

Personalized cancer diagnosis

1. Business Problem

1.1. Description

Source: <https://www.kaggle.com/c/msk-redefining-cancer-treatment/>

Data: Memorial Sloan Kettering Cancer Center (MSKCC)

Download training_variants.zip and training_text.zip from Kaggle.

Context:

Source: <https://www.kaggle.com/c/msk-redefining-cancer-treatment/discussion/35336#198462>

Problem statement :

Classify the given genetic variations/mutations based on evidence from text-based clinical literature.

1.2. Source/Useful Links

Some articles and reference blogs about the problem statement

1. <https://www.forbes.com/sites/matthewherper/2017/06/03/a-new-cancer-drug-helped-almost-everyone-who-took-it-almost-heres-what-it-teaches-us/#2a44ee2f6b25>

2. <https://www.youtube.com/watch?v=UwbuW7oK8rk>
3. <https://www.youtube.com/watch?v=qxXRKVompl8>

1.3. Real-world/Business objectives and constraints.

- No low-latency requirement.
- Interpretability is important.
- Errors can be very costly.
- Probability of a data-point belonging to each class is needed.

2. Machine Learning Problem Formulation

2.1. Data

2.1.1. Data Overview

- Source: <https://www.kaggle.com/c/msk-redefining-cancer-treatment/data>
- We have two data files: one contains the information about the genetic mutations and the other contains the clinical evidence (text) that human experts/pathologists use to classify the genetic mutations.
- Both these data files have a common column called ID
- Data file's information:
 - training_variants (ID, Gene, Variations, Class)
 - training_text (ID, Text)

2.1.2. Example Data Point

training_variants

ID, Gene, Variation, Class

0, FAM58A, Truncating Mutations, 1

1, CBL, W802*, 2

2, CBL, Q249E, 2

...

training_text

ID, Text

0||Cyclin-dependent kinases (CDKs) regulate a variety of fundamental cellular processes. CDK10 stands out as one of the last orphan CDKs for which no activating cyclin has been identified and no kinase activity revealed. Previous work has shown that CDK10 silencing increases ETS2 (v-ets erythroblastosis virus E26 oncogene homolog 2)-driven activation of the MAPK pathway, which confers tamoxifen resistance to breast cancer cells. The precise mechanisms by which CDK10 modulates ETS2 activity, and more generally the functions of CDK10, remain elusive. Here we demonstrate that CDK10 is a cyclin-dependent kinase by identifying cyclin M as an activating cyclin. Cyclin M, an orphan cyclin, is the product of FAM58A, whose mutations cause STAR syndrome, a human developmental anomaly whose features include toe syndactyly, telecanthus, and anogenital and renal malformations. We show that STAR syndrome-associated cyclin M mutants are unable to interact with CDK10. Cyclin M silencing phenocopies CDK10 silencing in increasing c-Raf and in conferring tamoxifen resistance to breast cancer cells. CDK10/cyclin M phosphorylates ETS2 in vitro, and in cells it positively controls ETS2 degradation by the proteasome. ETS2 protein levels are increased in cells derived from a STAR patient, and this increase is attributable to decreased cyclin M levels. Altogether, our results reveal an additional regulatory mechanism for ETS2, which plays key roles in cancer and development. They also shed light on the molecular mechanisms underlying STAR syndrome. Cyclin-dependent kinases (CDKs) play a pivotal role in the control of a number of fundamental cellular processes (1). The human genome contains 21 genes encoding proteins

that can be considered as members of the CDK family owing to their sequence similarity with bona fide CDKs, those known to be activated by cyclins (2). Although discovered almost 20 y ago (3, 4), CDK10 remains one of the two CDKs without an identified cyclin partner. This knowledge gap has largely impeded the exploration of its biological functions. CDK10 can act as a positive cell cycle regulator in some cells (5, 6) or as a tumor suppressor in others (7, 8). CDK10 interacts with the ETS2 (v-ets erythroblastosis virus E26 oncogene homolog 2) transcription factor and inhibits its transcriptional activity through an unknown mechanism (9). CDK10 knockdown derepresses ETS2, which increases the expression of the c-Raf protein kinase, activates the MAPK pathway, and induces resistance of MCF7 cells to tamoxifen (6). ...

