (114, 1),
 (134, 1),
 (179, 2),
 (193, 1),
 (194, 1),
 (200, 1),
 (202, 1),
 (217, 1),
```

```
 (230, 1),
 (262, 1),
 (264, 1),
 (305, 1),
 (343, 1),
 (388, 1),
 (1248, 1),
 (1331, 1)],
[(33, 1),
 (94, 1),
 (169, 1),
 (211, 1),
 (215, 1),
 (247, 1),
 (252, 1),
 (380, 1),
 (435, 1),
 (461, 1),
 (511, 1),
 (1190, 1),
 (1332, 1)],
[(13, 1),
 (37, 1),
 (92, 1),
 (94, 1),
 (105, 1),
 (129, 1),
 (147, 1),
 (164, 1),
 (192, 1),
 (1333, 1),
 (1334, 1)],
[(7, 1),
 (61, 1),
 (164, 1),
 (179, 1),
 (193, 1),
 (197, 1),
 (247, 1),
 (765, 1),
 (1023, 1)],
[(114, 1), (121, 1), (179, 1), (200, 1), (203, 1), (343, 1)],
[(11, 1),
 (33, 1),
 (94, 1),
 (125, 1),
 (139, 2),
 (153, 2),
 (164, 1),
 (179, 1),
 (203, 1),
 (243, 1),
 (354, 1),
 (510, 1),
 (554, 1),
 (916, 1),
 (1335, 1),
 (1336, 1),
 (1337, 1),
 (1338, 1),
 (1339, 1),
 (1340, 1),
 (1341, 1)],
[(24, 1),
 (33, 1),
 (36, 1),
 (87, 1),
 (92, 1),
 (110, 1),
 (133, 1),
 (578, 1),
 (831, 1),
 (1234, 1)],
[(33, 2),
 (70, 1),
 (79, 1),
 (250, 1),
 (268, 1),
 (288, 1),
 (300, 1),
 (374, 1),
 (387, 1),
 (512, 1),
 (636, 1),
 (762, 1),
 (1022, 1),
 (1342, 1)],
[(16, 1),
 (19, 1),
 (20, 2),
 (24, 1),
 (26, 1),
 (33, 2),
 (62, 1),
 (79, 4),
 (94, 2),
 (112, 1),
 (122, 1),
 (125, 1),
 (127, 1),
 (141, 1),
 (197, 1),
 (300, 1),
 (304, 1),
 (333, 1),
 (343, 1),
 (357, 1),
 (387, 1),
 (429, 1),
 (454, 1),
 (524, 1),
 (547, 1),
 (617, 1),
 (740, 1),
 (1186, 1),
```

```
 (1193, 1),
 (1296, 1),
 (1343, 2),
 (1344, 1)],
[(6, 1), (35, 1), (65, 1), (79, 1), (120, 1), (264, 1), (283, 1), (560, 1)],
[(33, 1),
 (61, 1),
 (79, 1),
 (89, 1),
 (90, 1),
 (136, 1),
 (197, 1),
 (388, 1),
 (458, 1)],
[(0, 1), (62, 1), (65, 1), (105, 1), (263, 1), (595, 1), (1047, 1)],
[(6, 2),
 (16, 1),
 (22, 3),
 (26, 1),
 (34, 1),
 (35, 5),
 (36, 1),
 (39, 1),
 (47, 2),
 (48, 1),
 (62, 1),
 (65, 4),
 (66, 2),
 (71, 1),
 (79, 3),
 (87, 4),
 (88, 6),
 (105, 1),
 (136, 2),
 (141, 1),
 (146, 1),
 (159, 1),
 (161, 1),
 (164, 3),
 (168, 1),
 (184, 1),
 (185, 1),
 (190, 1),
 (217, 1),
 (219, 1),
 (221, 1),
 (223, 1),
 (224, 1),
 (231, 1),
 (234, 1),
 (238, 2),
 (243, 1),
 (247, 1),
 (248, 1),
 (262, 2),
 (270, 2),
 (276, 1),
 (300, 1),
 (325, 2),
 (344, 1),
 (373, 1),
 (387, 1),
 (390, 1),
 (399, 1),
 (401, 1),
 (410, 1),
 (463, 1),
 (535, 1),
 (546, 1),
 (595, 1),
 (613, 1),
 (616, 1),
 (640, 1),
 (723, 1),
 (759, 1),
 (778, 1),
 (788, 1),
 (881, 1),
 (962, 1),
 (976, 1),
 (978, 2),
 (984, 1),
 (1002, 1),
 (1029, 1),
 (1047, 1),
 (1091, 1),
 (1092, 1),
 (1101, 1),
 (1103, 1),
 (1109, 1),
 (1119, 1),
 (1121, 1),
 (1170, 1),
 (1192, 1),
 (1345, 1),
 (1346, 1),
 (1347, 1),
 (1348, 1),
 (1349, 1),
 (1350, 1),
 (1351, 1),
 (1352, 1),
 (1353, 1),
 (1354, 1),
 (1355, 2),
 (1356, 1),
 (1357, 1)],
[(94, 1), (209, 1), (419, 1), (434, 1), (439, 1), (888, 1)],
[(16, 1),
 (33, 1),
 (110, 1),
 (133, 1),
 (230, 1),
 (345, 1),
 (422, 1),
 (517, 1)
```

```
 (517, 1),
 (993, 1),
 (1048, 1),
 (1104, 1)],
[(16, 2),
 (19, 1),
 (33, 1),
 (105, 1),
 (107, 1),
 (173, 1),
 (300, 2),
 (534, 1),
 (1203, 1),
 (1358, 1)],
[(52, 1),
 (62, 1),
 (79, 1),
 (80, 1),
 (86, 1),
 (93, 1),
 (94, 1),
 (165, 2),
 (171, 1),
 (197, 1),
 (249, 1),
 (309, 2),
 (345, 1),
 (388, 1),
 (514, 1),
 (660, 1),
 (664, 1),
 (670, 1),
 (742, 1),
 (945, 1),
 (946, 1),
 (947, 1),
 (948, 1)],
[(6, 1),
 (11, 1),
 (16, 2),
 (22, 1),
 (29, 1),
 (35, 1),
 (36, 1),
 (47, 1),
 (65, 1),
 (88, 1),
 (92, 1),
 (93, 1),
 (130, 1),
 (164, 1),
 (168, 1),
 (203, 1),
 (211, 2),
 (217, 1),
 (234, 1),
 (243, 3),
 (244, 1),
 (283, 1),
 (292, 1),
 (347, 1),
 (352, 1),
 (359, 1),
 (375, 1),
 (427, 2),
 (474, 1),
 (730, 1),
 (877, 2),
 (962, 1),
 (985, 1),
 (1359, 1),
 (1360, 1),
 (1361, 1)],
[(16, 1),
 (35, 1),
 (61, 1),
 (79, 1),
 (89, 2),
 (136, 1),
 (186, 1),
 (212, 1),
 (217, 1),
 (234, 1)],
[(93, 1), (179, 2), (306, 1)],
[(51, 1),
 (80, 1),
 (87, 1),
 (103, 1),
 (120, 1),
 (164, 1),
 (179, 1),
 (240, 1),
 (241, 1),
 (496, 1),
 (587, 1),
 (1106, 1),
 (1107, 1),
 (1108, 2)],
[(22, 1), (49, 1), (176, 1), (179, 1), (193, 1), (512, 1), (812, 1)],
[(114, 1), (193, 1), (262, 1), (603, 1), (1362, 1)],
[(19, 1), (33, 1), (603, 1), (636, 1), (986, 1), (1363, 1)],
[(22, 1),
 (24, 1),
 (37, 1),
 (65, 1),
 (69, 1),
 (104, 1),
 (179, 1),
 (193, 1),
 (200, 1),
 (343, 1),
 (347, 1),
 (388, 1),
 (1364, 1),
 (1365, 1)],
[(47, 1),
```

```
[(1, 1),
 (62, 1),
 (105, 1),
 (179, 1),
 (202, 1),
 (306, 1),
 (325, 1),
 (347, 1),
 (411, 1),
 (514, 1),
 (595, 1),
 (888, 1)],
[(7, 1),
 (12, 1),
 (20, 4),
 (49, 1),
 (53, 1),
 (70, 1),
 (74, 1),
 (78, 1),
 (80, 1),
 (87, 1),
 (93, 1),
 (127, 2),
 (136, 1),
 (139, 1),
 (146, 1),
 (164, 3),
 (181, 1),
 (193, 2),
 (211, 2),
 (225, 2),
 (306, 1),
 (307, 1),
 (314, 1),
 (318, 1),
 (368, 1),
 (371, 2),
 (433, 1),
 (479, 1),
 (521, 1),
 (529, 1),
 (572, 1),
 (595, 2),
 (675, 1),
 (711, 2),
 (744, 1),
 (759, 1),
 (857, 1),
 (920, 1),
 (1103, 1),
 (1148, 1),
 (1203, 1),
 (1248, 1),
 (1262, 1),
 (1366, 1),
 (1367, 1),
 (1368, 1),
 (1369, 1),
 (1370, 1),
 (1371, 1),
 (1372, 1),
 (1373, 1),
 (1374, 1),
 (1375, 1),
 (1376, 1),
 (1377, 1),
 (1378, 1),
 (1379, 1),
 (1380, 1)],
[(93, 1), (263, 1), (702, 1), (982, 1), (1155, 1), (1381, 1)],
[(103, 1), (302, 1), (1382, 1), (1383, 1)],
[(34, 1), (62, 1), (444, 1), (1248, 1), (1384, 1), (1385, 1)],
[(35, 1),
 (62, 1),
 (93, 1),
 (105, 1),
 (143, 1),
 (161, 1),
 (179, 1),
 (217, 1),
 (434, 1),
 (595, 1),
 (827, 1)],
[(16, 1),
 (35, 1),
 (56, 1),
 (62, 1),
 (68, 1),
 (92, 1),
 (93, 1),
 (161, 1),
 (185, 1),
 (287, 1),
 (662, 1),
 (691, 1),
 (1271, 1),
 (1272, 1),
 (1386, 1),
 (1387, 1)],
[(41, 1),
 (51, 1),
 (78, 1),
 (94, 1),
 (120, 1),
 (133, 1),
 (179, 1),
 (186, 1),
 (716, 1),
 (1272, 1),
 (1388, 1)],
[(53, 1),
 (92, 1),
 (162, 1),
 (318, 1),
 (385, 1),
```

```
 (1101, 1),
 (1389, 1),
 (1390, 1)],
[(36, 1),
 (43, 1),
 (47, 1),
 (48, 1),
 (65, 1),
 (71, 1),
 (79, 2),
 (88, 1),
 (133, 1),
 (136, 1),
 (179, 2),
 (248, 1),
 (260, 1),
 (264, 1),
 (292, 1),
 (329, 1),
 (435, 1),
 (474, 1),
 (478, 1),
 (558, 1),
 (667, 1),
 (762, 1),
 (1049, 1)],
[(53, 1), (94, 1), (120, 1), (185, 1), (197, 1), (306, 1)],
[(6, 1),
 (33, 1),
 (86, 1),
 (87, 1),
 (89, 1),
 (133, 2),
 (179, 1),
 (267, 1),
 (308, 1),
 (348, 1),
 (423, 1),
 (553, 1),
 (977, 1),
 (1391, 1),
 (1392, 1),
 (1393, 1)],
[(16, 1),
 (46, 2),
 (62, 1),
 (64, 1),
 (80, 1),
 (88, 3),
 (94, 1),
 (127, 1),
 (159, 2),
 (190, 2),
 (230, 1),
 (237, 1),
 (238, 1),
 (268, 1),
 (373, 1),
 (553, 1),
 (665, 1),
 (752, 1),
 (827, 1),
 (949, 1),
 (1394, 1),
 (1395, 1)],
[(53, 1), (192, 1), (197, 1), (522, 1), (528, 1), (636, 1)],
[(121, 1), (159, 1), (164, 1), (201, 1), (259, 1)],
[(16, 1),
 (47, 1),
 (88, 1),
 (179, 1),
 (211, 1),
 (238, 1),
 (373, 1),
 (665, 1),
 (778, 1),
 (976, 1),
 (978, 2),
 (1191, 1)],
[(52, 1),
 (80, 1),
 (90, 1),
 (114, 1),
 (258, 2),
 (271, 1),
 (314, 1),
 (343, 1),
 (399, 1),
 (434, 1),
 (620, 1),
 (1396, 1)],
[(37, 1),
 (51, 1),
 (80, 1),
 (92, 1),
 (93, 1),
 (133, 2),
 (146, 1),
 (164, 1),
 (188, 1),
 (192, 1),
 (207, 1),
 (268, 1),
 (300, 1),
 (358, 1),
 (391, 1),
 (429, 1),
 (598, 1),
 (734, 1),
 (1397, 1)],
[(6, 1),
 (24, 1),
 (26, 1),
 (33, 1),
 (143, 1),
```

```
 (179, 1),
 (192, 1),
 (296, 1),
 (388, 1),
 (409, 1),
 (427, 1),
 (485, 1),
 (603, 1),
 (1398, 1)],
[(22, 1),
 (77, 1),
 (92, 1),
 (94, 1),
 (103, 1),
 (128, 1),
 (524, 1),
 (790, 1),
 (1187, 1),
 (1251, 1),
 (1399, 1)],
[(45, 1), (78, 1), (92, 1), (164, 1), (900, 1), (1126, 1)],
[(26, 1), (53, 1), (185, 1), (226, 1), (664, 1)],
[(16, 1),
 (120, 1),
 (133, 1),
 (142, 1),
 (168, 1),
 (179, 1),
 (197, 1),
 (402, 1)],
[(43, 1), (164, 1), (296, 1), (343, 1), (387, 1), (478, 1), (1400, 1)],
[(0, 1),
 (11, 1),
 (33, 1),
 (99, 1),
 (114, 1),
 (139, 1),
 (197, 2),
 (201, 1),
 (228, 1),
 (242, 1),
 (329, 1),
 (510, 1),
 (924, 1),
 (1044, 1),
 (1104, 1),
 (1401, 1)],
[(16, 1), (24, 1), (93, 1), (112, 1), (354, 1), (554, 1), (941, 1)],
[(6, 1),
 (49, 1),
 (79, 2),
 (92, 1),
 (94, 1),
 (109, 1),
 (136, 1),
 (153, 1),
 (193, 1),
 (215, 1),
 (221, 1),
 (268, 1),
 (345, 1),
 (369, 1),
 (956, 1),
 (1397, 1)],
[(33, 1),
 (43, 1),
 (79, 1),
 (113, 1),
 (168, 1),
 (211, 1),
 (609, 1),
 (739, 1),
 (982, 1),
 (1262, 1)],
[(4, 1),
 (6, 1),
 (37, 1),
 (46, 1),
 (47, 2),
 (69, 1),
 (87, 1),
 (88, 1),
 (185, 1),
 (217, 1),
 (363, 1),
 (367, 1),
 (368, 1),
 (385, 1),
 (458, 1),
 (678, 1),
 (806, 1),
 (874, 1),
 (1328, 1),
 (1368, 1),
 (1402, 1)],
[(92, 2),
 (140, 1),
 (143, 1),
 (161, 1),
 (164, 1),
 (177, 1),
 (217, 1),
 (222, 1),
 (358, 1),
 (411, 1),
 (429, 1),
 (1403, 1)],
[(16, 1),
 (61, 1),
 (179, 1),
 (334, 1),
 (636, 1),
 (722, 1),
 (1404, 1),
 (1405, 1)],
```

```
[(16, 1),
 (19, 1),
 (24, 1),
 (79, 1),
 (164, 1),
 (169, 1),
 (306, 1),
 (352, 1),
 (409, 1)],
[(16, 1), (26, 1), (33, 1), (93, 1), (576, 1), (638, 1), (1406, 1)],
[(4, 1), (13, 1), (16, 1), (21, 1), (28, 1), (33, 1), (1326, 1), (1407, 1)],
[(24, 1),
 (33, 1),
 (52, 1),
 (77, 1),
 (126, 1),
 (133, 1),
 (171, 1),
 (344, 1),
 (433, 1),
 (651, 1),
 (1183, 1),
 (1408, 1)],
[(5, 1), (11, 1), (179, 1), (477, 1), (1248, 1), (1409, 1)],
[(6, 1),
 (20, 1),
 (26, 1),
 (33, 1),
 (41, 1),
 (49, 1),
 (53, 1),
 (78, 1),
 (94, 1),
 (192, 2),
 (194, 1),
 (235, 1),
 (961, 1),
 (1128, 1),
 (1410, 1),
 (1411, 1)],
[(89, 1), (92, 1), (258, 1), (343, 1), (1033, 1), (1412, 1)],
[(11, 1),
 (20, 3),
 (80, 1),
 (94, 1),
 (136, 1),
 (169, 1),
 (179, 2),
 (193, 1),
 (245, 1),
 (246, 1),
 (295, 1),
 (1336, 1),
 (1413, 1)],
[(62, 1), (89, 1), (92, 1), (114, 1), (164, 1), (165, 1), (295, 1), (343, 1)],
[(13, 1),
 (62, 1),
 (83, 1),
 (95, 1),
 (109, 1),
 (114, 1),
 (185, 1),
 (197, 1),
 (293, 1),
 (343, 1),
 (496, 1),
 (508, 1),
 (662, 1),
 (686, 1),
 (836, 1),
 (919, 1),
 (1414, 1),
 (1415, 1),
 (1416, 1),
 (1417, 1)],
[(20, 1), (62, 1), (64, 1), (94, 1), (121, 1), (472, 1)],
[(20, 1),
 (39, 1),
 (64, 2),
 (65, 1),
 (108, 1),
 (120, 1),
 (164, 1),
 (263, 2),
 (350, 1),
 (396, 1),
 (397, 1),
 (743, 1),
 (866, 1),
 (906, 1),
 (1118, 1),
 (1418, 1)],
[(16, 1),
 (20, 1),
 (80, 1),
 (83, 1),
 (90, 1),
 (92, 1),
 (161, 2),
 (193, 2),
 (387, 1),
 (521, 1),
 (643, 1),
 (1394, 2)],
[(26, 1),
 (77, 1),
 (105, 1),
 (122, 1),
 (163, 1),
 (168, 1),
 (211, 1),
 (296, 1),
 (368, 1),
 (489, 1)],
[(0, 1), (114, 2), (164, 1), (179, 1), (254, 1), (1419, 1)],
[(25, 1), (27, 1), (47, 1), (48, 1), (88, 1), (126, 1), (170, 1)]
```