2.2. Mapping the real-world problem to an ML problem

2.2.1. Type of Machine Learning Problem

There are nine different classes a genetic mutation can be classified into => Multi class classification problem

2.2.2. Performance Metric

Source: <https://www.kaggle.com/c/msk-redefining-cancer-treatment#evaluation>

Metric(s):

- Multi class log-loss
- Confusion matrix

2.2.3. Machine Learning Objectives and Constraints

Objective: Predict the probability of each data-point belonging to each of the nine classes.

Constraints:

- Interpretability
- Class probabilities are needed.
- Penalize the errors in class probabilities => Metric is Log-loss.
- No Latency constraints.

2.3. Train, CV and Test Datasets

Split the dataset randomly into three parts train, cross validation and test with 64%, 16%, 20% of data respectively

3. Exploratory Data Analysis

```
In [1]: import pandas as pd
import matplotlib.pyplot as plt
import re
import time
import warnings
import numpy as np
from nltk.corpus import stopwords
from sklearn.decomposition import TruncatedSVD
from sklearn.preprocessing import normalize
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.manifold import TSNE
import seaborn as sns
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix
from sklearn.metrics.classification import accuracy_score, log_loss
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import SGDClassifier
```

```

from imblearn.over_sampling import SMOTE
from collections import Counter
from scipy.sparse import hstack
from sklearn.multiclass import OneVsRestClassifier
from sklearn.svm import SVC
from sklearn.model_selection import StratifiedKFold
from collections import Counter, defaultdict
from sklearn.calibration import CalibratedClassifierCV
from sklearn.naive_bayes import MultinomialNB
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
from sklearn.model_selection import GridSearchCV
import math
from sklearn.metrics import normalized_mutual_info_score
from sklearn.ensemble import RandomForestClassifier
warnings.filterwarnings("ignore")

from mlxtend.classifier import StackingClassifier

from sklearn import model_selection
from sklearn.linear_model import LogisticRegression

```

```

C:\Users\rdbz3b\AppData\Local\Continuum\anaconda3\lib\site-packages\skl
earn\ensemble\weight_boosting.py:29: DeprecationWarning: numpy.core.uma
th_tests is an internal NumPy module and should not be imported. It wil
l be removed in a future NumPy release.
  from numpy.core.umath_tests import inner1d

```

3.1. Reading Data

3.1.1. Reading Gene and Variation Data

```

In [2]: data = pd.read_csv('training/training_variants')
print('Number of data points : ', data.shape[0])
print('Number of features : ', data.shape[1])

```

```
print('Features : ', data.columns.values)
data.head()
```

Number of data points : 3321

Number of features : 4

Features : ['ID' 'Gene' 'Variation' 'Class']

Out[2]:

	ID	Gene	Variation	Class
0	0	FAM58A	Truncating Mutations	1
1	1	CBL	W802*	2
2	2	CBL	Q249E	2
3	3	CBL	N454D	3
4	4	CBL	L399V	4

training/training_variants is a comma separated file containing the description of the genetic mutations used for training.

Fields are

- **ID** : the id of the row used to link the mutation to the clinical evidence
- **Gene** : the gene where this genetic mutation is located
- **Variation** : the aminoacid change for this mutations
- **Class** : 1-9 the class this genetic mutation has been classified on

3.1.2. Reading Text Data

```
In [3]: # note the separator in this file
data_text = pd.read_csv("training/training_text", sep="\\|\\|", engine="python", names=["ID", "TEXT"], skiprows=1)
print('Number of data points : ', data_text.shape[0])
print('Number of features : ', data_text.shape[1])
```

```
print('Features : ', data_text.columns.values)
data_text.head()
```

Number of data points : 3321
Number of features : 2
Features : ['ID' 'TEXT']

Out[3]:

	ID	TEXT
0	0	Cyclin-dependent kinases (CDKs) regulate a var...
1	1	Abstract Background Non-small cell lung canc...
2	2	Abstract Background Non-small cell lung canc...
3	3	Recent evidence has demonstrated that acquired...
4	4	Oncogenic mutations in the monomeric Casitas B...

3.1.3. Preprocessing of text

```
In [4]: # loading stop words from nltk library

import nltk
nltk.download('stopwords')

stop_words = set(stopwords.words('english'))

def nlp_preprocessing(total_text, index, column):
    if type(total_text) is not int:
        string = ""
        # replace every special char with space
        total_text = re.sub('[^a-zA-Z0-9\n]', ' ', total_text)
        # replace multiple spaces with single space
        total_text = re.sub('\s+', ' ', total_text)
        # converting all the chars into lower-case.
        total_text = total_text.lower()
```



```

        for word in total_text.split():
            # if the word is a not a stop word then retain that word from the data
            if not word in stop_words:
                string += word + " "

        data_text[column][index] = string

[nltk_data] Downloading package stopwords to
[nltk_data] C:\Users\rdbz3b\AppData\Roaming\nltk_data...
[nltk_data] Package stopwords is already up-to-date!