```
[(33, 1), (37, 1), (47, 1), (48, 1), (66, 1), (136, 1), (179, 1)],
[(7, 1), (16, 1), (33, 1), (95, 1), (109, 1), (133, 2), (177, 1)],
[(7, 2),
 (19, 1),
 (33, 1),
 (77, 1),
 (179, 1),
 (306, 1),
 (458, 1),
 (1177, 1),
 (1193, 1)],
[(33, 1),
 (62, 1),
 (79, 1),
 (83, 1),
 (141, 1),
 (176, 1),
 (180, 2),
 (197, 2),
 (231, 1),
 (300, 1),
 (666, 1),
 (1349, 1),
 (1420, 1)],
[(33, 1), (80, 1), (111, 1), (458, 1), (1193, 1)],
[(20, 1), (33, 1), (87, 2), (164, 2), (169, 1), (181, 2), (1212, 1)],
[(16, 1),
 (48, 1),
 (75, 1),
 (262, 1),
 (401, 1),
 (472, 1),
 (529, 1),
 (1160, 1),
 (1421, 1)],
[(16, 1),
 (49, 1),
 (94, 1),
 (103, 1),
 (127, 1),
 (162, 1),
 (167, 1),
 (179, 1),
 (193, 2),
 (194, 1),
 (628, 1),
 (1422, 1)],
[(24, 1), (33, 2), (79, 1), (112, 1), (196, 1), (442, 1), (624, 1)],
[(53, 1), (61, 1), (89, 1), (197, 1), (201, 1), (492, 1)],
[(16, 1),
 (26, 1),
 (47, 1),
 (93, 1),
 (94, 1),
 (105, 1),
 (268, 1),
 (479, 1),
 (691, 1)],
[(20, 1),
 (164, 1),
 (177, 1),
 (193, 1),
 (478, 1),
 (636, 1),
 (752, 1),
 (1423, 1)],
[(37, 1),
 (46, 2),
 (47, 2),
 (88, 1),
 (96, 1),
 (231, 1),
 (299, 1),
 (464, 1),
 (662, 1),
 (897, 1),
 (1251, 1),
 (1252, 1),
 (1424, 1),
 (1425, 1)],
[(16, 1),
 (20, 1),
 (51, 1),
 (65, 1),
 (72, 1),
 (79, 1),
 (97, 1),
 (103, 1),
 (121, 1),
 (192, 1),
 (197, 1),
 (201, 1),
 (217, 1),
 (243, 1),
 (412, 1),
 (420, 1),
 (561, 1),
 (1426, 1),
 (1427, 1)],
[(22, 2),
 (92, 1),
 (128, 1),
 (133, 1),
 (164, 1),
 (258, 1),
 (920, 1),
 (1217, 1),
 (1428, 1)],
[(4, 1),
 (47, 1),
 (59, 1),
 (86, 1),
 (88, 1),
 (109, 1),
 (179, 1),
 (197, 1),
```

```
 (201, 1),
 (243, 1),
 (608, 1),
 (1429, 1)],
[(16, 1), (33, 1), (89, 1), (97, 1), (99, 1), (103, 1), (211, 1)],
[(41, 1), (164, 1), (312, 1), (599, 1), (628, 1)],
[(179, 1), (193, 1), (197, 1), (387, 1), (1104, 1)],
[(47, 1), (78, 1), (88, 1), (94, 1), (179, 1), (1430, 1)],
[(16, 1),
 (19, 1),
 (21, 1),
 (24, 1),
 (114, 1),
 (133, 1),
 (211, 1),
 (254, 1),
 (343, 1),
 (452, 1),
 (765, 1),
 (821, 1),
 (1230, 1)],
[(16, 1),
 (24, 1),
 (87, 2),
 (164, 1),
 (190, 1),
 (202, 1),
 (217, 2),
 (231, 1),
 (613, 1),
 (787, 1),
 (1029, 1),
 (1231, 1),
 (1431, 2),
 (1432, 1),
 (1433, 1)],
[(0, 1),
 (39, 1),
 (48, 1),
 (64, 1),
 (164, 1),
 (243, 1),
 (251, 1),
 (253, 2),
 (329, 1),
 (461, 1),
 (535, 2),
 (561, 1),
 (562, 1),
 (639, 1),
 (750, 1),
 (790, 1),
 (825, 1),
 (831, 1),
 (832, 1),
 (845, 1),
 (856, 1),
 (1434, 1),
 (1435, 1)],
[(16, 1), (19, 1), (26, 1), (33, 1), (122, 1), (271, 1), (774, 1)],
[(7, 1), (133, 1), (168, 1), (211, 1), (1048, 1), (1276, 1)],
[(26, 1), (120, 1), (121, 1), (142, 1), (179, 2), (354, 1), (554, 1)],
[(16, 2), (24, 1), (37, 1), (92, 1), (143, 1), (217, 1)],
[(20, 1), (164, 1), (250, 1), (1049, 1)],
[(33, 2),
 (79, 1),
 (80, 1),
 (87, 1),
 (103, 1),
 (152, 1),
 (168, 1),
 (211, 1),
 (215, 1),
 (231, 1),
 (255, 1),
 (300, 1),
 (343, 1),
 (357, 1),
 (510, 1),
 (659, 1),
 (684, 1),
 (845, 1),
 (1018, 1),
 (1115, 1),
 (1177, 1),
 (1343, 1),
 (1436, 1),
 (1437, 1),
 (1438, 1)],
[(16, 1), (24, 1), (33, 1), (387, 1), (525, 1), (1439, 1)],
[(5, 1),
 (16, 1),
 (19, 1),
 (20, 1),
 (26, 1),
 (36, 2),
 (94, 1),
 (130, 1),
 (143, 1),
 (164, 1),
 (172, 1),
 (217, 1),
 (222, 1),
 (263, 1),
 (302, 1),
 (322, 1),
 (411, 1),
 (477, 1),
 (765, 1),
 (800, 1),
 (1111, 1),
 (1440, 1),
 (1441, 1)],
[(62, 1),
 (79, 1),
```

```
 (87, 1),
 (92, 1),
 (168, 1),
 (197, 2),
 (211, 1),
 (343, 1),
 (387, 1),
 (434, 1)],
[(11, 1),
 (16, 1),
 (21, 1),
 (92, 1),
 (161, 1),
 (179, 1),
 (185, 2),
 (207, 1),
 (209, 1),
 (217, 1),
 (225, 1),
 (233, 1),
 (244, 1),
 (296, 1),
 (358, 1),
 (421, 1),
 (514, 2),
 (587, 1),
 (611, 2),
 (788, 1),
 (1212, 1),
 (1442, 1),
 (1443, 1),
 (1444, 2)],
[(22, 1),
 (24, 1),
 (29, 1),
 (33, 1),
 (49, 1),
 (62, 2),
 (89, 1),
 (92, 1),
 (94, 2),
 (104, 1),
 (112, 1),
 (114, 1),
 (128, 1),
 (153, 1),
 (179, 2),
 (192, 1),
 (193, 1),
 (195, 1),
 (207, 1),
 (211, 1),
 (309, 1),
 (312, 1),
 (363, 1),
 (368, 1),
 (510, 1),
 (524, 1),
 (534, 2),
 (562, 1),
 (845, 1),
 (859, 1),
 (993, 1),
 (1159, 1)],
[(33, 1),
 (37, 1),
 (78, 1),
 (94, 2),
 (133, 2),
 (139, 1),
 (153, 2),
 (168, 1),
 (177, 1),
 (190, 1),
 (213, 1),
 (387, 1),
 (433, 1),
 (956, 1)],
[(6, 1),
 (13, 1),
 (16, 1),
 (52, 1),
 (65, 1),
 (71, 1),
 (81, 1),
 (88, 1),
 (93, 1),
 (185, 1),
 (201, 1),
 (292, 1),
 (349, 1),
 (472, 1),
 (549, 1),
 (620, 1),
 (916, 1)],
[(24, 1),
 (33, 1),
 (41, 1),
 (64, 1),
 (79, 1),
 (93, 1),
 (121, 1),
 (139, 1),
 (164, 1),
 (165, 1),
 (179, 1),
 (230, 1),
 (314, 1),
 (369, 1),
 (492, 2),
 (515, 1),
 (627, 1),
 (772, 1),
 (831, 1),
 (945, 1),
```

```
 (1101, 1),
 (1445, 1),
 (1446, 1),
 (1447, 1),
 (1448, 1),
 (1449, 1),
 (1450, 1),
 (1451, 1)],
[(33, 1),
 (92, 1),
 (121, 1),
 (164, 1),
 (172, 1),
 (179, 1),
 (385, 1),
 (975, 1),
 (1039, 1)],
[(22, 1),
 (24, 1),
 (26, 2),
 (29, 1),
 (33, 4),
 (45, 1),
 (53, 1),
 (59, 1),
 (77, 1),
 (78, 1),
 (79, 2),
 (83, 3),
 (87, 1),
 (88, 1),
 (92, 1),
 (94, 1),
 (103, 1),
 (109, 1),
 (119, 1),
 (120, 1),
 (128, 1),
 (133, 1),
 (138, 1),
 (142, 1),
 (143, 1),
 (146, 1),
 (150, 1),
 (161, 2),
 (165, 1),
 (176, 3),
 (180, 1),
 (201, 1),
 (213, 1),
 (249, 1),
 (258, 1),
 (288, 1),
 (293, 1),
 (324, 1),
 (353, 1),
 (374, 2),
 (385, 1),
 (435, 1),
 (490, 1),
 (492, 1),
 (496, 2),
 (556, 1),
 (571, 1),
 (603, 1),
 (620, 1),
 (647, 2),
 (679, 1),
 (763, 1),
 (772, 1),
 (875, 1),
 (879, 1),
 (904, 1),
 (937, 2),
 (1023, 1),
 (1039, 1),
 (1063, 1),
 (1151, 1),
 (1155, 1),
 (1385, 1),
 (1386, 1),
 (1452, 1),
 (1453, 1),
 (1454, 1),
 (1455, 1),
 (1456, 1),
 (1457, 1),
 (1458, 1),
 (1459, 1),
 (1460, 1),
 (1461, 1),
 (1462, 1),
 (1463, 1),
 (1464, 1)],
[(24, 1),
 (26, 1),
 (33, 1),
 (49, 1),
 (168, 1),
 (244, 1),
 (245, 1),
 (246, 1),
 (406, 1),
 (435, 1),
 (492, 1)],
[(24, 1),
 (33, 3),
 (39, 1),
 (139, 1),
 (153, 1),
 (205, 1),
 (244, 1),
 (316, 1),
 (1465, 1)],
[(20, 1),
```

```
 (24, 1),
 (49, 1),
 (79, 1),
 (87, 1),
 (94, 1),
 (153, 1),
 (154, 1),
 (160, 1),
 (167, 1),
 (169, 1),
 (175, 1),
 (179, 1),
 (194, 1),
 (197, 1),
 (258, 1),
 (388, 1),
 (643, 1),
 (914, 1),
 (1053, 1),
 (1422, 1),
 (1466, 1)],
[(41, 1), (94, 1), (139, 1), (153, 1), (160, 1), (179, 1)],
[(16, 1),
 (33, 1),
 (35, 1),
 (51, 1),
 (64, 1),
 (69, 1),
 (79, 1),
 (90, 1),
 (197, 1),
 (250, 1),
 (302, 1),
 (343, 1),
 (474, 1),
 (569, 1),
 (1036, 1),
 (1068, 1),
 (1467, 1),
 (1468, 1),
 (1469, 1)],
[(13, 1),
 (16, 1),
 (29, 1),
 (45, 1),
 (87, 1),
 (92, 1),
 (105, 1),
 (114, 1),
 (138, 1),
 (146, 1),
 (164, 1),
 (177, 1),
 (193, 2),
 (287, 1),
 (349, 1),
 (496, 1),
 (669, 1),
 (897, 1),
 (982, 1),
 (1023, 1),
 (1470, 1)],
[(33, 1), (127, 1), (342, 1), (386, 1), (410, 1), (845, 1)],
[(47, 1),
 (52, 1),
 (61, 1),
 (89, 1),
 (90, 1),
 (179, 1),
 (306, 1),
 (316, 1),
 (629, 1),
 (1085, 2)],
[(22, 1),
 (43, 1),
 (94, 1),
 (95, 1),
 (539, 1),
 (628, 1),
 (636, 1),
 (671, 1),
 (1422, 1)],
[(33, 1), (139, 1), (153, 1), (179, 1), (458, 1)],
[(79, 1), (92, 1), (207, 1), (268, 1), (315, 1), (409, 1)],
[(5, 1), (6, 1), (49, 1), (83, 1), (94, 1), (357, 1), (519, 1), (1177, 1)],
[(16, 1),
 (35, 1),
 (314, 1),
 (363, 1),
 (643, 1),
 (877, 1),
 (1471, 1),
 (1472, 1)],
[(16, 1),
 (19, 1),
 (24, 1),
 (55, 1),
 (168, 1),
 (211, 1),
 (541, 1),
 (955, 1),
 (1126, 1)],
[(0, 1),
 (20, 2),
 (26, 2),
 (35, 1),
 (37, 1),
 (41, 1),
 (47, 1),
 (48, 2),
 (65, 1),
 (78, 1),
 (88, 1),
 (102, 1),
 (155, 1),
 (170, 2)
```

```
 (179, 2),
 (185, 1),
 (243, 1),
 (286, 1),
 (325, 1),
 (411, 1),
 (628, 1),
 (791, 1),
 (1109, 1)],
[(11, 1),
 (20, 3),
 (24, 1),
 (93, 1),
 (165, 1),
 (211, 1),
 (263, 1),
 (442, 1),
 (458, 1),
 (603, 1),
 (836, 1)],
[(22, 1), (114, 1), (128, 1), (237, 1), (343, 1), (388, 1)],
[(121, 1), (139, 1), (179, 1), (192, 1), (458, 1), (515, 1), (634, 1)],
[(41, 1), (94, 1), (179, 1), (636, 1)],
[(13, 1),
 (25, 1),
 (49, 1),
 (87, 1),
 (139, 1),
 (141, 1),
 (193, 1),
 (258, 1),
 (388, 2)],
[(16, 1), (92, 1), (93, 1), (143, 1), (234, 1), (302, 1), (591, 1)],
[(7, 2),
 (22, 1),
 (50, 1),
 (51, 1),
 (52, 1),
 (61, 1),
 (62, 2),
 (76, 1),
 (78, 1),
 (83, 1),
 (89, 1),
 (109, 1),
 (120, 2),
 (133, 2),
 (139, 2),
 (142, 1),
 (153, 2),
 (159, 1),
 (164, 1),
 (168, 1),
 (172, 1),
 (179, 1),
 (216, 1),
 (262, 1),
 (276, 1),
 (348, 1),
 (367, 2),
 (406, 2),
 (546, 1),
 (593, 1),
 (651, 1),
 (667, 1),
 (759, 1),
 (901, 1),
 (962, 1),
 (970, 3),
 (1039, 1),
 (1079, 1),
 (1224, 1),
 (1365, 1),
 (1473, 1),
 (1474, 1),
 (1475, 1),
 (1476, 1),
 (1477, 1),
 (1478, 1),
 (1479, 1),
 (1480, 1)],
[(79, 1),
 (89, 1),
 (112, 1),
 (133, 1),
 (150, 1),
 (169, 1),
 (179, 1),
 (197, 1),
 (255, 1),
 (307, 1),
 (671, 2),
 (780, 1),
 (1050, 1)],
[(71, 1),
 (81, 1),
 (87, 1),
 (217, 1),
 (231, 1),
 (238, 1),
 (302, 1),
 (410, 1),
 (569, 1),
 (689, 1),
 (984, 1),
 (1058, 1)],
[(47, 1), (65, 1), (79, 1), (105, 1), (120, 1), (268, 1), (387, 1), (477, 1)],
[(16, 1),
 (45, 1),
 (47, 1),
 (65, 1),
 (79, 1),
 (81, 1),
 (225, 1),
 (248, 1),
 (264, 1),
```

```
 (201, 1),
 (325, 1),
 (411, 1),
 (420, 1),
 (562, 1),
 (1481, 1)],
[(6, 1),
 (20, 2),
 (49, 1),
 (94, 1),
 (104, 1),
 (128, 1),
 (146, 1),
 (167, 1),
 (192, 1),
 (268, 2),
 (387, 1),
 (1422, 1)],
[(16, 1),
 (47, 1),
 (179, 1),
 (211, 1),
 (260, 1),
 (322, 1),
 (477, 1),
 (550, 1),
 (640, 1),
 (919, 1)],
[(6, 1), (24, 1), (33, 1), (37, 1), (143, 1), (516, 1), (1482, 1), (1483, 1)],
[(19, 1), (33, 1), (231, 1), (645, 1), (808, 1)],
[(6, 1),
 (52, 1),
 (103, 1),
 (105, 1),
 (126, 1),
 (127, 1),
 (201, 1),
 (211, 1),
 (283, 1),
 (369, 1),
 (425, 1),
 (595, 1),
 (693, 1),
 (848, 1),
 (1020, 1),
 (1044, 1)],
[(13, 1), (33, 1), (49, 1), (318, 1), (385, 1), (496, 1), (918, 1)],
[(6, 2),
 (19, 1),
 (22, 1),
 (48, 1),
 (60, 1),
 (62, 1),
 (65, 5),
 (88, 1),
 (92, 2),
 (103, 2),
 (159, 1),
 (164, 1),
 (235, 1),
 (283, 1),
 (288, 3),
 (295, 1),
 (369, 1),
 (595, 1),
 (614, 2),
 (704, 1),
 (788, 1),
 (791, 1),
 (801, 1),
 (917, 1),
 (984, 1),
 (1484, 1),
 (1485, 1),
 (1486, 1)],
[(16, 1), (79, 1), (93, 1), (176, 1), (179, 1), (197, 1), (201, 1)],
[(18, 1), (37, 1), (176, 1), (179, 1), (240, 1), (272, 1), (547, 1)],
[(20, 2),
 (47, 1),
 (79, 1),
 (88, 1),
 (197, 1),
 (201, 1),
 (292, 1),
 (560, 1),
 (665, 1),
 (1349, 1)],
[(6, 1),
 (49, 1),
 (79, 1),
 (108, 1),
 (159, 2),
 (168, 1),
 (222, 1),
 (262, 1),
 (381, 1),
 (572, 1),
 (637, 1),
 (1039, 1),
 (1211, 2),
 (1487, 1)],
[(0, 1),
 (6, 1),
 (20, 1),
 (35, 1),
 (88, 1),
 (179, 1),
 (203, 1),
 (243, 1),
 (252, 1),
 (279, 1),
 (659, 1),
 (689, 1),
 (1008, 1)],
[(6, 1),
 (22, 1),
```

```
 (33, 1),
 (83, 2),
 (87, 1),
 (109, 1),
 (173, 1),
 (231, 1),
 (458, 1),
 (546, 1),
 (603, 1),
 (1488, 1)],
[(26, 1), (33, 1), (37, 1), (121, 1), (179, 1), (1489, 1)],
[(219, 1), (222, 1), (294, 1)],
[(13, 1),
 (16, 1),
 (33, 3),
 (49, 1),
 (79, 1),
 (179, 1),
 (194, 1),
 (197, 1),
 (234, 1),
 (372, 1),
 (924, 1)],
[(3, 2),
 (16, 1),
 (20, 1),
 (26, 1),
 (83, 1),
 (164, 1),
 (169, 1),
 (185, 1),
 (211, 1),
 (240, 1),
 (308, 1),
 (387, 1),
 (510, 1),
 (614, 1),
 (636, 1),
 (641, 1),
 (729, 1),
 (1224, 1),
 (1225, 1)],
[(64, 1),
 (114, 1),
 (179, 1),
 (185, 1),
 (193, 1),
 (197, 1),
 (254, 1),
 (299, 1),
 (343, 1)],
[(7, 1), (16, 2), (24, 1), (121, 1), (211, 1)],
[(89, 1), (92, 1), (262, 1), (381, 1), (401, 1), (532, 1), (1490, 1)],
[(11, 1),
 (20, 2),
 (36, 1),
 (81, 1),
 (83, 1),
 (90, 1),
 (94, 1),
 (185, 1),
 (207, 1),
 (209, 1),
 (273, 1),
 (300, 1),
 (329, 1),
 (335, 1),
 (391, 1),
 (396, 1),
 (616, 1),
 (622, 2),
 (1148, 1),
 (1446, 1),
 (1491, 1),
 (1492, 1)],
[(48, 2), (62, 1), (354, 1), (387, 1), (554, 1), (945, 1), (1493, 1)],
[(16, 1),
 (22, 1),
 (47, 1),
 (65, 1),
 (87, 1),
 (93, 1),
 (387, 1),
 (406, 1),
 (472, 1),
 (982, 1),
 (984, 1),
 (985, 1),
 (1494, 1)],
[(4, 1),
 (6, 1),
 (11, 1),
 (16, 2),
 (24, 1),
 (29, 1),
 (36, 1),
 (39, 1),
 (47, 1),
 (48, 2),
 (80, 1),
 (87, 1),
 (88, 2),
 (93, 2),
 (96, 1),
 (97, 1),
 (120, 1),
 (134, 1),
 (143, 1),
 (144, 1),
 (164, 1),
 (168, 1),
 (177, 1),
 (217, 1),
 (247, 2),
 (262, 1),
```

```
 (283, 1),
 (324, 1),
 (354, 1),
 (362, 1),
 (409, 1),
 (419, 1),
 (472, 1),
 (516, 2),
 (535, 2),
 (562, 1),
 (631, 1),
 (698, 1),
 (702, 1),
 (725, 1),
 (778, 1),
 (831, 1),
 (863, 1),
 (874, 1),
 (959, 1),
 (1087, 1),
 (1453, 1),
 (1495, 1),
 (1496, 1),
 (1497, 1),
 (1498, 1),
 (1499, 1),
 (1500, 1)],
[(64, 1), (128, 1), (136, 1), (268, 1), (434, 1), (636, 1), (1468, 1)],
[(22, 1),
 (41, 1),
 (92, 1),
 (128, 1),
 (131, 1),
 (146, 1),
 (160, 1),
 (164, 1),
 (216, 1),
 (302, 1),
 (310, 1),
 (472, 1),
 (479, 1),
 (877, 1),
 (1501, 1),
 (1502, 1),
 (1503, 1),
 (1504, 1)],
[(0, 1),
 (19, 2),
 (24, 1),
 (26, 1),
 (42, 1),
 (48, 1),
 (49, 1),
 (51, 1),
 (52, 1),
 (53, 1),
 (80, 2),
 (87, 1),
 (127, 1),
 (160, 1),
 (164, 1),
 (165, 1),
 (179, 1),
 (185, 1),
 (192, 2),
 (207, 1),
 (211, 1),
 (296, 3),
 (307, 1),
 (310, 1),
 (316, 1),
 (359, 1),
 (387, 1),
 (388, 1),
 (405, 1),
 (463, 1),
 (487, 1),
 (567, 1),
 (607, 1),
 (628, 1),
 (634, 1),
 (638, 2),
 (658, 1),
 (670, 1),
 (788, 1),
 (833, 1),
 (918, 1),
 (1023, 1),
 (1093, 1),
 (1177, 1),
 (1505, 1),
 (1506, 1),
 (1507, 1),
 (1508, 1)],
[(6, 1), (49, 1), (62, 1), (92, 1), (114, 1), (139, 1), (193, 2), (343, 1)],
[(11, 1),
 (16, 1),
 (26, 2),
 (45, 1),
 (47, 1),
 (79, 1),
 (81, 1),
 (88, 1),
 (90, 1),
 (94, 1),
 (105, 1),
 (120, 1),
 (179, 1),
 (186, 1),
 (197, 1),
 (201, 1),
 (296, 1),
 (325, 1),
 (373, 1),
 (411, 1),
```

```
 (477, 1),
 (690, 1),
 (730, 1),
 (906, 1),
 (1280, 1),
 (1509, 1),
 (1510, 1)],
[(33, 1),
 (61, 1),
 (114, 1),
 (128, 1),
 (176, 1),
 (179, 2),
 (249, 1),
 (1511, 1)],
[(16, 1), (22, 1), (33, 1), (61, 1), (176, 1), (1230, 1)],
[(0, 1),
 (4, 1),
 (6, 1),
 (16, 2),
 (37, 1),
 (43, 1),
 (47, 1),
 (62, 3),
 (66, 1),
 (87, 1),
 (88, 1),
 (133, 1),
 (143, 1),
 (161, 1),
 (203, 1),
 (217, 1),
 (238, 1),
 (252, 1),
 (255, 1),
 (287, 1),
 (300, 1),
 (387, 1),
 (388, 1),
 (409, 1),
 (452, 1),
 (465, 1),
 (759, 1),
 (1426, 1),
 (1512, 1),
 (1513, 1),
 (1514, 1),
 (1515, 1),
 (1516, 1)],
[(164, 1),
 (201, 1),
 (329, 1),
 (454, 1),
 (456, 1),
 (494, 1),
 (514, 1),
 (535, 1),
 (913, 1),
 (916, 1)],
[(6, 1),
 (13, 1),
 (26, 1),
 (34, 1),
 (36, 1),
 (49, 1),
 (67, 1),
 (89, 1),
 (92, 2),
 (94, 1),
 (114, 2),
 (117, 1),
 (125, 1),
 (136, 2),
 (162, 1),
 (164, 1),
 (185, 1),
 (201, 1),
 (217, 1),
 (277, 1),
 (343, 2),
 (380, 1),
 (410, 1),
 (521, 1),
 (587, 1),
 (595, 1),
 (667, 1),
 (762, 1),
 (1004, 1),
 (1012, 1),
 (1101, 1),
 (1328, 1),
 (1354, 2),
 (1369, 1),
 (1517, 1),
 (1518, 1),
 (1519, 1),
 (1520, 1),
 (1521, 1),
 (1522, 1),
 (1523, 1),
 (1524, 1)],
[(94, 1),
 (153, 1),
 (179, 1),
 (183, 1),
 (193, 1),
 (325, 1),
 (411, 1),
 (636, 1),
 (1166, 1)],
[(36, 1),
 (52, 1),
 (62, 2),
 (79, 1),
 (86, 1),
 (93, 1)
```

```
 (93, 1),
 (103, 1),
 (136, 1),
 (174, 1),
 (201, 1),
 (297, 1),
 (374, 1),
 (419, 1),
 (595, 1),
 (651, 1),
 (672, 1),
 (727, 1),
 (806, 2),
 (1105, 1),
 (1525, 1)],
[(20, 1),
 (33, 1),
 (62, 1),
 (77, 1),
 (94, 2),
 (126, 1),
 (133, 1),
 (136, 1),
 (150, 1),
 (176, 1),
 (179, 1),
 (217, 1),
 (230, 1),
 (263, 1),
 (389, 1),
 (529, 1),
 (611, 1),
 (651, 1),
 (872, 1),
 (999, 1),
 (1029, 1),
 (1526, 1),
 (1527, 1),
 (1528, 1)],
[(33, 1), (127, 1), (193, 1), (312, 1), (368, 1), (537, 1), (1529, 1)],
[(33, 1), (139, 1), (153, 1), (179, 1), (254, 1), (514, 1), (999, 1)],
[(4, 1),
 (19, 1),
 (47, 2),
 (55, 2),
 (56, 2),
 (62, 1),
 (69, 2),
 (92, 1),
 (139, 1),
 (179, 1),
 (213, 1),
 (217, 1),
 (263, 1),
 (306, 1),
 (359, 1),
 (510, 1),
 (691, 1),
 (702, 1),
 (848, 1),
 (1530, 1),
 (1531, 1),
 (1532, 1)],
[(16, 1),
 (24, 1),
 (49, 1),
 (185, 1),
 (192, 1),
 (193, 1),
 (251, 1),
 (324, 1),
 (536, 1),
 (1349, 1),
 (1533, 1)],
[(1, 1),
 (22, 1),
 (34, 1),
 (36, 1),
 (62, 1),
 (63, 1),
 (87, 2),
 (88, 1),
 (103, 1),
 (106, 1),
 (133, 1),
 (164, 1),
 (165, 1),
 (176, 1),
 (200, 1),
 (201, 1),
 (243, 1),
 (247, 1),
 (249, 1),
 (293, 1),
 (300, 1),
 (329, 1),
 (401, 1),
 (432, 1),
 (433, 1),
 (434, 1),
 (435, 1),
 (436, 1),
 (437, 1),
 (438, 1),
 (439, 1),
 (440, 1),
 (441, 1),
 (442, 2),
 (443, 1),
 (444, 1),
 (445, 1),
 (446, 1)],
[(35, 1),
 (36, 1),
 (41, 1),
 (47, 2),
```

```
 (17, 2),
 (48, 1),
 (71, 1),
 (88, 1),
 (164, 1),
 (179, 1),
 (238, 1),
 (286, 1)],
[(24, 1), (179, 1), (225, 1), (233, 1), (244, 1), (1534, 1)],
[(20, 2), (22, 1), (80, 1), (1535, 1), (1536, 1)],
[(24, 1),
 (33, 1),
 (41, 1),
 (62, 1),
 (127, 1),
 (192, 1),
 (237, 1),
 (238, 1),
 (790, 1)],
[(37, 1),
 (78, 1),
 (94, 1),
 (168, 1),
 (181, 1),
 (193, 1),
 (293, 1),
 (587, 1),
 (603, 1),
 (626, 1),
 (641, 1),
 (643, 1),
 (806, 1),
 (870, 1),
 (892, 1),
 (900, 1),
 (938, 1),
 (1037, 1),
 (1039, 2),
 (1263, 1),
 (1332, 1),
 (1537, 1),
 (1538, 1)],
[(94, 1),
 (193, 1),
 (374, 1),
 (380, 1),
 (385, 1),
 (734, 1),
 (956, 1),
 (1539, 1)],
[(22, 1), (98, 1), (128, 1), (179, 2), (197, 1), (514, 1), (1029, 1)],
[(7, 1),
 (15, 1),
 (43, 1),
 (48, 1),
 (79, 1),
 (92, 1),
 (133, 1),
 (143, 1),
 (180, 1),
 (262, 1),
 (312, 1),
 (318, 1),
 (354, 1),
 (373, 1),
 (385, 1),
 (461, 1),
 (517, 1),
 (524, 1),
 (528, 1),
 (554, 1),
 (571, 1),
 (575, 1),
 (611, 1),
 (741, 1),
 (910, 1),
 (1175, 1),
 (1266, 1),
 (1427, 1),
 (1510, 1),
 (1527, 1),
 (1540, 1),
 (1541, 1),
 (1542, 1)],
[(26, 1), (33, 1), (79, 1), (139, 1), (197, 1), (515, 1)],
[(49, 1),
 (53, 1),
 (83, 1),
 (129, 1),
 (157, 1),
 (194, 1),
 (197, 1),
 (205, 1),
 (207, 1),
 (247, 1),
 (914, 1),
 (1048, 1)],
[(3, 1),
 (6, 1),
 (16, 3),
 (33, 1),
 (45, 1),
 (49, 1),
 (51, 1),
 (59, 1),
 (84, 1),
 (87, 1),
 (94, 1),
 (141, 1),
 (143, 1),
 (194, 1),
 (243, 1),
 (318, 1),
 (381, 1),
 (385, 2),
 (440, 1),
```

(494, 1),
(516, 1),
(571, 1),
(586, 1),
(643, 1),
(714, 1),
(716, 2),
(806, 1),
(1365, 1),
(1524, 1),
(1543, 1)],
[(7, 1),
(79, 1),
(133, 1),
(164, 1),
(193, 1),
(197, 1),
(349, 1),
(388, 1),
(1166, 1),
(1451, 1),
(1544, 1)],
[(33, 1),
(127, 1),
(164, 1),
(343, 1),
(354, 1),
(554, 1),
(761, 1),
(1186, 1)],
[(79, 1),
(104, 1),
(114, 1),
(160, 1),
(179, 1),
(192, 1),
(197, 1),
(750, 1),
(1422, 1)],
[(7, 1),
(22, 1),
(33, 1),
(36, 1),
(61, 2),
(79, 1),
(93, 2),
(104, 1),
(128, 1),
(160, 1),
(179, 1),
(200, 1),
(203, 1),
(243, 1),
(806, 1)],
[(16, 1), (34, 1), (48, 1), (419, 1), (532, 1), (1371, 1)],
[(33, 1), (133, 1), (157, 1), (231, 1), (1545, 1), (1546, 1)],
[(19, 1), (53, 1), (263, 2), (331, 1), (613, 1), (702, 1), (1547, 1)],
[(121, 1), (146, 1), (179, 1), (193, 1), (249, 1), (1511, 1)],
[(24, 1), (26, 1), (33, 1), (53, 1), (92, 1), (211, 1), (1548, 1)],
[(41, 1), (83, 1), (90, 1), (92, 1), (150, 1), (374, 1), (667, 1), (1549, 1)],
[(33, 1),
(121, 1),
(173, 1),
(177, 1),
(268, 1),
(295, 1),
(316, 1),
(356, 1),
(374, 1),
(1550, 1),
(1551, 1)],
[(94, 1),
(95, 1),
(150, 1),
(193, 1),
(999, 1),
(1552, 1),
(1553, 1),
(1554, 1),
(1555, 1)],
[(26, 1), (52, 1), (173, 1), (192, 1), (1556, 1)],
[(6, 1), (16, 2), (33, 1), (49, 1), (93, 1), (211, 1), (627, 1), (850, 1)],
[(16, 1),
(19, 1),
(20, 1),
(35, 1),
(47, 2),
(48, 1),
(88, 1),
(172, 1),
(477, 1),
(877, 1),
(1049, 1),
(1088, 1)],
[(62, 1),
(95, 1),
(176, 1),
(177, 1),
(197, 1),
(198, 1),
(591, 1),
(611, 1),
(890, 1),
(1557, 1)],
[(6, 1),
(20, 1),
(47, 1),
(48, 1),
(52, 1),
(56, 1),
(79, 1),
(105, 1),
(164, 1),
(197, 1),
(201, 1),

```
 (225, 1),
 (477, 2),
 (484, 1),
 (603, 1),
 (691, 1),
 (693, 1),
 (994, 1),
 (1047, 1),
 (1049, 1),
 (1088, 1),
 (1151, 1)],
[(53, 1), (79, 1), (197, 1), (225, 1), (802, 1), (1120, 1)],
[(6, 1),
 (20, 1),
 (26, 1),
 (47, 2),
 (52, 2),
 (53, 1),
 (62, 1),
 (80, 1),
 (92, 1),
 (104, 1),
 (106, 1),
 (109, 1),
 (126, 2),
 (143, 1),
 (155, 1),
 (164, 1),
 (173, 1),
 (185, 2),
 (188, 1),
 (201, 1),
 (211, 2),
 (216, 1),
 (249, 1),
 (276, 1),
 (277, 1),
 (294, 1),
 (300, 1),
 (307, 1),
 (308, 1),
 (312, 1),
 (316, 1),
 (378, 3),
 (499, 1),
 (528, 1),
 (536, 1),
 (586, 1),
 (587, 1),
 (613, 1),
 (614, 4),
 (627, 1),
 (649, 1),
 (651, 1),
 (692, 2),
 (700, 1),
 (719, 2),
 (727, 1),
 (740, 1),
 (788, 1),
 (797, 1),
 (799, 2),
 (854, 1),
 (901, 1),
 (990, 1),
 (1177, 1),
 (1211, 1),
 (1328, 1),
 (1549, 1),
 (1558, 1),
 (1559, 1),
 (1560, 1),
 (1561, 1),
 (1562, 1),
 (1563, 2),
 (1564, 1),
 (1565, 1),
 (1566, 3),
 (1567, 1)],
[(6, 1),
 (7, 2),
 (16, 2),
 (19, 1),
 (24, 1),
 (33, 1),
 (36, 1),
 (37, 1),
 (70, 1),
 (79, 1),
 (80, 1),
 (82, 1),
 (83, 1),
 (94, 2),
 (95, 1),
 (126, 1),
 (172, 2),
 (173, 2),
 (184, 2),
 (185, 1),
 (188, 1),
 (211, 1),
 (231, 1),
 (323, 1),
 (357, 1),
 (387, 1),
 (511, 1),
 (742, 1),
 (854, 1),
 (923, 1),
 (945, 1),
 (1230, 1),
 (1456, 1),
 (1568, 1),
 (1569, 1),
 (1570, 1),
```