```

```

In [5]: #text processing stage.
start_time = time.clock()
for index, row in data_text.iterrows():
    if type(row['TEXT']) is str:
        nlp_preprocessing(row['TEXT'], index, 'TEXT')
    else:
        print("there is no text description for id:",index)
print('Time took for preprocessing the text :',time.clock() - start_time, "seconds")

```

```

there is no text description for id: 1109
there is no text description for id: 1277
there is no text description for id: 1407
there is no text description for id: 1639
there is no text description for id: 2755
Time took for preprocessing the text : 169.357795375 seconds

```

```

In [6]: #merging both gene_variations and text data based on ID
result = pd.merge(data, data_text,on='ID', how='left')
result.head()

```

Out[6]:

	ID	Gene	Variation	Class	TEXT
0	0	FAM58A	Truncating Mutations	1	cyclin dependent kinases cdks regulate variety...
1	1	CBL	W802*	2	abstract background non small cell lung cancer...

	ID	Gene	Variation	Class	TEXT
2	2	CBL	Q249E	2	abstract background non small cell lung cancer...
3	3	CBL	N454D	3	recent evidence demonstrated acquired uniparen...
4	4	CBL	L399V	4	oncogenic mutations monomeric casitas b lineag...

In [7]: `result[result.isnull().any(axis=1)]`

Out[7]:

	ID	Gene	Variation	Class	TEXT
1109	1109	FANCA	S1088F	1	NaN
1277	1277	ARID5B	Truncating Mutations	1	NaN
1407	1407	FGFR3	K508M	6	NaN
1639	1639	FLT1	Amplification	6	NaN
2755	2755	BRAF	G596C	7	NaN

In [8]: `result.loc[result['TEXT'].isnull(), 'TEXT'] = result['Gene'] + ' ' + result['Variation']`

In [9]: `result[result['ID']==1109]`

Out[9]:

	ID	Gene	Variation	Class	TEXT
1109	1109	FANCA	S1088F	1	FANCA S1088F

3.1.4. Test, Train and Cross Validation Split

3.1.4.1. Splitting data into train, test and cross validation (64:20:16)

In [10]: `y_true = result['Class'].values
result.Gene = result.Gene.str.replace('\s+', '_')`

```

result.Variation = result.Variation.str.replace('\s+', '_')

# split the data into test and train by maintaining same distribution of
# output variable 'y_true' [stratify=y_true]
X_train, test_df, y_train, y_test = train_test_split(result, y_true, str
atify=y_true, test_size=0.2)
# split the train data into train and cross validation by maintaining s
ame distribution of output variable 'y_train' [stratify=y_train]
train_df, cv_df, y_train, y_cv = train_test_split(X_train, y_train, str
atify=y_train, test_size=0.2)

```

We split the data into train, test and cross validation data sets, preserving the ratio of class distribution in the original data set

```

In [11]: print('Number of data points in train data:', train_df.shape[0])
print('Number of data points in test data:', test_df.shape[0])
print('Number of data points in cross validation data:', cv_df.shape[0]
)

```

```

Number of data points in train data: 2124
Number of data points in test data: 665
Number of data points in cross validation data: 532

```

3.1.4.2. Distribution of y_i's in Train, Test and Cross Validation datasets

```

In [12]: # it returns a dict, keys as class labels and values as the number of d
ata points in that class
train_class_distribution = train_df['Class'].value_counts().sort_index
()
test_class_distribution = test_df['Class'].value_counts().sort_index()
cv_class_distribution = cv_df['Class'].value_counts().sort_index()

my_colors = 'rgbkymc'
train_class_distribution.plot(kind='bar')
plt.xlabel('Class')
plt.ylabel('Data points per Class')
plt.title('Distribution of yi in train data')