```
 (1571, 1),
 (1572, 1),
 (1573, 1),
 (1574, 1)],
[(7, 1),
 (94, 1),
 (95, 1),
 (133, 1),
 (179, 1),
 (195, 1),
 (457, 1),
 (511, 1),
 (636, 1),
 (1422, 1)],
[(6, 1),
 (11, 1),
 (16, 1),
 (26, 1),
 (92, 1),
 (133, 1),
 (197, 1),
 (252, 1),
 (1570, 1)],
[(26, 1),
 (37, 1),
 (41, 1),
 (94, 1),
 (179, 1),
 (405, 1),
 (479, 1),
 (759, 1),
 (804, 1),
 (892, 1),
 (1553, 1)],
[(33, 1), (121, 1), (179, 1), (249, 1), (549, 1), (624, 1)],
[(16, 1),
 (33, 1),
 (79, 1),
 (92, 1),
 (211, 1),
 (231, 1),
 (385, 1),
 (388, 1),
 (434, 1),
 (651, 1),
 (937, 1)],
[(6, 1),
 (26, 1),
 (37, 1),
 (47, 1),
 (51, 1),
 (93, 1),
 (94, 1),
 (173, 1),
 (192, 1),
 (299, 1),
 (518, 1),
 (664, 2),
 (1044, 1)],
[(20, 1), (33, 1), (37, 1), (92, 1), (168, 1), (203, 1), (343, 1), (583, 1)],
[(16, 1),
 (33, 1),
 (89, 1),
 (97, 1),
 (103, 1),
 (160, 1),
 (211, 1),
 (480, 1),
 (806, 1),
 (954, 1),
 (1575, 1)],
[(33, 1),
 (98, 1),
 (179, 1),
 (197, 1),
 (230, 1),
 (514, 1),
 (872, 1),
 (1101, 1),
 (1576, 1),
 (1577, 1),
 (1578, 1)],
[(43, 1),
 (79, 1),
 (122, 1),
 (131, 1),
 (179, 1),
 (661, 1),
 (774, 1),
 (839, 1),
 (1579, 1)],
[(6, 1),
 (16, 1),
 (18, 1),
 (20, 1),
 (24, 1),
 (26, 1),
 (77, 1),
 (79, 1),
 (83, 2),
 (150, 1),
 (185, 3),
 (193, 1),
 (205, 1),
 (234, 2),
 (238, 1),
 (244, 1),
 (258, 1),
 (262, 1),
 (316, 2),
 (344, 1),
 (426, 1),
 (433, 1),
 (489, 1),
 (535, 1),
 (540, 1)
```

```
 (540, 1),
 (667, 1),
 (741, 1),
 (747, 1),
 (769, 3),
 (821, 1),
 (922, 1),
 (1036, 1),
 (1143, 2),
 (1328, 1),
 (1336, 1),
 (1389, 4),
 (1390, 4),
 (1566, 2),
 (1580, 1),
 (1581, 1),
 (1582, 1)],
[(16, 1),
 (39, 1),
 (47, 1),
 (65, 1),
 (86, 1),
 (88, 1),
 (99, 1),
 (103, 1),
 (217, 1),
 (262, 1),
 (475, 1),
 (591, 1),
 (781, 1)],
[(16, 1),
 (33, 1),
 (49, 1),
 (103, 1),
 (133, 2),
 (184, 1),
 (187, 1),
 (193, 1),
 (201, 1),
 (249, 1),
 (349, 1),
 (368, 1),
 (733, 1),
 (962, 1),
 (1583, 1)],
[(109, 1),
 (176, 1),
 (197, 1),
 (201, 1),
 (369, 1),
 (374, 1),
 (402, 1),
 (1584, 1)],
[(80, 1),
 (103, 1),
 (164, 2),
 (231, 1),
 (296, 1),
 (399, 1),
 (1251, 1),
 (1585, 1),
 (1586, 1)],
[(62, 1), (207, 1), (419, 1), (702, 1), (1008, 1)],
[(24, 1), (179, 1), (192, 1), (918, 1), (1587, 1)],
[(8, 1),
 (21, 1),
 (90, 1),
 (94, 1),
 (152, 1),
 (164, 1),
 (226, 1),
 (290, 1),
 (302, 1),
 (305, 1),
 (381, 1),
 (395, 1),
 (399, 1),
 (607, 1),
 (765, 1),
 (855, 1),
 (1431, 1)],
[(89, 1), (211, 1), (402, 1), (890, 1), (913, 1)],
[(16, 1),
 (41, 1),
 (179, 1),
 (192, 1),
 (201, 1),
 (343, 1),
 (1588, 1),
 (1589, 1)],
[(16, 3),
 (18, 1),
 (24, 1),
 (128, 1),
 (168, 1),
 (244, 1),
 (318, 1),
 (343, 1),
 (385, 1)],
[(1, 1),
 (6, 1),
 (16, 1),
 (20, 1),
 (22, 1),
 (33, 3),
 (51, 1),
 (52, 1),
 (64, 1),
 (89, 1),
 (95, 1),
 (103, 1),
 (105, 1),
 (114, 2),
 (133, 2),
 (152, 1),
 (154, 2),
```

```
 (167, 1),
 (185, 1),
 (193, 2),
 (195, 1),
 (201, 1),
 (216, 1),
 (231, 1),
 (296, 1),
 (302, 1),
 (316, 1),
 (318, 1),
 (333, 1),
 (346, 1),
 (353, 1),
 (369, 1),
 (385, 2),
 (388, 1),
 (439, 1),
 (458, 2),
 (492, 2),
 (494, 1),
 (510, 1),
 (535, 1),
 (643, 1),
 (663, 1),
 (703, 1),
 (772, 1),
 (773, 1),
 (811, 2),
 (825, 1),
 (826, 1),
 (910, 1),
 (1176, 1),
 (1230, 1),
 (1260, 1),
 (1274, 1),
 (1330, 1),
 (1427, 1),
 (1510, 2),
 (1544, 1),
 (1590, 1),
 (1591, 1),
 (1592, 1),
 (1593, 1),
 (1594, 1),
 (1595, 1),
 (1596, 1),
 (1597, 1),
 (1598, 1),
 (1599, 1),
 (1600, 1),
 (1601, 1),
 (1602, 2),
 (1603, 1)],
[(8, 1),
 (13, 1),
 (29, 1),
 (33, 1),
 (41, 1),
 (45, 1),
 (53, 1),
 (78, 1),
 (87, 1),
 (153, 1),
 (177, 1),
 (201, 1),
 (256, 1),
 (395, 1),
 (433, 1),
 (528, 1),
 (537, 1),
 (675, 1),
 (719, 1),
 (1085, 1),
 (1319, 1),
 (1604, 1)],
[(7, 1), (62, 1), (104, 1), (176, 1), (288, 1), (387, 1), (1605, 1)],
[(20, 3),
 (26, 1),
 (33, 1),
 (94, 1),
 (95, 1),
 (96, 1),
 (97, 1),
 (98, 1),
 (99, 1),
 (100, 1),
 (101, 1),
 (102, 1),
 (103, 1)],
[(6, 1),
 (35, 1),
 (49, 1),
 (59, 1),
 (79, 1),
 (114, 1),
 (179, 2),
 (194, 1),
 (197, 1),
 (343, 1),
 (458, 1),
 (1052, 1),
 (1055, 1),
 (1606, 1)],
[(22, 1),
 (35, 1),
 (36, 1),
 (65, 1),
 (87, 1),
 (154, 1),
 (179, 1),
 (201, 1),
 (292, 1),
 (412, 1),
 (452, 1),
```

```
 (457, 1),
 (458, 1),
 (474, 1),
 (659, 1),
 (1607, 1)],
[(179, 1), (185, 1), (260, 1), (477, 2), (702, 1), (1608, 1)],
[(33, 1), (92, 1), (190, 1), (343, 1)],
[(70, 2), (85, 1), (354, 1), (554, 1), (730, 1), (919, 1)],
[(6, 2),
 (47, 1),
 (62, 1),
 (65, 1),
 (66, 1),
 (71, 1),
 (243, 1),
 (354, 1),
 (357, 1),
 (418, 1),
 (494, 1),
 (544, 1),
 (554, 1),
 (984, 1),
 (1609, 1)],
[(16, 1), (51, 1), (61, 1), (70, 1), (93, 1), (423, 1)],
[(6, 1),
 (12, 1),
 (45, 1),
 (47, 1),
 (62, 2),
 (64, 1),
 (87, 1),
 (92, 1),
 (162, 1),
 (197, 1),
 (316, 1),
 (397, 1),
 (725, 1),
 (1196, 1),
 (1418, 1),
 (1610, 1),
 (1611, 1)],
[(0, 1),
 (57, 1),
 (64, 1),
 (107, 1),
 (179, 1),
 (193, 1),
 (249, 1),
 (357, 1),
 (458, 1),
 (1104, 1),
 (1612, 1),
 (1613, 1)],
[(62, 1), (87, 1), (92, 1), (143, 1), (146, 1), (168, 1), (211, 1), (388, 1)],
[(6, 1),
 (7, 1),
 (19, 1),
 (33, 1),
 (104, 1),
 (138, 1),
 (276, 1),
 (349, 1),
 (369, 1)],
[(20, 1),
 (47, 2),
 (79, 1),
 (88, 1),
 (94, 1),
 (179, 1),
 (341, 1),
 (540, 1),
 (1614, 1)],
[(16, 1),
 (36, 1),
 (43, 1),
 (150, 1),
 (179, 1),
 (217, 1),
 (251, 1),
 (478, 1),
 (678, 1),
 (1546, 1)],
[(20, 2),
 (41, 1),
 (46, 1),
 (47, 1),
 (52, 1),
 (64, 1),
 (65, 1),
 (69, 1),
 (94, 1),
 (105, 1),
 (177, 1),
 (179, 2),
 (185, 1),
 (197, 2),
 (201, 1),
 (318, 1),
 (341, 1),
 (433, 1),
 (549, 1),
 (638, 1),
 (653, 1),
 (691, 1),
 (800, 1),
 (1395, 2),
 (1560, 1),
 (1615, 1),
 (1616, 1)],
[(35, 1),
 (79, 2),
 (161, 1),
 (179, 1),
 (216, 1),
 (325, 1),
```

```
 (358, 1),
 (1248, 1),
 (1617, 1),
 (1618, 1)],
[(262, 1), (314, 1), (324, 1), (1058, 1), (1619, 1)],
[(26, 1), (48, 1), (75, 1), (401, 1), (869, 1)],
[(16, 1), (93, 1), (193, 1), (374, 1), (1039, 1)],
[(47, 1),
 (64, 1),
 (79, 1),
 (117, 1),
 (243, 1),
 (262, 1),
 (302, 1),
 (318, 1),
 (766, 1)],
[(35, 1),
 (79, 1),
 (103, 1),
 (197, 1),
 (627, 1),
 (791, 1),
 (1164, 1),
 (1620, 1)],
[(62, 1),
 (120, 1),
 (155, 1),
 (387, 1),
 (406, 1),
 (417, 1),
 (949, 1),
 (1621, 1)],
[(16, 1), (33, 1), (169, 1), (211, 1), (302, 1), (571, 1)],
[(35, 1), (47, 1), (93, 1), (179, 1), (235, 1), (275, 1), (1261, 1)],
[(92, 1), (93, 1), (419, 1), (900, 1), (1622, 1)],
[(26, 1), (33, 1), (127, 1), (133, 1), (179, 1)],
[(47, 1), (164, 1), (240, 1), (364, 1), (641, 1)],
[(4, 1),
 (80, 1),
 (90, 1),
 (133, 1),
 (179, 1),
 (221, 1),
 (396, 1),
 (429, 1),
 (476, 1),
 (982, 1)],
[(16, 1),
 (33, 2),
 (131, 1),
 (164, 1),
 (506, 1),
 (554, 1),
 (916, 1),
 (917, 1),
 (1623, 1)],
[(13, 1),
 (16, 2),
 (33, 1),
 (94, 1),
 (160, 1),
 (318, 1),
 (385, 1),
 (409, 1),
 (1624, 1)],
[(4, 1),
 (7, 2),
 (13, 1),
 (15, 2),
 (16, 1),
 (22, 1),
 (24, 2),
 (26, 3),
 (28, 1),
 (33, 4),
 (62, 1),
 (79, 1),
 (83, 1),
 (86, 1),
 (87, 2),
 (93, 1),
 (95, 1),
 (96, 1),
 (103, 1),
 (109, 1),
 (110, 3),
 (112, 1),
 (133, 2),
 (150, 1),
 (154, 1),
 (160, 1),
 (176, 1),
 (185, 1),
 (187, 1),
 (203, 2),
 (207, 1),
 (211, 1),
 (243, 1),
 (266, 2),
 (315, 1),
 (318, 1),
 (342, 1),
 (385, 1),
 (391, 1),
 (439, 1),
 (472, 1),
 (485, 1),
 (537, 1),
 (559, 1),
 (592, 1),
 (647, 1),
 (685, 1),
 (780, 1),
 (793, 1),
 (954, 1),
```

```
 (1101, 1),
 (1108, 1),
 (1192, 1),
 (1262, 1),
 (1510, 2),
 (1560, 1),
 (1625, 1),
 (1626, 1),
 (1627, 1),
 (1628, 1),
 (1629, 1),
 (1630, 1)],
[(16, 1),
 (93, 1),
 (94, 1),
 (179, 1),
 (197, 1),
 (201, 1),
 (607, 1),
 (790, 1),
 (1631, 1)],
[(16, 1),
 (20, 1),
 (62, 2),
 (74, 1),
 (79, 1),
 (89, 1),
 (94, 2),
 (111, 1),
 (121, 1),
 (128, 1),
 (136, 1),
 (153, 1),
 (191, 1),
 (211, 1),
 (272, 1),
 (477, 1),
 (478, 1),
 (593, 2),
 (780, 1),
 (800, 1),
 (1419, 1),
 (1510, 1),
 (1632, 1)],
[(36, 1), (61, 1), (160, 1), (179, 2), (300, 1), (492, 1), (777, 1)],
[(19, 1),
 (211, 1),
 (306, 1),
 (519, 1),
 (622, 1),
 (892, 1),
 (1064, 1),
 (1633, 1)],
[(11, 1),
 (22, 1),
 (33, 1),
 (43, 1),
 (65, 1),
 (79, 1),
 (143, 2),
 (144, 1),
 (161, 2),
 (175, 1),
 (179, 1),
 (217, 2),
 (358, 1),
 (535, 2),
 (1454, 1),
 (1634, 1),
 (1635, 1)],
[(6, 1), (49, 1), (179, 1), (343, 1), (1036, 1)],
[(0, 1),
 (8, 1),
 (13, 1),
 (20, 2),
 (46, 1),
 (47, 2),
 (51, 1),
 (64, 2),
 (78, 1),
 (79, 1),
 (94, 1),
 (164, 2),
 (179, 2),
 (184, 1),
 (185, 1),
 (197, 1),
 (201, 1),
 (244, 1),
 (300, 1),
 (344, 1),
 (380, 1),
 (395, 1),
 (396, 2),
 (397, 1),
 (427, 1),
 (472, 1),
 (479, 1),
 (482, 1),
 (510, 1),
 (571, 1),
 (577, 1),
 (614, 1),
 (1071, 1),
 (1636, 1),
 (1637, 1),
 (1638, 1)],
[(11, 1),
 (52, 1),
 (53, 1),
 (64, 1),
 (79, 1),
 (193, 1),
 (230, 1),
 (339, 1),
 (507, 1),
```

```
 (567, 1),
 (618, 1),
 (772, 1),
 (946, 1),
 (1120, 1)],
[(6, 1),
 (16, 1),
 (55, 1),
 (88, 1),
 (197, 1),
 (201, 1),
 (263, 1),
 (702, 1),
 (877, 1),
 (1192, 1)],
[(20, 1), (87, 1), (92, 1), (262, 1), (669, 1), (1212, 1), (1639, 1)],
[(20, 1), (35, 1), (47, 1), (139, 1), (179, 2), (181, 1)],
[(16, 2), (49, 1), (76, 1), (179, 1), (211, 2), (283, 1), (457, 1)],
[(43, 1), (79, 1), (202, 1), (203, 1), (787, 1), (1640, 1)],
[(26, 1),
 (64, 1),
 (74, 1),
 (164, 1),
 (179, 1),
 (192, 1),
 (193, 1),
 (259, 1),
 (296, 1),
 (405, 1),
 (479, 1),
 (564, 1),
 (844, 2),
 (870, 1),
 (962, 1),
 (1641, 1),
 (1642, 1),
 (1643, 1)],
[(6, 1),
 (16, 1),
 (33, 1),
 (65, 1),
 (114, 1),
 (173, 1),
 (197, 1),
 (200, 1),
 (536, 1),
 (914, 1),
 (1141, 1)],
[(37, 1),
 (94, 1),
 (179, 1),
 (193, 1),
 (374, 1),
 (535, 1),
 (547, 1),
 (647, 1),
 (1039, 1),
 (1121, 1)],
[(0, 1),
 (24, 1),
 (79, 2),
 (112, 1),
 (121, 1),
 (161, 1),
 (219, 1),
 (244, 1),
 (535, 1),
 (567, 1),
 (1498, 1),
 (1644, 1),
 (1645, 1)],
[(7, 1), (133, 1), (160, 1), (179, 1), (300, 1), (354, 1), (511, 1)],
[(16, 1), (122, 1), (211, 1), (373, 1), (901, 1), (1280, 1), (1646, 1)],
[(121, 1),
 (179, 1),
 (195, 1),
 (197, 1),
 (201, 1),
 (249, 1),
 (306, 1),
 (936, 1)],
[(22, 1),
 (65, 1),
 (92, 2),
 (143, 1),
 (161, 1),
 (179, 1),
 (207, 1),
 (238, 1),
 (243, 1),
 (377, 1)],
[(16, 3),
 (33, 1),
 (94, 1),
 (97, 1),
 (139, 1),
 (153, 1),
 (487, 1),
 (636, 1),
 (671, 1)],
[(16, 1),
 (35, 1),
 (103, 1),
 (108, 1),
 (232, 1),
 (262, 3),
 (325, 1),
 (595, 1),
 (608, 1),
 (648, 1),
 (691, 1),
 (702, 1),
 (962, 1),
 (1159, 1),
 (1647, 1)],
[(91, 1), (316, 1), (388, 1), (1648, 1), (1649, 1)],
```

```
[(91, 1), (518, 1), (588, 1), (1018, 1), (1019, 1)],
[(6, 1),
 (179, 1),
 (193, 1),
 (252, 1),
 (306, 1),
 (461, 1),
 (496, 1),
 (1650, 1)],
[(33, 1), (103, 1), (121, 1), (478, 1), (1651, 1)],
[(0, 1),
 (21, 1),
 (22, 1),
 (33, 1),
 (79, 1),
 (104, 1),
 (128, 1),
 (139, 1),
 (153, 1),
 (167, 1),
 (200, 2),
 (514, 1),
 (759, 1),
 (762, 1),
 (1288, 1),
 (1652, 1),
 (1653, 1),
 (1654, 1)],
[(77, 1), (94, 1), (143, 1), (192, 1), (429, 1), (470, 1), (725, 1)],
[(33, 1), (160, 1), (179, 1), (197, 1), (785, 1), (872, 1), (914, 1)],
[(16, 1), (79, 1), (92, 1), (262, 1), (355, 1), (577, 1), (1655, 1)],
[(155, 1), (222, 1), (325, 1), (329, 1), (1656, 1), (1657, 1)],
[(16, 1), (62, 1), (159, 1), (248, 1), (294, 1), (420, 1)],
[(33, 1),
 (51, 1),
 (52, 1),
 (74, 1),
 (79, 1),
 (80, 1),
 (94, 1),
 (114, 1),
 (164, 1),
 (175, 2),
 (185, 1),
 (264, 1),
 (268, 1),
 (275, 1),
 (286, 1),
 (333, 1),
 (343, 1),
 (387, 1),
 (615, 1),
 (631, 1),
 (790, 1),
 (959, 1),
 (1159, 1),
 (1331, 1),
 (1423, 1),
 (1484, 1),
 (1658, 1),
 (1659, 1),
 (1660, 1),
 (1661, 1),
 (1662, 1)],
[(19, 1),
 (26, 2),
 (41, 1),
 (171, 1),
 (192, 2),
 (791, 1),
 (1143, 1),
 (1663, 1),
 (1664, 1),
 (1665, 1),
 (1666, 1)],
[(20, 1),
 (34, 1),
 (52, 1),
 (80, 1),
 (94, 1),
 (181, 1),
 (193, 3),
 (205, 2),
 (215, 1),
 (359, 1),
 (499, 1),
 (508, 1),
 (771, 1),
 (1062, 1),
 (1389, 1),
 (1532, 1),
 (1667, 1)],
[(4, 1),
 (36, 1),
 (62, 1),
 (79, 1),
 (197, 1),
 (198, 1),
 (211, 1),
 (224, 1),
 (247, 2),
 (750, 1),
 (919, 1),
 (1171, 1)],
[(6, 1),
 (7, 2),
 (11, 1),
 (65, 1),
 (92, 1),
 (99, 1),
 (143, 1),
 (168, 1),
 (211, 1),
 (243, 2),
 (246, 1),
 (252, 1),
```

```
 (295, 1),
 (324, 1),
 (368, 1),
 (485, 1),
 (511, 1),
 (537, 1),
 (559, 1),
 (1668, 1),
 (1669, 1)],
[(20, 1),
 (33, 1),
 (49, 1),
 (65, 1),
 (79, 1),
 (80, 1),
 (83, 1),
 (121, 1),
 (157, 1),
 (179, 1),
 (191, 1),
 (194, 2),
 (200, 1),
 (203, 1),
 (211, 2),
 (277, 2),
 (299, 1),
 (385, 1),
 (387, 1),
 (458, 1),
 (496, 1),
 (1670, 1),
 (1671, 1),
 (1672, 1)],
[(16, 1),
 (20, 1),
 (26, 1),
 (53, 1),
 (64, 2),
 (92, 2),
 (98, 1),
 (105, 2),
 (113, 1),
 (136, 1),
 (165, 1),
 (184, 1),
 (193, 4),
 (211, 1),
 (262, 1),
 (316, 1),
 (343, 1),
 (475, 1),
 (508, 2),
 (540, 1),
 (721, 1),
 (785, 2),
 (802, 1),
 (821, 1),
 (1046, 1),
 (1418, 1),
 (1438, 1),
 (1527, 1),
 (1673, 1)],
[(4, 1),
 (6, 1),
 (16, 2),
 (45, 1),
 (69, 1),
 (103, 1),
 (105, 1),
 (112, 2),
 (114, 2),
 (122, 1),
 (143, 1),
 (162, 1),
 (175, 1),
 (193, 1),
 (213, 1),
 (244, 1),
 (314, 1),
 (318, 1),
 (343, 1),
 (385, 1),
 (442, 2),
 (496, 1),
 (540, 1),
 (554, 1),
 (616, 1),
 (643, 2),
 (672, 1),
 (674, 1),
 (719, 1),
 (731, 2),
 (774, 1),
 (775, 1),
 (836, 1),
 (869, 1),
 (1080, 1),
 (1148, 1),
 (1218, 1),
 (1674, 1),
 (1675, 1),
 (1676, 1),
 (1677, 1),
 (1678, 1),
 (1679, 1),
 (1680, 1),
 (1681, 1),
 (1682, 1),
 (1683, 1),
 (1684, 1),
 (1685, 1),
 (1686, 1),
 (1687, 1)],
[(62, 1),
 (79, 1),
```

```
  (87, 1),
  (197, 1),
  (201, 1),
  (247, 2),
  (270, 1),
  (343, 1),
  (387, 1),
  (444, 2),
  (730, 1)],
 [(4, 1),
  (47, 1),
  (88, 1),
  (94, 1),
  (105, 1),
  (108, 5),
  (202, 1),
  (217, 1),
  (243, 1),
  (373, 1),
  (702, 4),
  (778, 1),
  (976, 1),
  (1029, 1),
  (1047, 1),
  (1688, 1),
  (1689, 1)],
 [(79, 1),
  (94, 1),
  (139, 1),
  (153, 1),
  (179, 1),
  (197, 1),
  (636, 1),
  (872, 1),
  (1690, 1)],
 [(19, 1),
  (62, 1),
  (79, 1),
  (94, 1),
  (114, 1),
  (164, 1),
  (167, 1),
  (179, 1),
  (197, 1),
  (759, 1),
  (762, 1),
  (806, 1),
  (993, 1),
  (1068, 1),
  (1691, 1),
  (1692, 1),
  (1693, 1)],
 [(6, 1), (79, 1), (105, 1), (107, 1), (193, 1), (197, 1)],
 [(39, 1),
  (47, 5),
  (87, 4),
  (90, 1),
  (92, 1),
  (94, 1),
  (166, 1),
  (172, 1),
  (194, 1),
  (217, 1),
  (244, 1),
  (258, 1),
  (316, 1),
  (401, 1),
  (448, 1),
  (554, 1),
  (641, 1),
  (855, 1),
  (1343, 1)],
 [(13, 1),
  (22, 1),
  (83, 1),
  (109, 1),
  (237, 1),
  (262, 1),
  (374, 1),
  (385, 1),
  (496, 1),
  (643, 1),
  (1187, 1),
  (1543, 1),
  (1694, 1),
  (1695, 1),
  (1696, 1),
  (1697, 1)],
 [(33, 1),
  (51, 1),
  (70, 1),
  (110, 1),
  (133, 1),
  (140, 1),
  (160, 1),
  (179, 1),
  (184, 2),
  (230, 1),
  (434, 1),
  (549, 1),
  (666, 1),
  (684, 1),
  (836, 1),
  (901, 1),
  (919, 1),
  (970, 1),
  (1244, 1),
  (1407, 1),
  (1514, 1),
  (1698, 1),
  (1699, 1)],
 [(33, 1), (139, 1), (153, 1), (176, 1), (179, 1), (603, 1), (1700, 1)],
 [(33, 1),
  (52, 1),
  (80, 1),
```

```
 (106, 1),
 (164, 1),
 (458, 1),
 (661, 1),
 (1343, 1),
 (1402, 1),
 (1701, 1)],
[(48, 2),
 (79, 1),
 (175, 1),
 (185, 1),
 (193, 1),
 (238, 1),
 (262, 1),
 (343, 1),
 (461, 2),
 (472, 1),
 (651, 1),
 (713, 1),
 (721, 1),
 (784, 1),
 (962, 1),
 (1702, 1),
 (1703, 1),
 (1704, 1)],
[(20, 1),
 (22, 1),
 (87, 1),
 (128, 1),
 (177, 1),
 (179, 1),
 (195, 1),
 (238, 1),
 (258, 1),
 (913, 1),
 (1629, 1)],
[(16, 1),
 (20, 1),
 (47, 1),
 (48, 1),
 (65, 2),
 (71, 1),
 (79, 1),
 (87, 1),
 (88, 1),
 (103, 2),
 (130, 1),
 (164, 1),
 (264, 1),
 (294, 1),
 (400, 1),
 (410, 1),
 (781, 1),
 (1321, 1)],
[(256, 1), (324, 1), (426, 1), (1705, 1)],
[(20, 1),
 (24, 1),
 (33, 1),
 (77, 1),
 (94, 1),
 (112, 1),
 (164, 1),
 (192, 1),
 (231, 1),
 (316, 1),
 (662, 1),
 (1187, 1)],
[(47, 1), (168, 1), (211, 1), (387, 2), (420, 1), (1706, 1)],
[(25, 1),
 (62, 1),
 (185, 1),
 (197, 1),
 (219, 1),
 (262, 1),
 (427, 1),
 (488, 1),
 (627, 1),
 (702, 1),
 (1029, 1),
 (1159, 1)],
[(20, 1),
 (90, 1),
 (97, 1),
 (133, 1),
 (153, 1),
 (169, 1),
 (239, 1),
 (316, 1),
 (446, 1),
 (452, 1),
 (480, 1),
 (523, 1),
 (624, 1),
 (671, 1),
 (855, 1),
 (1707, 1),
 (1708, 1),
 (1709, 1)],
[(16, 1),
 (20, 1),
 (33, 1),
 (49, 1),
 (57, 1),
 (61, 2),
 (88, 1),
 (92, 1),
 (94, 1),
 (150, 1),
 (164, 2),
 (173, 1),
 (177, 1),
 (179, 2),
 (193, 1),
 (194, 1),
 (211, 1),
 (870, 1)
```

```
 (870, 1),
 (1328, 1),
 (1432, 1),
 (1697, 1),
 (1710, 1)],
[(47, 1),
 (55, 1),
 (92, 1),
 (94, 1),
 (108, 1),
 (154, 1),
 (155, 1),
 (231, 1),
 (275, 1),
 (286, 1),
 (288, 1),
 (357, 1),
 (510, 1),
 (806, 1)],
[(16, 1),
 (79, 1),
 (139, 1),
 (161, 1),
 (201, 2),
 (211, 1),
 (265, 1),
 (351, 1),
 (1029, 2)],
[(5, 1),
 (22, 1),
 (46, 1),
 (57, 1),
 (83, 1),
 (143, 1),
 (155, 1),
 (159, 1),
 (164, 2),
 (168, 2),
 (178, 1),
 (179, 2),
 (201, 1),
 (217, 1),
 (258, 1),
 (263, 1),
 (316, 1),
 (322, 1),
 (325, 1),
 (332, 1),
 (351, 1),
 (410, 1),
 (420, 2),
 (421, 1),
 (628, 1),
 (678, 1),
 (689, 1),
 (691, 1),
 (702, 1),
 (719, 1),
 (788, 1),
 (857, 1),
 (1309, 2),
 (1331, 1),
 (1648, 1),
 (1711, 1),
 (1712, 1),
 (1713, 1),
 (1714, 1),
 (1715, 1)],
[(19, 1),
 (22, 1),
 (24, 1),
 (37, 1),
 (95, 1),
 (97, 1),
 (133, 1),
 (175, 1),
 (177, 1),
 (179, 1),
 (216, 1),
 (236, 1),
 (258, 1),
 (262, 1),
 (306, 1),
 (308, 1),
 (391, 1),
 (399, 1),
 (507, 1),
 (535, 1),
 (549, 1),
 (587, 1),
 (645, 1),
 (765, 1),
 (812, 1),
 (1011, 1),
 (1115, 2),
 (1162, 1),
 (1399, 1),
 (1716, 1),
 (1717, 1),
 (1718, 1),
 (1719, 1),
 (1720, 1),
 (1721, 1),
 (1722, 1),
 (1723, 1),
 (1724, 1),
 (1725, 1),
 (1726, 1),
 (1727, 1)],
[(41, 1), (92, 1), (179, 1), (211, 1), (628, 1), (645, 1)],
[(16, 1),
 (33, 1),
 (61, 1),
 (79, 1),
 (94, 1),
 (108, 1),
```

```
 (100, 1),
 (120, 1),
 (191, 1),
 (197, 2),
 (263, 1),
 (286, 1),
 (300, 1),
 (318, 1),
 (329, 1),
 (349, 1),
 (385, 1),
 (514, 1),
 (629, 2),
 (780, 1),
 (1728, 1)],
[(1, 2),
 (16, 2),
 (62, 1),
 (74, 1),
 (79, 2),
 (80, 2),
 (92, 1),
 (94, 1),
 (108, 2),
 (115, 1),
 (122, 1),
 (150, 1),
 (162, 1),
 (211, 1),
 (215, 1),
 (230, 1),
 (234, 1),
 (300, 1),
 (304, 1),
 (318, 1),
 (346, 1),
 (385, 1),
 (387, 1),
 (524, 1),
 (535, 1),
 (553, 1),
 (603, 1),
 (614, 2),
 (752, 1),
 (771, 1),
 (774, 2),
 (1020, 1),
 (1121, 2),
 (1285, 1),
 (1729, 1),
 (1730, 1)],
[(16, 1),
 (26, 1),
 (51, 1),
 (52, 2),
 (94, 1),
 (103, 1),
 (127, 1),
 (136, 1),
 (143, 1),
 (153, 3),
 (161, 1),
 (181, 1),
 (185, 1),
 (193, 3),
 (215, 4),
 (344, 1),
 (433, 1),
 (541, 1),
 (581, 1),
 (643, 4),
 (651, 1),
 (667, 1),
 (703, 1),
 (706, 1),
 (709, 1),
 (757, 1),
 (848, 1),
 (943, 1),
 (1026, 1),
 (1105, 2),
 (1152, 1),
 (1731, 1),
 (1732, 1)],
[(6, 1),
 (33, 4),
 (39, 1),
 (49, 1),
 (62, 1),
 (83, 1),
 (87, 1),
 (111, 1),
 (131, 1),
 (141, 1),
 (152, 2),
 (163, 2),
 (168, 1),
 (194, 1),
 (319, 2),
 (346, 2),
 (397, 1),
 (492, 2),
 (598, 1),
 (614, 1),
 (638, 1),
 (845, 1),
 (899, 1),
 (962, 1),
 (1234, 2),
 (1262, 1),
 (1386, 1),
 (1435, 2),
 (1484, 1),
 (1733, 1)],
[(24, 1),
 (92, 2),
```

```
 (143, 1),
 (161, 1),
 (185, 1),
 (385, 1),
 (494, 1),
 (603, 1)],
[(16, 1), (26, 2), (52, 1), (78, 1), (193, 1), (254, 1), (999, 1), (1553, 1)],
[(7, 1),
 (61, 1),
 (83, 1),
 (125, 1),
 (133, 1),
 (164, 3),
 (221, 1),
 (368, 1),
 (458, 1),
 (535, 1),
 (587, 2),
 (616, 1),
 (838, 1),
 (954, 2),
 (1509, 1),
 (1734, 1),
 (1735, 1)],
[(13, 1),
 (26, 1),
 (79, 1),
 (101, 1),
 (114, 1),
 (179, 1),
 (197, 1),
 (198, 1),
 (262, 1),
 (331, 1),
 (914, 1),
 (1288, 1)],
[(6, 1),
 (33, 1),
 (34, 1),
 (45, 1),
 (49, 1),
 (80, 1),
 (85, 1),
 (89, 1),
 (133, 2),
 (164, 1),
 (165, 1),
 (176, 1),
 (179, 1),
 (201, 1),
 (268, 1),
 (296, 1),
 (302, 1),
 (496, 1),
 (662, 1),
 (785, 1),
 (979, 1),
 (1181, 1),
 (1182, 1),
 (1183, 1)],
[(19, 1),
 (33, 1),
 (103, 1),
 (111, 1),
 (128, 1),
 (202, 1),
 (231, 1),
 (478, 1),
 (945, 1),
 (1398, 1)],
[(7, 1),
 (33, 1),
 (97, 1),
 (103, 1),
 (128, 1),
 (139, 1),
 (153, 1),
 (164, 1),
 (177, 1),
 (179, 1),
 (458, 1),
 (627, 1),
 (671, 1),
 (806, 1)],
[(6, 1),
 (93, 1),
 (112, 1),
 (133, 1),
 (183, 1),
 (201, 1),
 (237, 1),
 (386, 1),
 (425, 1),
 (539, 1),
 (544, 1),
 (892, 1)],
[(7, 1),
 (16, 1),
 (26, 1),
 (45, 1),
 (79, 2),
 (80, 2),
 (91, 1),
 (202, 1),
 (305, 1),
 (385, 1),
 (399, 2),
 (535, 1),
 (553, 1),
 (643, 1)],
[(62, 1),
 (89, 1),
 (93, 1),
 (96, 1),
 (146, 1),
```