```

```

plt.grid()
plt.show()

# ref: argsort https://docs.scipy.org/doc/numpy/reference/generated/numpy.argsort.html
# -(train_class_distribution.values): the minus sign will give us in decreasing order
sorted_yi = np.argsort(-train_class_distribution.values)
for i in sorted_yi:
    print('Number of data points in class', i+1, ':', train_class_distribution.values[i], '(', np.round((train_class_distribution.values[i]/train_df.shape[0]*100), 3), '%)')

print('-'*80)
my_colors = 'rgbkymc'
test_class_distribution.plot(kind='bar')
plt.xlabel('Class')
plt.ylabel('Data points per Class')
plt.title('Distribution of yi in test data')
plt.grid()
plt.show()

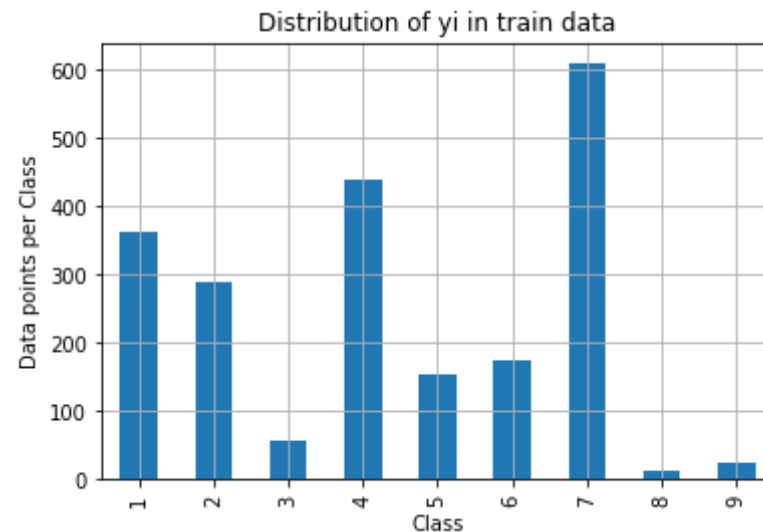
# ref: argsort https://docs.scipy.org/doc/numpy/reference/generated/numpy.argsort.html
# -(train_class_distribution.values): the minus sign will give us in decreasing order
sorted_yi = np.argsort(-test_class_distribution.values)
for i in sorted_yi:
    print('Number of data points in class', i+1, ':', test_class_distribution.values[i], '(', np.round((test_class_distribution.values[i]/test_df.shape[0]*100), 3), '%)')

print('-'*80)
my_colors = 'rgbkymc'
cv_class_distribution.plot(kind='bar')
plt.xlabel('Class')
plt.ylabel('Data points per Class')
plt.title('Distribution of yi in cross validation data')

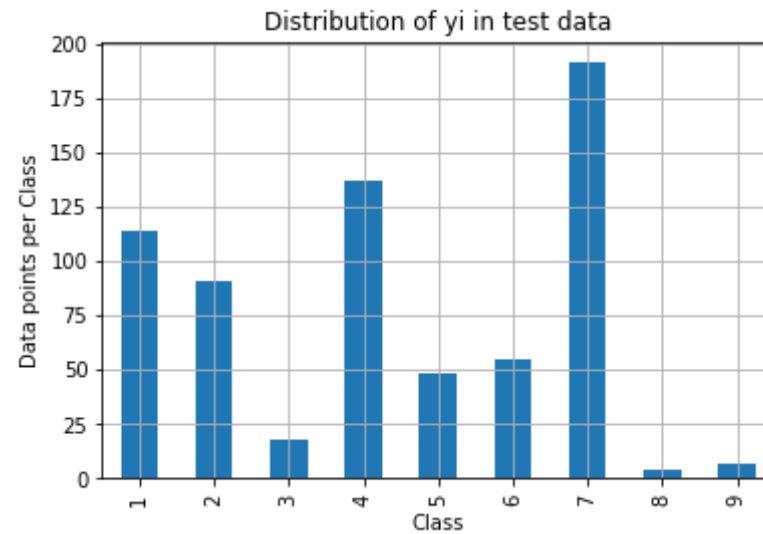
```

```
plt.grid()
plt.show()

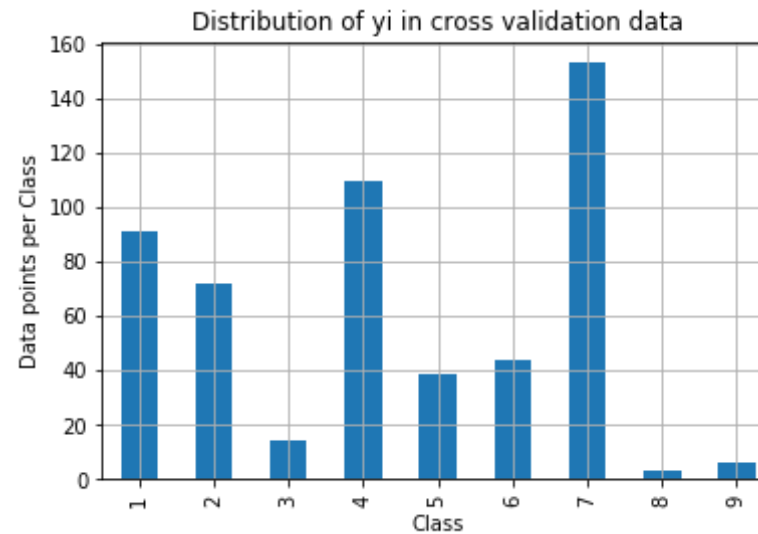
# ref: argsort https://docs.scipy.org/doc/numpy/reference/generated/numpy.argsort.html
# -(train_class_distribution.values): the minus sign will give us in decreasing order
sorted_yi = np.argsort(-train_class_distribution.values)
for i in sorted_yi:
    print('Number of data points in class', i+1, ': ', cv_class_distribution.values[i], ' (', np.round((cv_class_distribution.values[i]/cv_df.shape[0]*100), 3), '%)')
```



```
Number of data points in class 7 : 609 ( 28.672 %)
Number of data points in class 4 : 439 ( 20.669 %)
Number of data points in class 1 : 363 ( 17.09 %)
Number of data points in class 2 : 289 ( 13.606 %)
Number of data points in class 6 : 176 ( 8.286 %)
Number of data points in class 5 : 155 ( 7.298 %)
Number of data points in class 3 : 57 ( 2.684 %)
Number of data points in class 9 : 24 ( 1.13 %)
Number of data points in class 8 : 12 ( 0.565 %)
```



Number of data points in class 7 : 191 (28.722 %)
Number of data points in class 4 : 137 (20.602 %)
Number of data points in class 1 : 114 (17.143 %)
Number of data points in class 2 : 91 (13.684 %)
Number of data points in class 6 : 55 (8.271 %)
Number of data points in class 5 : 48 (7.218 %)
Number of data points in class 3 : 18 (2.707 %)
Number of data points in class 9 : 7 (1.053 %)
Number of data points in class 8 : 4 (0.602 %)



Number of data points in class 7 : 153 (28.759 %)
Number of data points in class 4 : 110 (20.677 %)
Number of data points in class 1 : 91 (17.105 %)
Number of data points in class 2 : 72 (13.534 %)
Number of data points in class 6 : 44 (8.271 %)
Number of data points in class 5 : 39 (7.331 %)
Number of data points in class 3 : 14 (2.632 %)
Number of data points in class 9 : 6 (1.128 %)
Number of data points in class 8 : 3 (0.564 %)

3.2 Prediction using a 'Random' Model

In a 'Random' Model, we generate the NINE class probabilities randomly such that they sum to 1.

```
In [13]: from collections import Counter, defaultdict
from sklearn.calibration import CalibratedClassifierCV
from sklearn.naive_bayes import MultinomialNB
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
```

```

from sklearn.model_selection import GridSearchCV
import math
from sklearn.metrics import normalized_mutual_info_score
from sklearn.ensemble import RandomForestClassifier
warnings.filterwarnings("ignore")

```

```

In [14]: # This function plots the confusion matrices given y_i, y_i_hat.
def plot_confusion_matrix(test_y, predict_y):
    C = confusion_matrix(test_y, predict_y)
    # C = 9,9 matrix, each cell (i,j) represents number of points of class i
    # are predicted class j

    A = (((C.T)/(C.sum(axis=1))).T)
    #divid each element of the confusion matrix with the sum of elements
    # in that column

    # C = [[1, 2],
    #      [3, 4]]
    # C.T = [[1, 3],
    #        [2, 4]]
    # C.sum(axis = 1) axis=0 corresponds to columns and axis=1 corresponds
    # to rows in two dimensional array
    # C.sum(axis=1) = [[3, 7]]
    # ((C.T)/(C.sum(axis=1))) = [[1/3, 3/7]
    #                             [2/3, 4/7]]

    # ((C.T)/(C.sum(axis=1))).T = [[1/3, 2/3]
    #                               [3/7, 4/7]]
    # sum of row elements = 1

    B = (C/C.sum(axis=0))
    #divid each element of the confusion matrix with the sum of elements
    # in that row
    # C = [[1, 2],
    #      [3, 4]]
    # C.sum(axis = 0) axis=0 corresponds to columns and axis=1 corresponds
    # to rows in two dimensional array
    # C.sum(axis=0) = [[4, 6]]
    # (C/C.sum(axis=0)) = [[1/4, 2/6],