```
 (211, 1),
 (231, 1),
 (945, 1),
 (1736, 1),
 (1737, 1)],
[(16, 1),
 (26, 1),
 (79, 2),
 (105, 2),
 (159, 1),
 (171, 1),
 (197, 1),
 (201, 1),
 (231, 1),
 (262, 1),
 (299, 1),
 (343, 1),
 (433, 1),
 (627, 1),
 (821, 1),
 (1557, 1),
 (1738, 1),
 (1739, 1)],
[(1, 1),
 (16, 1),
 (52, 1),
 (62, 1),
 (69, 1),
 (90, 1),
 (133, 5),
 (150, 1),
 (152, 2),
 (177, 1),
 (179, 1),
 (180, 1),
 (197, 1),
 (201, 2),
 (247, 1),
 (333, 1),
 (373, 1),
 (420, 1),
 (446, 1),
 (571, 1),
 (603, 1),
 (720, 1),
 (743, 1),
 (785, 1),
 (872, 1),
 (890, 1),
 (913, 1),
 (959, 1),
 (1053, 1),
 (1385, 1),
 (1740, 1)],
[(16, 1), (195, 1), (354, 1), (554, 1), (890, 1), (1085, 1), (1231, 1)],
[(16, 1),
 (33, 1),
 (62, 1),
 (87, 1),
 (179, 1),
 (191, 1),
 (194, 1),
 (213, 1),
 (850, 1),
 (1741, 1)],
[(16, 1), (133, 1), (183, 1), (192, 1), (458, 1)],
[(22, 1),
 (36, 1),
 (49, 1),
 (56, 1),
 (90, 1),
 (128, 1),
 (179, 1),
 (202, 1),
 (296, 1),
 (325, 1),
 (420, 1),
 (478, 1),
 (1029, 1),
 (1192, 1),
 (1742, 1)],
[(7, 2),
 (20, 1),
 (33, 2),
 (62, 1),
 (79, 1),
 (85, 1),
 (87, 1),
 (94, 1),
 (103, 1),
 (104, 1),
 (120, 1),
 (133, 2),
 (164, 1),
 (201, 1),
 (243, 1),
 (262, 1),
 (297, 1),
 (306, 1),
 (312, 1),
 (316, 1),
 (343, 1),
 (410, 2),
 (512, 2),
 (800, 1),
 (906, 1),
 (1023, 1),
 (1163, 1),
 (1743, 1),
 (1744, 1),
 (1745, 1),
 (1746, 1),
 (1747, 1),
 (1748, 1),
 (1749, 1)],
```

```
[(34, 1),
 (75, 1),
 (80, 1),
 (164, 1),
 (177, 1),
 (179, 1),
 (627, 1),
 (636, 1),
 (647, 1),
 (650, 1),
 (848, 2),
 (901, 1),
 (1079, 1)],
[(90, 1), (179, 1), (225, 1), (302, 1), (343, 1)],
[(37, 1),
 (87, 1),
 (122, 1),
 (179, 1),
 (193, 1),
 (262, 1),
 (319, 1),
 (408, 1),
 (492, 1),
 (547, 1),
 (626, 1),
 (774, 1),
 (1234, 1)],
[(15, 1),
 (47, 1),
 (79, 1),
 (93, 2),
 (168, 1),
 (211, 1),
 (420, 1),
 (525, 1),
 (667, 1),
 (740, 1),
 (752, 1),
 (805, 1),
 (1263, 1),
 (1544, 1),
 (1750, 1),
 (1751, 1)],
[(6, 1),
 (16, 1),
 (46, 1),
 (47, 2),
 (69, 1),
 (79, 1),
 (88, 2),
 (186, 1),
 (197, 1),
 (211, 1),
 (243, 1),
 (373, 2),
 (664, 1),
 (678, 1),
 (788, 1),
 (939, 1),
 (1190, 1),
 (1191, 1),
 (1192, 1)],
[(4, 2),
 (6, 1),
 (26, 1),
 (35, 1),
 (45, 1),
 (61, 1),
 (92, 1),
 (127, 1),
 (143, 2),
 (161, 1),
 (217, 3),
 (225, 1),
 (250, 1),
 (252, 1),
 (264, 1),
 (316, 1),
 (358, 1),
 (367, 1),
 (387, 1),
 (406, 1),
 (461, 1),
 (1166, 1),
 (1343, 1),
 (1752, 1),
 (1753, 1)],
[(16, 1), (79, 2), (80, 1), (83, 1), (197, 1), (207, 1), (247, 1), (1442, 1)],
[(16, 1), (33, 1), (139, 1), (153, 1), (160, 1), (458, 1)],
[(92, 1), (143, 1), (168, 1), (197, 1), (211, 1), (243, 1), (914, 1)],
[(20, 1),
 (26, 1),
 (37, 1),
 (109, 1),
 (110, 1),
 (136, 1),
 (153, 1),
 (179, 1),
 (800, 1),
 (1754, 1)],
[(94, 1),
 (257, 1),
 (268, 1),
 (271, 1),
 (343, 1),
 (344, 1),
 (1349, 1),
 (1350, 1),
 (1755, 1)],
[(20, 1),
 (62, 1),
 (79, 1),
 (139, 1),
 (179, 1),
 (373, 1),
 (387, 1),
```

```
 (387, 1),
 (710, 1),
 (1756, 1)],
[(39, 1),
 (49, 1),
 (51, 1),
 (52, 1),
 (59, 1),
 (83, 2),
 (89, 1),
 (94, 2),
 (103, 2),
 (104, 1),
 (146, 1),
 (165, 1),
 (173, 1),
 (193, 1),
 (211, 1),
 (225, 1),
 (308, 1),
 (348, 1),
 (387, 1),
 (454, 1),
 (456, 1),
 (479, 1),
 (543, 1),
 (546, 1),
 (667, 1),
 (740, 1),
 (741, 1),
 (778, 1),
 (866, 1),
 (890, 1),
 (906, 1),
 (946, 1),
 (1004, 1),
 (1053, 1),
 (1336, 1),
 (1757, 1),
 (1758, 1),
 (1759, 1)],
[(22, 1),
 (35, 2),
 (62, 2),
 (79, 3),
 (92, 1),
 (103, 1),
 (128, 1),
 (164, 1),
 (188, 1),
 (222, 1),
 (262, 1),
 (264, 1),
 (312, 1),
 (325, 2),
 (331, 1),
 (411, 2),
 (468, 1),
 (571, 1),
 (841, 1),
 (1296, 1),
 (1309, 1),
 (1550, 1),
 (1715, 1),
 (1760, 1),
 (1761, 1),
 (1762, 1)],
[(22, 1),
 (128, 1),
 (152, 1),
 (179, 1),
 (192, 1),
 (346, 1),
 (410, 1),
 (562, 1),
 (845, 1),
 (1667, 1)],
[(24, 1),
 (80, 1),
 (83, 1),
 (85, 1),
 (103, 1),
 (133, 2),
 (140, 1),
 (155, 1),
 (164, 1),
 (211, 1),
 (368, 1),
 (387, 1),
 (399, 1),
 (629, 1),
 (772, 1),
 (1183, 1),
 (1395, 1),
 (1506, 1),
 (1638, 1)],
[(75, 1), (133, 1), (160, 1), (193, 1), (250, 1), (535, 1), (1279, 1)],
[(6, 1),
 (19, 1),
 (22, 1),
 (33, 1),
 (92, 1),
 (94, 1),
 (128, 2),
 (139, 1),
 (153, 1),
 (155, 1),
 (173, 1),
 (184, 1),
 (255, 1),
 (525, 1),
 (679, 1),
 (1457, 1),
 (1763, 1),
 (1764, 1),
 (1765, 1)],
```

```
 (1769, 1)],
[(41, 1), (51, 1), (179, 1), (195, 1), (227, 1)],
[(86, 1), (268, 1), (343, 1), (510, 1), (762, 1), (806, 1), (1766, 1)],
[(79, 1), (83, 1), (133, 1), (197, 1), (207, 1), (391, 1), (1767, 1)],
[(16, 1),
 (70, 1),
 (179, 1),
 (217, 1),
 (630, 1),
 (714, 1),
 (827, 1),
 (1768, 1)],
[(0, 1),
 (4, 1),
 (15, 1),
 (16, 1),
 (22, 2),
 (33, 1),
 (64, 1),
 (77, 1),
 (79, 1),
 (80, 2),
 (83, 2),
 (89, 1),
 (90, 2),
 (92, 1),
 (103, 3),
 (105, 1),
 (114, 1),
 (150, 1),
 (164, 2),
 (168, 2),
 (173, 1),
 (179, 2),
 (181, 1),
 (193, 1),
 (198, 1),
 (230, 1),
 (264, 1),
 (380, 1),
 (387, 1),
 (389, 1),
 (517, 1),
 (821, 1),
 (897, 1),
 (1093, 1),
 (1122, 1),
 (1484, 1),
 (1670, 1),
 (1769, 1),
 (1770, 1),
 (1771, 1),
 (1772, 1)],
[(22, 1),
 (36, 1),
 (43, 1),
 (47, 1),
 (65, 1),
 (69, 1),
 (79, 2),
 (88, 1),
 (97, 1),
 (120, 1),
 (154, 1),
 (179, 1),
 (202, 1),
 (262, 1),
 (286, 1),
 (406, 1),
 (477, 1),
 (648, 1),
 (678, 1),
 (762, 1),
 (836, 1),
 (1029, 1),
 (1096, 1),
 (1281, 1),
 (1390, 1),
 (1773, 1),
 (1774, 1),
 (1775, 1)],
[(6, 2),
 (36, 2),
 (94, 1),
 (103, 3),
 (139, 3),
 (153, 5),
 (175, 1),
 (179, 2),
 (193, 1),
 (195, 2),
 (201, 2),
 (211, 1),
 (349, 2),
 (458, 1),
 (577, 1),
 (629, 1),
 (752, 1),
 (936, 1),
 (1776, 1),
 (1777, 1)],
[(77, 1), (94, 1), (133, 1), (176, 1), (575, 1), (1183, 1), (1391, 1)],
[(0, 1),
 (35, 1),
 (64, 1),
 (66, 1),
 (79, 1),
 (179, 2),
 (203, 1),
 (223, 1),
 (238, 1),
 (292, 1),
 (689, 1),
 (984, 1),
 (1778, 1)],
[(154, 1), (188, 1), (249, 1), (389, 1), (1779, 1), (1780, 1), (1781, 1)],
```

```
[(16, 1),
 (47, 1),
 (93, 1),
 (97, 1),
 (133, 1),
 (168, 1),
 (179, 1),
 (219, 1),
 (243, 1),
 (270, 1),
 (373, 1),
 (827, 1),
 (978, 1),
 (1029, 1),
 (1045, 1),
 (1046, 1),
 (1077, 1),
 (1280, 1)],
[(79, 1),
 (93, 1),
 (177, 1),
 (178, 1),
 (188, 1),
 (197, 1),
 (1782, 1),
 (1783, 1),
 (1784, 1)],
[(16, 1),
 (26, 1),
 (34, 1),
 (139, 1),
 (153, 1),
 (197, 1),
 (496, 1),
 (679, 1),
 (902, 1),
 (1457, 1),
 (1785, 1)],
[(6, 1),
 (16, 2),
 (33, 1),
 (36, 1),
 (65, 1),
 (104, 1),
 (133, 1),
 (179, 1),
 (200, 1),
 (252, 1)],
[(16, 1),
 (35, 1),
 (65, 1),
 (79, 2),
 (179, 1),
 (217, 1),
 (267, 1),
 (331, 1),
 (691, 1),
 (1786, 1)],
[(16, 1),
 (61, 1),
 (69, 1),
 (79, 1),
 (334, 1),
 (434, 1),
 (1276, 1),
 (1544, 1),
 (1787, 1),
 (1788, 1),
 (1789, 1),
 (1790, 1)],
[(16, 1), (24, 1), (33, 1), (160, 1), (179, 1)],
[(79, 2), (150, 1), (164, 1), (230, 1), (559, 1)],
[(22, 1),
 (47, 1),
 (65, 1),
 (80, 1),
 (89, 1),
 (125, 1),
 (159, 1),
 (161, 2),
 (164, 1),
 (177, 1),
 (185, 1),
 (197, 1),
 (306, 1),
 (591, 1),
 (785, 1),
 (799, 2),
 (890, 1),
 (897, 1),
 (1136, 1),
 (1346, 1),
 (1394, 1),
 (1791, 1),
 (1792, 1)],
[(19, 1),
 (22, 1),
 (79, 1),
 (85, 1),
 (92, 1),
 (103, 2),
 (120, 1),
 (128, 1),
 (142, 1),
 (201, 1),
 (211, 1),
 (309, 1),
 (343, 1),
 (410, 1),
 (458, 2),
 (543, 1),
 (544, 1),
 (548, 1),
 (647, 1),
 (747, 1),
```

```
 (865, 1),
 (872, 1),
 (1237, 1),
 (1326, 1),
 (1464, 1),
 (1793, 1),
 (1794, 1)],
[(94, 1), (126, 1), (192, 1), (211, 1), (492, 1), (587, 1)],
[(16, 1), (95, 1), (252, 1), (461, 1), (872, 1)],
[(11, 1),
 (20, 1),
 (48, 1),
 (56, 1),
 (65, 1),
 (66, 1),
 (72, 1),
 (79, 1),
 (94, 1),
 (105, 1),
 (120, 1),
 (168, 2),
 (179, 1),
 (260, 1),
 (265, 1),
 (390, 1),
 (411, 1),
 (464, 1),
 (477, 1),
 (691, 1),
 (693, 1),
 (1778, 1),
 (1795, 1),
 (1796, 1)],
[(79, 1),
 (92, 1),
 (105, 2),
 (107, 1),
 (255, 1),
 (300, 1),
 (343, 1),
 (387, 1),
 (624, 1),
 (773, 1),
 (824, 1),
 (825, 1),
 (826, 1)],
[(16, 1), (26, 1), (109, 1), (648, 1), (1797, 1)],
[(41, 1), (79, 1), (80, 1), (105, 1), (283, 1), (759, 1), (1798, 1)],
[(61, 1),
 (69, 1),
 (92, 1),
 (162, 1),
 (188, 1),
 (308, 1),
 (373, 1),
 (570, 1),
 (836, 1),
 (1432, 1)],
[(0, 1),
 (20, 1),
 (22, 1),
 (64, 1),
 (70, 1),
 (121, 1),
 (154, 1),
 (179, 1),
 (231, 1),
 (272, 1),
 (372, 1),
 (388, 1),
 (689, 1),
 (788, 1),
 (869, 1),
 (1401, 1),
 (1670, 1),
 (1799, 1)],
[(6, 1),
 (33, 1),
 (92, 2),
 (173, 1),
 (184, 1),
 (343, 1),
 (442, 1),
 (1159, 1),
 (1800, 1)],
[(6, 2),
 (20, 1),
 (37, 1),
 (45, 1),
 (49, 1),
 (79, 1),
 (86, 1),
 (93, 2),
 (94, 3),
 (129, 1),
 (136, 1),
 (139, 1),
 (153, 1),
 (160, 1),
 (176, 1),
 (177, 1),
 (183, 1),
 (193, 1),
 (216, 1),
 (258, 1),
 (262, 2),
 (310, 1),
 (374, 3),
 (442, 1),
 (499, 1),
 (578, 1),
 (627, 1),
 (650, 1),
 (671, 1),
 (683, 2),
```

```
 (728, 1),
 (734, 1),
 (854, 2),
 (1122, 1),
 (1745, 1),
 (1801, 1),
 (1802, 1),
 (1803, 1),
 (1804, 1)],
[(98, 1), (179, 1), (197, 1), (514, 1), (954, 1), (1210, 1)],
[(13, 1),
 (22, 2),
 (26, 1),
 (128, 1),
 (179, 1),
 (227, 1),
 (387, 1),
 (405, 1),
 (413, 1),
 (629, 1),
 (1805, 1),
 (1806, 1)],
[(19, 1),
 (22, 1),
 (23, 1),
 (24, 1),
 (33, 1),
 (65, 1),
 (77, 1),
 (92, 1),
 (112, 1),
 (128, 1),
 (243, 1),
 (270, 1),
 (295, 1),
 (316, 1),
 (342, 1),
 (375, 1),
 (399, 1),
 (1807, 1),
 (1808, 1)],
[(43, 1), (79, 1), (268, 1), (317, 1), (462, 1), (777, 1)],
[(6, 1),
 (7, 1),
 (15, 2),
 (19, 1),
 (22, 2),
 (26, 1),
 (33, 1),
 (61, 1),
 (80, 1),
 (94, 1),
 (104, 1),
 (128, 1),
 (181, 1),
 (300, 1),
 (306, 1),
 (1809, 1),
 (1810, 1)],
[(16, 1), (18, 1), (80, 1), (133, 1), (165, 1), (240, 1)],
[(0, 1),
 (19, 1),
 (22, 1),
 (26, 1),
 (33, 1),
 (49, 1),
 (53, 1),
 (74, 1),
 (83, 1),
 (94, 1),
 (104, 1),
 (313, 1),
 (472, 1),
 (636, 1),
 (638, 1),
 (747, 1),
 (1811, 1)],
[(16, 1), (48, 1), (93, 1), (207, 1), (419, 1)],
[(36, 1),
 (37, 3),
 (39, 2),
 (45, 1),
 (47, 3),
 (52, 1),
 (56, 1),
 (59, 2),
 (62, 1),
 (69, 4),
 (79, 1),
 (83, 3),
 (89, 1),
 (94, 5),
 (103, 2),
 (105, 2),
 (106, 3),
 (128, 2),
 (150, 1),
 (153, 1),
 (164, 9),
 (177, 1),
 (192, 2),
 (238, 1),
 (247, 3),
 (276, 4),
 (357, 1),
 (387, 1),
 (419, 1),
 (433, 1),
 (496, 2),
 (499, 1),
 (547, 1),
 (569, 1),
 (587, 1),
 (634, 1),
 (639, 1),
 (642, 1)
```