```



```

# [3/4, 4/6]]

labels = [1,2,3,4,5,6,7,8,9]
# representing A in heatmap format
print("-"*20, "Confusion matrix", "-"*20)
plt.figure(figsize=(20,7))
sns.heatmap(C, annot=True, cmap="YlGnBu", fmt=".3f", xticklabels=labels, yticklabels=labels)
plt.xlabel('Predicted Class')
plt.ylabel('Original Class')
plt.show()

print("-"*20, "Precision matrix (Column Sum=1)", "-"*20)
plt.figure(figsize=(20,7))
sns.heatmap(B, annot=True, cmap="YlGnBu", fmt=".3f", xticklabels=labels, yticklabels=labels)
plt.xlabel('Predicted Class')
plt.ylabel('Original Class')
plt.show()

# representing B in heatmap format
print("-"*20, "Recall matrix (Row sum=1)", "-"*20)
plt.figure(figsize=(20,7))
sns.heatmap(A, annot=True, cmap="YlGnBu", fmt=".3f", xticklabels=labels, yticklabels=labels)
plt.xlabel('Predicted Class')
plt.ylabel('Original Class')
plt.show()

```

```

In [15]: # we need to generate 9 numbers and the sum of numbers should be 1
# one solution is to generate 9 numbers and divide each of the numbers
# by their sum
# ref: https://stackoverflow.com/a/18662466/4084039
test_data_len = test_df.shape[0]
cv_data_len = cv_df.shape[0]
train_data_len = train_df.shape[0]

# we create a output array that has exactly same size as the CV data

```

```

train_predicted_y = np.zeros((train_data_len,9))
for i in range(train_data_len):
    rand_probs = np.random.rand(1,9)
    train_predicted_y[i] = ((rand_probs/sum(sum(rand_probs))))[0]
print("Log loss on train Data using Random Model",log_loss(y_train,train_predicted_y, eps=1e-15))

# we create a output array that has exactly same size as the CV data
cv_predicted_y = np.zeros((cv_data_len,9))
for i in range(cv_data_len):
    rand_probs = np.random.rand(1,9)
    cv_predicted_y[i] = ((rand_probs/sum(sum(rand_probs))))[0]
print("Log loss on Cross Validation Data using Random Model",log_loss(y_cv,cv_predicted_y, eps=1e-15))

# Test-Set error.
#we create a output array that has exactly same as the test data
test_predicted_y = np.zeros((test_data_len,9))
for i in range(test_data_len):
    rand_probs = np.random.rand(1,9)
    test_predicted_y[i] = ((rand_probs/sum(sum(rand_probs))))[0]
print("Log loss on Test Data using Random Model",log_loss(y_test,test_predicted_y, eps=1e-15))

#summarizing data
random_best_alpha = None
random_cv_log_loss = log_loss(y_cv,cv_predicted_y, eps=1e-15)
random_test_log_loss = log_loss(y_test,test_predicted_y, eps=1e-15)
random_train_log_loss = log_loss(y_train,train_predicted_y, eps=1e-15)

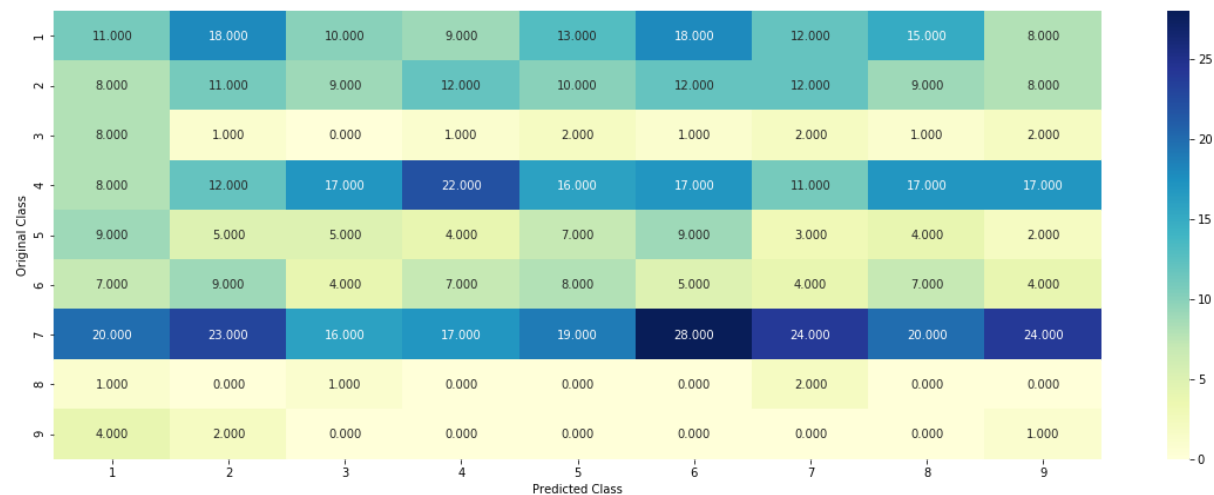
predicted_y =np.argmax(test_predicted_y, axis=1)
plot_confusion_matrix(y_test, predicted_y+1)

```

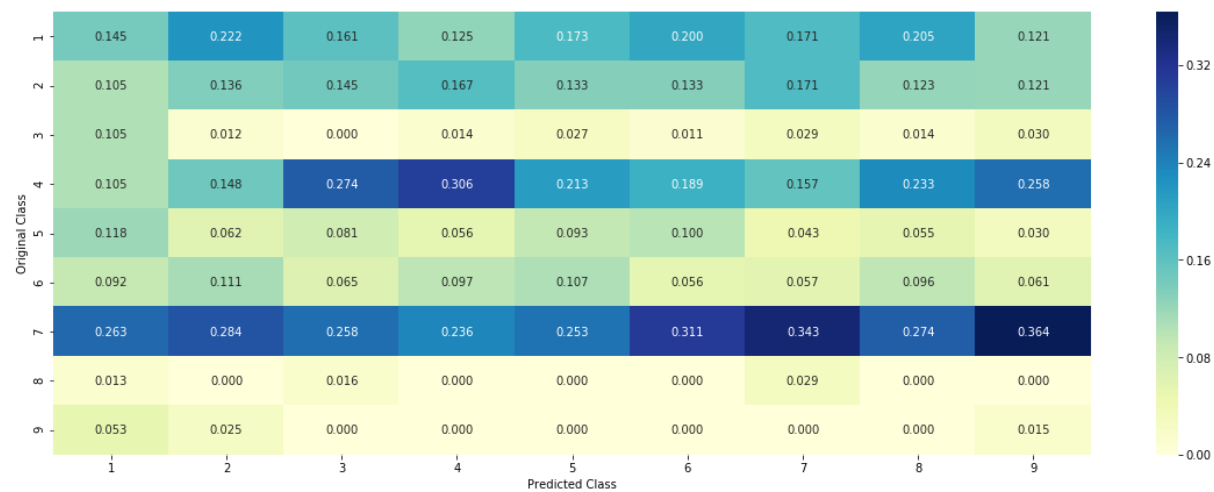
```

Log loss on train Data using Random Model 2.456188097717028
Log loss on Cross Validation Data using Random Model 2.485301808515957
Log loss on Test Data using Random Model 2.4235013336037117
----- Confusion matrix -----

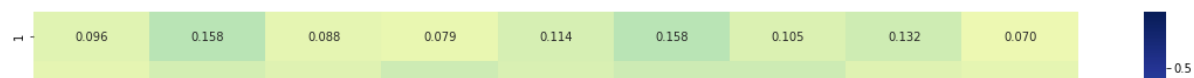
```