```
 (643, 1),
 (651, 1),
 (691, 2),
 (693, 1),
 (703, 2),
 (724, 1),
 (759, 1),
 (765, 1),
 (788, 3),
 (804, 1),
 (845, 2),
 (847, 4),
 (900, 1),
 (965, 3),
 (1023, 1),
 (1240, 1),
 (1308, 2),
 (1349, 1),
 (1421, 1),
 (1480, 1),
 (1537, 1),
 (1624, 1),
 (1812, 1),
 (1813, 1),
 (1814, 1),
 (1815, 1),
 (1816, 1),
 (1817, 1),
 (1818, 1),
 (1819, 1),
 (1820, 1),
 (1821, 1)],
[(5, 1), (120, 1), (121, 1), (142, 1), (197, 1), (611, 1), (664, 1)],
[(19, 1), (33, 1), (37, 1), (49, 1), (182, 1), (306, 1), (472, 1)],
[(16, 1),
 (26, 2),
 (33, 1),
 (164, 2),
 (179, 1),
 (200, 1),
 (345, 1),
 (706, 1),
 (924, 1),
 (1700, 1),
 (1822, 1)],
[(16, 1),
 (35, 1),
 (65, 1),
 (71, 1),
 (81, 1),
 (149, 2),
 (164, 1),
 (165, 1),
 (179, 2),
 (249, 1),
 (292, 1),
 (312, 1),
 (322, 1),
 (487, 1),
 (549, 1),
 (759, 1),
 (1823, 1),
 (1824, 1),
 (1825, 1)],
[(6, 1),
 (26, 1),
 (48, 1),
 (238, 2),
 (280, 1),
 (288, 1),
 (383, 1),
 (410, 1),
 (510, 1),
 (535, 1),
 (559, 1),
 (591, 1),
 (787, 1),
 (892, 1),
 (1826, 1)],
[(13, 1), (16, 1), (24, 1), (92, 1), (193, 1), (320, 1)],
[(16, 1), (24, 1), (79, 1), (133, 1), (179, 1), (197, 1), (296, 1)],
[(6, 1),
 (19, 1),
 (92, 1),
 (93, 1),
 (94, 1),
 (98, 1),
 (141, 1),
 (146, 1),
 (168, 1),
 (369, 1),
 (492, 1),
 (954, 1),
 (1827, 1),
 (1828, 1)],
[(22, 1),
 (35, 2),
 (48, 1),
 (62, 1),
 (83, 1),
 (128, 1),
 (133, 1),
 (179, 2),
 (293, 1),
 (421, 1),
 (788, 1),
 (979, 1),
 (1761, 1)],
[(7, 1),
 (15, 1),
 (33, 1),
 (92, 1),
 (213, 1),
 (230, 1),
 (510, 1),
 (975, 1),
```

```
 (975, 1),
 (1052, 1),
 (1072, 1),
 (1115, 1),
 (1829, 1)],
[(69, 1), (90, 1), (121, 1), (168, 1), (288, 1), (343, 1)],
[(53, 1), (93, 1), (136, 1), (193, 1), (207, 1), (315, 1)],
[(41, 1), (188, 1), (192, 1), (201, 1)],
[(1, 1),
 (61, 1),
 (70, 2),
 (97, 1),
 (101, 1),
 (139, 1),
 (153, 1),
 (165, 1),
 (168, 1),
 (176, 1),
 (177, 1),
 (192, 1),
 (433, 1),
 (514, 1),
 (800, 1),
 (1039, 1),
 (1188, 1),
 (1830, 1)],
[(20, 1), (94, 1), (133, 1), (179, 1), (1670, 1), (1831, 1)],
[(16, 2), (93, 1), (94, 1), (287, 1)],
[(33, 2),
 (61, 1),
 (95, 1),
 (97, 1),
 (177, 1),
 (207, 1),
 (230, 1),
 (315, 1),
 (480, 1),
 (497, 1),
 (798, 1),
 (954, 2),
 (1048, 1),
 (1176, 1),
 (1605, 1),
 (1832, 1)],
[(4, 1),
 (107, 1),
 (179, 1),
 (193, 1),
 (268, 1),
 (343, 1),
 (480, 1),
 (517, 1),
 (1221, 1),
 (1778, 1)],
[(53, 1),
 (83, 1),
 (100, 1),
 (272, 1),
 (587, 1),
 (777, 1),
 (937, 1),
 (1833, 1)],
[(13, 1),
 (20, 1),
 (33, 1),
 (52, 1),
 (62, 1),
 (64, 1),
 (79, 1),
 (164, 2),
 (172, 1),
 (790, 1)],
[(19, 1), (91, 1), (121, 1), (164, 1), (179, 1), (1399, 1)],
[(16, 2), (110, 1), (133, 1), (892, 1)],
[(33, 1), (603, 1), (661, 1), (1230, 1), (1834, 1), (1835, 1)],
[(16, 1),
 (51, 1),
 (65, 1),
 (81, 1),
 (252, 1),
 (264, 1),
 (322, 1),
 (385, 1),
 (390, 1),
 (750, 1),
 (949, 1),
 (1039, 1),
 (1269, 1),
 (1836, 1)],
[(87, 1),
 (139, 1),
 (153, 1),
 (203, 1),
 (434, 1),
 (514, 1),
 (957, 1),
 (985, 1),
 (1765, 1),
 (1837, 1)],
[(79, 1), (93, 2), (192, 1), (247, 1), (659, 1), (777, 1), (804, 1)],
[(16, 1), (29, 1), (36, 1), (179, 1), (268, 1), (387, 1), (1211, 1)],
[(33, 1), (92, 2), (133, 1), (155, 1), (318, 1), (385, 1), (780, 1)],
[(0, 1),
 (19, 1),
 (238, 1),
 (250, 1),
 (263, 2),
 (312, 1),
 (531, 1),
 (667, 1),
 (702, 1),
 (1045, 1),
 (1046, 1),
 (1424, 1),
 (1838, 1)],
[(6, 1),
```

```
 (78, 1),
 (89, 1),
 (90, 1),
 (136, 1),
 (268, 1),
 (367, 1),
 (435, 1),
 (1148, 1)],
[(92, 1),
 (161, 1),
 (197, 1),
 (201, 1),
 (217, 1),
 (388, 1),
 (429, 1),
 (1839, 1)],
[(35, 1), (61, 1), (90, 1), (179, 1), (420, 1), (1778, 1)],
[(35, 2),
 (36, 1),
 (93, 1),
 (108, 1),
 (168, 1),
 (211, 1),
 (288, 1),
 (310, 1),
 (420, 1),
 (517, 1),
 (537, 1),
 (648, 1),
 (672, 1),
 (704, 1),
 (733, 1),
 (1166, 1),
 (1685, 1),
 (1840, 1),
 (1841, 1)],
[(3, 1),
 (7, 1),
 (15, 1),
 (33, 1),
 (49, 1),
 (65, 1),
 (79, 1),
 (83, 1),
 (98, 2),
 (99, 1),
 (105, 1),
 (139, 1),
 (150, 1),
 (164, 1),
 (165, 1),
 (200, 1),
 (207, 1),
 (231, 1),
 (252, 1),
 (268, 1),
 (331, 1),
 (343, 1),
 (611, 1),
 (1029, 1),
 (1064, 1),
 (1101, 1),
 (1115, 1),
 (1187, 1),
 (1317, 1),
 (1778, 1),
 (1780, 1),
 (1842, 1),
 (1843, 1),
 (1844, 1),
 (1845, 1),
 (1846, 1),
 (1847, 1)],
[(7, 1),
 (20, 1),
 (136, 1),
 (139, 1),
 (193, 1),
 (539, 1),
 (633, 1),
 (890, 1),
 (971, 1),
 (1123, 1),
 (1660, 1)],
[(16, 2), (24, 1), (33, 1), (300, 1), (406, 1)],
[(18, 1),
 (35, 1),
 (65, 1),
 (72, 1),
 (168, 1),
 (179, 1),
 (185, 1),
 (197, 1),
 (244, 1),
 (264, 1),
 (300, 1),
 (329, 1),
 (477, 1),
 (519, 1),
 (702, 1),
 (1418, 1),
 (1848, 1)],
[(16, 1), (79, 1), (83, 1), (197, 1), (207, 1), (234, 1), (419, 1)],
[(71, 1), (79, 1), (120, 1), (262, 1), (292, 1), (690, 1), (759, 1)],
[(16, 1),
 (18, 1),
 (200, 1),
 (207, 1),
 (315, 1),
 (342, 1),
 (410, 1),
 (1192, 1),
 (1822, 1)],
[(0, 1),
 (20, 1),
```

```
 (33, 1),
 (51, 1),
 (79, 1),
 (80, 1),
 (100, 1),
 (179, 1),
 (192, 1),
 (197, 1),
 (201, 1),
 (306, 1),
 (331, 1),
 (1849, 1)],
[(16, 1), (33, 1), (37, 1), (855, 1), (1850, 1)],
[(21, 1), (53, 1), (138, 1), (139, 1), (153, 1), (164, 1)],
[(20, 2),
 (41, 1),
 (53, 1),
 (136, 1),
 (193, 1),
 (628, 1),
 (669, 1),
 (674, 1),
 (804, 1),
 (1211, 1),
 (1851, 1)],
[(33, 1), (52, 1), (97, 1), (121, 1), (179, 1), (249, 1), (759, 1)],
[(0, 1),
 (7, 1),
 (99, 1),
 (104, 1),
 (179, 1),
 (197, 1),
 (349, 1),
 (1210, 1),
 (1852, 1)],
[(79, 1), (121, 1), (125, 1), (197, 1), (458, 1), (675, 1), (1450, 1)],
[(16, 1), (20, 1), (24, 1), (33, 1), (136, 1), (316, 1)],
[(56, 1),
 (65, 1),
 (73, 1),
 (79, 1),
 (83, 1),
 (94, 1),
 (125, 1),
 (150, 1),
 (164, 1),
 (184, 1),
 (197, 1),
 (201, 1),
 (217, 1),
 (224, 1),
 (262, 1),
 (263, 1),
 (286, 1),
 (410, 1),
 (470, 1),
 (702, 1),
 (840, 1),
 (1853, 1),
 (1854, 1),
 (1855, 1)],
[(16, 1), (22, 1), (79, 1), (92, 1), (128, 1), (197, 1), (211, 1), (243, 1)],
[(105, 2), (197, 1), (343, 1), (433, 1), (1856, 1)],
[(22, 1),
 (35, 1),
 (65, 2),
 (80, 1),
 (161, 1),
 (168, 1),
 (179, 1),
 (230, 1),
 (254, 1),
 (255, 2),
 (258, 1),
 (286, 1),
 (358, 1),
 (367, 2),
 (429, 1),
 (517, 1),
 (535, 1),
 (627, 2),
 (675, 1),
 (759, 1),
 (777, 1),
 (791, 1),
 (945, 2),
 (982, 1),
 (1183, 1),
 (1204, 1),
 (1266, 1),
 (1394, 1),
 (1857, 1)],
[(33, 1),
 (45, 1),
 (64, 1),
 (78, 1),
 (83, 1),
 (87, 1),
 (89, 1),
 (93, 1),
 (150, 1),
 (193, 1),
 (308, 1),
 (373, 1),
 (553, 1),
 (833, 1)],
[(33, 1), (34, 1), (62, 1), (485, 1), (559, 1)],
[(19, 1),
 (35, 1),
 (47, 2),
 (65, 1),
 (94, 1),
 (155, 1),
 (168, 2),
 (211, 2),
```

```
 (232, 1),
 (264, 1),
 (367, 1),
 (477, 1),
 (571, 1),
 (572, 2),
 (765, 1),
 (791, 1),
 (806, 1),
 (1096, 1),
 (1509, 1),
 (1762, 1),
 (1858, 1),
 (1859, 2),
 (1860, 1),
 (1861, 1)],
[(22, 1), (48, 1), (61, 1), (261, 1), (300, 1), (1862, 1)],
[(33, 1), (79, 1), (92, 1), (103, 1), (1863, 1)],
[(36, 1),
 (61, 1),
 (62, 1),
 (64, 1),
 (90, 1),
 (94, 1),
 (121, 1),
 (136, 1),
 (160, 1),
 (192, 1),
 (262, 2),
 (419, 1),
 (1551, 1)],
[(49, 1),
 (92, 1),
 (93, 1),
 (127, 2),
 (193, 1),
 (322, 1),
 (406, 1),
 (513, 1),
 (626, 1),
 (820, 1),
 (1086, 1),
 (1234, 1),
 (1293, 1),
 (1389, 2),
 (1864, 1),
 (1865, 1)],
[(16, 1), (33, 1), (104, 1), (211, 1), (625, 1)],
[(79, 1),
 (80, 1),
 (83, 1),
 (94, 1),
 (161, 1),
 (439, 1),
 (592, 1),
 (1049, 1),
 (1866, 1),
 (1867, 1)],
[(33, 2), (83, 1), (195, 1), (207, 1), (954, 1), (1868, 1), (1869, 1)],
[(6, 1),
 (16, 1),
 (22, 1),
 (33, 1),
 (128, 1),
 (168, 1),
 (169, 1),
 (211, 1),
 (252, 1),
 (304, 1),
 (369, 1),
 (617, 1)],
[(16, 1), (33, 1), (83, 1), (160, 1), (230, 1), (231, 1), (1870, 1)],
[(35, 2), (92, 1), (105, 1), (179, 1), (216, 1), (595, 2), (773, 1)],
[(4, 1),
 (47, 1),
 (52, 1),
 (69, 1),
 (83, 1),
 (87, 1),
 (88, 1),
 (90, 1),
 (92, 1),
 (109, 1),
 (217, 1),
 (231, 1),
 (406, 1),
 (667, 1),
 (743, 1),
 (1102, 1),
 (1865, 1)],
[(26, 1), (36, 1), (53, 2), (79, 1), (115, 1), (160, 1), (247, 1), (1871, 1)],
[(16, 1),
 (33, 1),
 (79, 1),
 (87, 1),
 (105, 1),
 (268, 1),
 (869, 1),
 (1193, 1),
 (1470, 1)],
[(16, 1),
 (45, 1),
 (47, 1),
 (65, 1),
 (79, 1),
 (81, 1),
 (225, 1),
 (248, 1),
 (264, 1),
 (325, 1),
 (411, 1),
 (420, 1),
 (562, 1),
 (1481, 1)],
[(41, 1), (62, 1), (94, 1), (120, 1), (142, 1), (227, 1)],
[(13, 1), (24, 1), (93, 2), (100, 1), (220, 1)],
```

```
[(13, 1), (24, 1), (92, 2), (190, 1), (320, 1)],
[(6, 1),
 (7, 1),
 (20, 1),
 (33, 1),
 (37, 1),
 (92, 1),
 (94, 1),
 (133, 3),
 (188, 1),
 (243, 1),
 (252, 1),
 (255, 1),
 (287, 1),
 (343, 2),
 (369, 1),
 (740, 1),
 (962, 1),
 (1192, 1),
 (1872, 1),
 (1873, 1)],
[(0, 1),
 (16, 1),
 (26, 1),
 (37, 1),
 (93, 1),
 (94, 1),
 (207, 1),
 (315, 1),
 (433, 1),
 (607, 1),
 (900, 1),
 (914, 1),
 (1537, 1),
 (1874, 1)],
[(160, 1),
 (249, 1),
 (329, 1),
 (429, 1),
 (496, 1),
 (812, 1),
 (1457, 1),
 (1875, 1)],
[(16, 2),
 (34, 2),
 (52, 1),
 (114, 2),
 (150, 1),
 (159, 1),
 (339, 1),
 (343, 2),
 (363, 1),
 (571, 1),
 (660, 1),
 (710, 1),
 (1080, 1)],
[(49, 1),
 (98, 1),
 (194, 1),
 (238, 1),
 (255, 2),
 (258, 1),
 (388, 1),
 (434, 1),
 (458, 1),
 (491, 1),
 (721, 1),
 (1876, 1)],
[(20, 1),
 (26, 1),
 (79, 2),
 (87, 1),
 (106, 1),
 (114, 1),
 (139, 2),
 (153, 1),
 (181, 1),
 (197, 1),
 (201, 1),
 (306, 1),
 (343, 1),
 (458, 1),
 (651, 1),
 (790, 1),
 (1212, 1),
 (1399, 1),
 (1877, 1)],
[(5, 1),
 (19, 1),
 (36, 1),
 (46, 2),
 (47, 3),
 (62, 1),
 (77, 1),
 (78, 2),
 (80, 1),
 (90, 1),
 (92, 1),
 (103, 1),
 (161, 1),
 (179, 1),
 (217, 1),
 (230, 1),
 (239, 1),
 (262, 1),
 (263, 1),
 (279, 1),
 (287, 1),
 (349, 1),
 (363, 1),
 (367, 1),
 (374, 2),
 (399, 1),
 (432, 1),
 (536, 1),
 (549, 1),
```

```
 (667, 2),
 (841, 1),
 (848, 1),
 (1133, 1),
 (1411, 1),
 (1878, 2)],
[(16, 1), (51, 1), (61, 1), (70, 1), (93, 1), (423, 1)],
[(23, 1),
 (64, 1),
 (79, 1),
 (94, 1),
 (207, 1),
 (238, 1),
 (258, 1),
 (356, 1),
 (489, 1),
 (774, 1),
 (1033, 1),
 (1123, 1),
 (1234, 1),
 (1650, 1)],
[(35, 2),
 (71, 1),
 (179, 1),
 (217, 1),
 (263, 1),
 (329, 1),
 (443, 1),
 (773, 1),
 (1025, 1)],
[(11, 1),
 (13, 1),
 (15, 1),
 (19, 1),
 (22, 1),
 (26, 2),
 (33, 1),
 (36, 1),
 (49, 1),
 (52, 1),
 (92, 1),
 (94, 1),
 (104, 1),
 (185, 1),
 (194, 1),
 (226, 1),
 (446, 1),
 (536, 1),
 (539, 1),
 (1879, 1),
 (1880, 1),
 (1881, 1),
 (1882, 1)],
[(16, 1), (83, 1), (207, 1), (217, 2), (906, 2), (1883, 1), (1884, 1)],
[(16, 2),
 (24, 1),
 (83, 1),
 (92, 1),
 (93, 1),
 (114, 1),
 (205, 1),
 (207, 1),
 (343, 1),
 (360, 1),
 (365, 1),
 (410, 1),
 (534, 1),
 (962, 1),
 (1442, 1)],
[(20, 1),
 (33, 1),
 (36, 1),
 (57, 1),
 (62, 1),
 (79, 3),
 (94, 1),
 (155, 1),
 (227, 1),
 (268, 1),
 (380, 1),
 (477, 1),
 (719, 1)],
[(114, 1), (128, 1), (179, 1), (636, 1)],
[(35, 1), (79, 1), (197, 1), (420, 1), (525, 1), (785, 1)],
[(0, 1), (92, 1), (343, 1), (426, 1), (932, 1), (994, 1)],
[(33, 1), (83, 1), (92, 1), (133, 1), (205, 1), (207, 1)],
[(61, 1),
 (128, 1),
 (151, 1),
 (179, 2),
 (385, 1),
 (496, 1),
 (514, 1),
 (1885, 1)],
[(1, 1),
 (13, 1),
 (49, 1),
 (51, 1),
 (79, 1),
 (92, 1),
 (127, 1),
 (179, 1),
 (193, 3),
 (197, 1),
 (304, 1),
 (322, 1),
 (401, 1),
 (435, 1),
 (720, 1),
 (731, 1),
 (985, 1)],
[(6, 1), (34, 1), (37, 1), (47, 1), (52, 1), (1886, 1)],
[(16, 1), (93, 1), (120, 1), (222, 1), (262, 1), (411, 1), (1540, 1)],
[(16, 1), (20, 1), (62, 2), (105, 1), (177, 1), (211, 1), (492, 1), (627, 1)],
[(34, 1),
```

```
 (150, 1),
 (300, 1),
 (343, 1),
 (457, 1),
 (603, 1),
 (869, 1),
 (1329, 1),
 (1790, 1)],
[(20, 1),
 (37, 1),
 (87, 1),
 (98, 1),
 (127, 1),
 (164, 1),
 (193, 2),
 (258, 1),
 (316, 1),
 (324, 1),
 (385, 1),
 (388, 1),
 (487, 1),
 (517, 1),
 (547, 1),
 (571, 1),
 (802, 1),
 (846, 1),
 (1022, 1),
 (1390, 1),
 (1667, 1),
 (1887, 1)],
[(20, 1), (33, 1), (164, 1), (458, 1), (719, 1), (1888, 1)],
[(3, 1),
 (20, 1),
 (87, 2),
 (117, 1),
 (140, 1),
 (399, 1),
 (492, 1),
 (717, 1),
 (736, 1),
 (839, 1),
 (1177, 1),
 (1263, 1),
 (1326, 1)],
[(47, 1),
 (62, 1),
 (69, 1),
 (79, 1),
 (105, 1),
 (119, 1),
 (159, 1),
 (161, 1),
 (217, 1),
 (222, 1),
 (231, 1),
 (288, 1),
 (304, 1),
 (314, 1),
 (316, 2),
 (399, 1),
 (411, 1),
 (472, 1),
 (1248, 1),
 (1309, 1),
 (1715, 1),
 (1889, 1),
 (1890, 1)],
[(20, 1), (35, 1), (37, 1), (87, 1), (406, 1), (527, 1)],
[(20, 3),
 (26, 1),
 (33, 1),
 (94, 1),
 (95, 1),
 (96, 1),
 (97, 1),
 (98, 1),
 (99, 1),
 (100, 1),
 (101, 1),
 (102, 1),
 (103, 1)],
[(20, 3),
 (26, 1),
 (64, 1),
 (79, 1),
 (87, 1),
 (103, 1),
 (194, 1),
 (197, 1),
 (434, 1),
 (496, 1),
 (510, 1),
 (790, 1),
 (825, 2),
 (850, 1),
 (1399, 1),
 (1891, 1),
 (1892, 1)],
[(5, 1),
 (6, 1),
 (20, 1),
 (36, 1),
 (47, 1),
 (93, 1),
 (95, 1),
 (150, 1),
 (161, 1),
 (179, 1),
 (197, 2),
 (201, 1),
 (217, 1),
 (231, 1),
 (254, 2),
 (276, 1),
 (286, 1),
```

```
 (308, 1),
 (316, 1),
 (374, 1),
 (442, 1),
 (793, 1),
 (1390, 1),
 (1403, 1),
 (1893, 1),
 (1894, 1)],
[(1, 1),
 (6, 1),
 (7, 1),
 (26, 1),
 (143, 1),
 (164, 2),
 (165, 2),
 (176, 1),
 (316, 1),
 (349, 1),
 (478, 1),
 (788, 1),
 (916, 1),
 (917, 1),
 (1159, 1),
 (1509, 1),
 (1895, 1),
 (1896, 1)],
[(13, 1),
 (26, 1),
 (64, 2),
 (89, 1),
 (92, 1),
 (136, 1),
 (139, 1),
 (143, 1),
 (153, 1),
 (162, 1),
 (179, 1),
 (193, 1),
 (199, 1),
 (211, 1),
 (342, 1),
 (343, 1),
 (368, 1),
 (373, 1),
 (397, 1),
 (669, 1)],
[(26, 1),
 (47, 1),
 (55, 1),
 (56, 1),
 (68, 1),
 (69, 1),
 (120, 1),
 (161, 1),
 (197, 1),
 (213, 1),
 (281, 1),
 (477, 1),
 (691, 1),
 (768, 1),
 (1096, 1),
 (1882, 1),
 (1897, 1)],
[(5, 1),
 (65, 1),
 (92, 1),
 (94, 1),
 (121, 1),
 (192, 1),
 (197, 1),
 (201, 1),
 (217, 1),
 (314, 1),
 (350, 1),
 (352, 1),
 (387, 1),
 (406, 1),
 (607, 1),
 (730, 1),
 (750, 1),
 (1293, 1),
 (1898, 1)],
[(16, 2), (211, 1), (316, 1), (426, 1), (775, 1), (1899, 1)],
[(16, 1),
 (47, 1),
 (56, 1),
 (59, 1),
 (69, 1),
 (80, 1),
 (136, 1),
 (161, 1),
 (217, 1),
 (245, 1),
 (246, 1),
 (691, 1)],
[(20, 1), (92, 1), (94, 1), (247, 1), (287, 1), (1900, 1)],
[(20, 1),
 (22, 1),
 (36, 1),
 (79, 1),
 (103, 1),
 (110, 1),
 (126, 1),
 (128, 1),
 (133, 1),
 (160, 1),
 (164, 1),
 (180, 1),
 (205, 1),
 (211, 1),
 (329, 1),
 (354, 1),
 (357, 1),
 (389, 1),
```

```
 (458, 1),
 (479, 1),
 (587, 1),
 (1576, 1),
 (1825, 1),
 (1901, 1),
 (1902, 1),
 (1903, 1)],
[(79, 2),
 (87, 1),
 (107, 1),
 (237, 1),
 (386, 1),
 (425, 1),
 (591, 1),
 (630, 1),
 (664, 1),
 (762, 1),
 (1904, 1),
 (1905, 1)],
[(4, 1),
 (7, 1),
 (19, 1),
 (20, 1),
 (22, 1),
 (35, 1),
 (36, 1),
 (46, 2),
 (47, 2),
 (56, 1),
 (65, 1),
 (81, 1),
 (93, 1),
 (94, 1),
 (103, 1),
 (105, 1),
 (117, 1),
 (159, 1),
 (179, 1),
 (222, 1),
 (231, 1),
 (259, 1),
 (262, 1),
 (296, 1),
 (325, 1),
 (367, 2),
 (410, 1),
 (452, 1),
 (484, 1),
 (535, 1),
 (562, 1),
 (624, 1),
 (693, 1),
 (750, 1),
 (808, 1),
 (820, 1),
 (1047, 1),
 (1103, 1),
 (1422, 1),
 (1745, 1),
 (1906, 1),
 (1907, 1),
 (1908, 1),
 (1909, 1),
 (1910, 1)],
[(114, 1), (168, 1), (211, 1), (374, 1), (517, 1), (774, 1), (1039, 1)],
[(16, 1),
 (94, 2),
 (122, 1),
 (160, 1),
 (168, 1),
 (179, 1),
 (193, 1),
 (211, 3),
 (321, 1),
 (324, 1),
 (774, 1),
 (1911, 1)],
[(16, 1),
 (34, 1),
 (55, 1),
 (79, 1),
 (185, 1),
 (197, 2),
 (201, 1),
 (213, 1),
 (263, 2),
 (702, 1),
 (1171, 1)],
[(4, 1),
 (16, 1),
 (35, 1),
 (47, 1),
 (69, 1),
 (179, 1),
 (197, 1),
 (201, 1),
 (211, 1),
 (262, 1),
 (306, 1),
 (664, 1),
 (1912, 1)],
[(0, 1),
 (4, 1),
 (6, 1),
 (7, 1),
 (8, 1),
 (11, 1),
 (24, 1),
 (77, 1),
 (92, 1),
 (94, 1),
 (103, 1),
 (112, 1),
 (139, 1),
 (155, 2),
```

```
  (155, 2),
  (164, 1),
  (165, 2),
  (169, 1),
  (178, 1),
  (185, 2),
  (244, 1),
  (252, 1),
  (254, 1),
  (262, 1),
  (287, 1),
  (381, 1),
  (389, 1),
  (395, 1),
  (409, 1),
  (510, 1),
  (521, 1),
  (670, 1),
  (1052, 1),
  (1159, 1),
  (1221, 1),
  (1510, 1),
  (1913, 1)],
 ...]
```

In [ ]:

```
from gensim.models import LdaModel
```

In [ ]:

```
NUM_TOPICS = 9
ldamodel = LdaModel(doc_term_matrix, num_topics=NUM_TOPICS, id2word=dictionary, passes=30)
```

In [ ]:

```
topics = ldamodel.show_topics()
for topic in topics:
    print(topic)
    print()
```

```
(0, '0.067*"love" + 0.044*"buy" + 0.042*"tablet" + 0.031*"old" + 0.026*"year" + 0.024*"gift" + 0.022*"one" + 0.021*"kid" + 0.019*"game" + 0.019*"great"'

(1, '0.023*"buy" + 0.022*"purchase" + 0.022*"one" + 0.019*"get" + 0.018*"best" + 0.015*"charge" + 0.014*"kindle" + 0.013*"time" + 0.013*"plus" + 0.012*"

(2, '0.048*"alexa" + 0.031*"love" + 0.026*"ask" + 0.023*"use" + 0.019*"question" + 0.015*"music" + 0.013*"screen" + 0.013*"family" + 0.011*"weather" + 0
')

(3, '0.079*"tablet" + 0.039*"good" + 0.039*"great" + 0.036*"price" + 0.029*"fire" + 0.024*"amazon" + 0.020*"work" + 0.017*"screen" + 0.016*"hd" + 0.014*

(4, '0.059*"echo" + 0.036*"show" + 0.029*"great" + 0.021*"video" + 0.021*"home" + 0.019*"music" + 0.019*"alexa" + 0.019*"amazon" + 0.017*"sound" + 0.015

(5, '0.031*"apps" + 0.026*"tablet" + 0.022*"amazon" + 0.020*"kid" + 0.015*"great" + 0.014*"download" + 0.014*"book" + 0.012*"store" + 0.011*"get" + 0.01

(6, '0.022*"work" + 0.020*"light" + 0.016*"like" + 0.016*"good" + 0.015*"well" + 0.014*"tap" + 0.014*"device" + 0.013*"music" + 0.012*"sound" + 0.012*"g

(7, '0.126*"easy" + 0.092*"use" + 0.054*"great" + 0.044*"set" + 0.042*"product" + 0.027*"recommend" + 0.025*"love" + 0.018*"fun" + 0.015*"buy" + 0.015*"

(8, '0.051*"read" + 0.044*"kindle" + 0.030*"book" + 0.016*"use" + 0.015*"love" + 0.015*"screen" + 0.014*"good" + 0.014*"light" + 0.014*"much" + 0.012*"g
```

In [ ]:

```
word_dict = {}
for i in range(NUM_TOPICS):
    words = ldamodel.show_topic(i, topn = 20)
    word_dict["Topic # " + "{}".format(i)] = [i[0] for i in words]
```

In [ ]:

```
pd.DataFrame(word_dict)
```

Out[ ]:

	Topic # 0	Topic # 1	Topic # 2	Topic # 3	Topic # 4	Topic # 5	Topic # 6	Topic # 7	Topic # 8
0	love	buy	alexa	tablet	echo	apps	work	easy	read
1	buy	purchase	love	good	show	tablet	light	use	kindle
2	tablet	one	ask	great	great	amazon	like	great	book
3	old	get	use	price	video	kid	good	set	use
4	year	best	question	fire	home	great	well	product	love
5	gift	charge	music	amazon	music	download	tap	recommend	screen
6	one	kindle	screen	work	alexa	book	device	love	good
7	kid	time	family	screen	amazon	store	music	fun	light
8	game	plus	weather	hd	sound	get	sound	buy	much
9	great	bulb	answer	quality	love	free	great	work	great
10	purchase	go	thing	size	device	use	turn	would	reader
11	get	good	device	fast	smart	device	use	lot	like
12	son	model	talk	buy	good	available	speaker	learn	fire
13	play	echo	turn	nice	dot	play	get	enjoy	one
14	daughter	first	enjoy	one	product	prime	room	kid	would
15	christmas	hue	fun	picture	use	buy	product	day	size
16	use	plug	play	would	like	also	amazon	every	want
17	kindle	problem	get	enough	play	google	button	highly	new
18	child	back	day	need	work	time	portable	really	easy
19	good	generation	know	best	also	ipad	far	thing	battery

In [ ]:

```
ldamodel.show_topic(0, topn = 20)
```

Out[ ]:

```
[('love', 0.06724249),
```

```
 ('buy', 0.04424884),
 ('tablet', 0.04203031),
 ('old', 0.030797651),
 ('year', 0.0258037),
 ('gift', 0.02364784),
 ('one', 0.021550724),
 ('kid', 0.021057405),
 ('game', 0.019351698),
 ('great', 0.01871655),
 ('purchase', 0.017218987),
 ('get', 0.017063275),
 ('son', 0.0156080285),
 ('play', 0.015598591),
 ('daughter', 0.015259634),
 ('christmas', 0.013514392),
 ('use', 0.012027702),
 ('kindle', 0.011110389),
 ('child', 0.01043539),
 ('good', 0.009310327)]
```

In [ ]:

```
!pip install pyLDAvis
```

```
Collecting pyLDAvis
  Downloading https://files.pythonhosted.org/packages/a5/3a/af82e070a8a96e13217c8f362f9a73e82d61ac8fff3a2561946a97f96266/pyLDAvis-2.1.2.tar.gz (1.6MB)
     |████████████████████████████████| 1.6MB 2.8MB/s
Requirement already satisfied: wheel>=0.23.0 in /usr/local/lib/python3.6/dist-packages (from pyLDAvis) (0.34.2)
Requirement already satisfied: numpy>=1.9.2 in /usr/local/lib/python3.6/dist-packages (from pyLDAvis) (1.18.5)
Requirement already satisfied: scipy>=0.18.0 in /usr/local/lib/python3.6/dist-packages (from pyLDAvis) (1.4.1)
Requirement already satisfied: pandas>=0.17.0 in /usr/local/lib/python3.6/dist-packages (from pyLDAvis) (1.0.5)
Requirement already satisfied: joblib>=0.8.4 in /usr/local/lib/python3.6/dist-packages (from pyLDAvis) (0.15.1)
Requirement already satisfied: jinja2>=2.7.2 in /usr/local/lib/python3.6/dist-packages (from pyLDAvis) (2.11.2)
Requirement already satisfied: numexpr in /usr/local/lib/python3.6/dist-packages (from pyLDAvis) (2.7.1)
Requirement already satisfied: pytest in /usr/local/lib/python3.6/dist-packages (from pyLDAvis) (3.6.4)
Requirement already satisfied: future in /usr/local/lib/python3.6/dist-packages (from pyLDAvis) (0.16.0)
Collecting funcy
  Downloading https://files.pythonhosted.org/packages/ce/4b/6ffa76544e46614123de31574ad95758c421aae391a1764921b8a81e1eae/funcy-1.14.tar.gz (548kB)
     |████████████████████████████████| 552kB 17.7MB/s
Requirement already satisfied: python-dateutil>=2.6.1 in /usr/local/lib/python3.6/dist-packages (from pandas>=0.17.0->pyLDAvis) (2.8.1)
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.6/dist-packages (from pandas>=0.17.0->pyLDAvis) (2018.9)
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.6/dist-packages (from jinja2>=2.7.2->pyLDAvis) (1.1.1)
Requirement already satisfied: six>=1.10.0 in /usr/local/lib/python3.6/dist-packages (from pytest->pyLDAvis) (1.12.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.6/dist-packages (from pytest->pyLDAvis) (47.3.1)
Requirement already satisfied: more-itertools>=4.0.0 in /usr/local/lib/python3.6/dist-packages (from pytest->pyLDAvis) (8.4.0)
Requirement already satisfied: py>=1.5.0 in /usr/local/lib/python3.6/dist-packages (from pytest->pyLDAvis) (1.8.2)
Requirement already satisfied: atomicwrites>=1.0 in /usr/local/lib/python3.6/dist-packages (from pytest->pyLDAvis) (1.4.0)
Requirement already satisfied: attrs>=17.4.0 in /usr/local/lib/python3.6/dist-packages (from pytest->pyLDAvis) (19.3.0)
Requirement already satisfied: pluggy<0.8,>=0.5 in /usr/local/lib/python3.6/dist-packages (from pytest->pyLDAvis) (0.7.1)
Building wheels for collected packages: pyLDAvis, funcy
  Building wheel for pyLDAvis (setup.py) ... done
  Created wheel for pyLDAvis: filename=pyLDAvis-2.1.2-py2.py3-none-any.whl size=97711 sha256=f0741ed35ff29f7f66b965c0643ff54d089df7790dec7c87f6220ac8b81
  Stored in directory: /root/.cache/pip/wheels/98/71/24/513a99e58bb6b8465bae4d2d5e9dba8f0bef8179e3051ac414
  Building wheel for funcy (setup.py) ... done
  Created wheel for funcy: filename=funcy-1.14-py2.py3-none-any.whl size=32042 sha256=38f06b1034552c0dccf72665d438ba1d57f331ea8c9ca794e274c354207eef3b
  Stored in directory: /root/.cache/pip/wheels/20/5a/d8/1d875df03deae6f178dfdf70238cca33f948ef8a6f5209f2eb
Successfully built pyLDAvis funcy
Installing collected packages: funcy, pyLDAvis
Successfully installed funcy-1.14 pyLDAvis-2.1.2
```

In [ ]:

```
import pyLDAvis.gensim
```

Displaying Results & Getting Insights

In [ ]:

```
Lda_display = pyLDAvis.gensim.prepare(ldamodel, doc_term_matrix, dictionary, sort_topics=False)
pyLDAvis.display(Lda_display)
```

Out[ ]:

Creating a Wordcloud

In [ ]:

```
from wordcloud import WordCloud, STOPWORDS
txt = data2["Processed_Review"].values
wc = WordCloud(width=200, height=100, background_color="white", stopwords=STOPWORDS).generate(str(txt))
fig = plt.figure(figsize=(20,20), facecolor='k', edgecolor='w')
plt.imshow(wc, interpolation="bilinear")
plt.axis("off")
plt.tight_layout()
plt.show()
```


alexa
thing
order
black
price
two
way
great
speak
fire
powerful
start
best
want