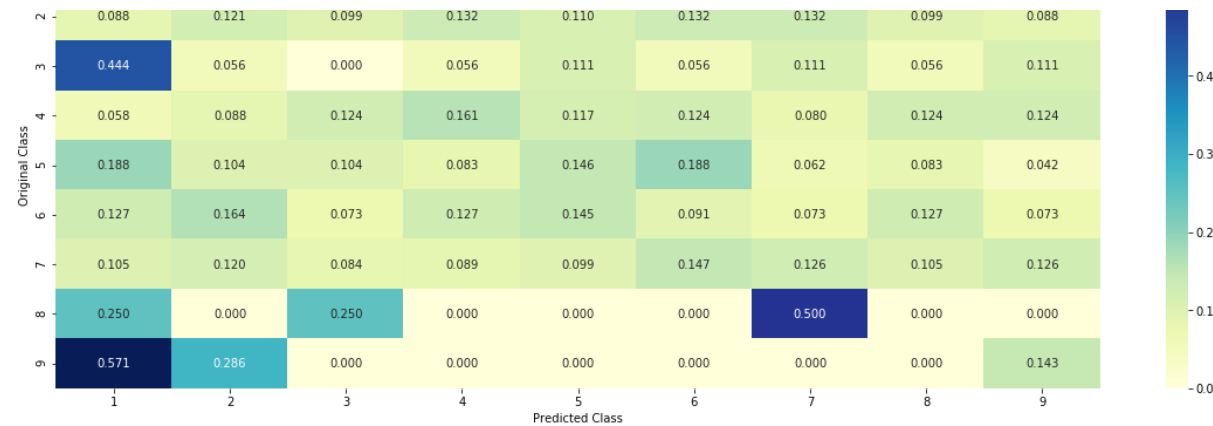


----- Precision matrix (Column Sum=1) -----



----- Recall matrix (Row sum=1) -----





3.3 Univariate Analysis

```
In [16]: # code for response coding with Laplace smoothing.
# alpha : used for laplace smoothing
# feature: ['gene', 'variation']
# df: ['train_df', 'test_df', 'cv_df']
# algorithm
# -----
# Consider all unique values and the number of occurrences of given feature in train data dataframe
# build a vector (1*9) , the first element = (number of times it occurred in class1 + 10*alpha / number of time it occurred in total data+90*alpha)
# gv_dict is like a look up table, for every gene it store a (1*9) representation of it
# for a value of feature in df:
# if it is in train data:
# we add the vector that was stored in 'gv_dict' look up table to 'gv_fea'
# if it is not there is train:
# we add [1/9, 1/9, 1/9, 1/9, 1/9, 1/9, 1/9, 1/9, 1/9] to 'gv_fea'
# return 'gv_fea'
# -----
```

```

# get_gv_fea_dict: Get Gene variation Feature Dict
def get_gv_fea_dict(alpha, feature, df):
    # value_count: it contains a dict like
    # print(train_df['Gene'].value_counts())
    # output:
    #      {BRCA1      174
    #       TP53      106
    #       EGFR       86
    #       BRCA2       75
    #       PTEN       69
    #       KIT        61
    #       BRAF       60
    #       ERBB2       47
    #       PDGFRA      46
    #       ...}
    # print(train_df['Variation'].value_counts())
    # output:
    # {
    #   Truncating_Mutations      63
    #   Deletion                   43
    #   Amplification              43
    #   Fusions                    22
    #   Overexpression             3
    #   E17K                      3
    #   Q61L                      3
    #   S222D                     2
    #   P130S                     2
    #   ...
    # }
    value_count = train_df[feature].value_counts()

    # gv_dict : Gene Variation Dict, which contains the probability array for each gene/variation
    gv_dict = dict()

    # denominator will contain the number of time that particular feature occurred in whole data
    for i, denominator in value_count.items():
        # vec will contain  $p(y_i=1/G_i)$  probability of gene/variation b

```

```

belongs to particular class
# vec is 9 dimensional vector
vec = []
for k in range(1,10):
    # print(train_df.loc[(train_df['Class']==1) & (train_df['Gene']=='BRCA1')])
    #
    #      ID      Gene      Variation      Class
    # 2470  2470  BRCA1      S1715C      1
    # 2486  2486  BRCA1      S1841R      1
    # 2614  2614  BRCA1      M1R      1
    # 2432  2432  BRCA1      L1657P      1
    # 2567  2567  BRCA1      T1685A      1
    # 2583  2583  BRCA1      E1660G      1
    # 2634  2634  BRCA1      W1718L      1
    # cls_cnt.shape[0] will return the number of rows

    cls_cnt = train_df.loc[(train_df['Class']==k) & (train_df[feature]==i)]

    # cls_cnt.shape[0](numerator) will contain the number of time that particular feature occurred in whole data
    vec.append((cls_cnt.shape[0] + alpha*10)/ (denominator + 90*alpha))

    # we are adding the gene/variation to the dict as key and vec as value
    gv_dict[i]=vec
return gv_dict

# Get Gene variation feature
def get_gv_feature(alpha, feature, df):
    # print(gv_dict)
    #      {'BRCA1': [0.20075757575757575, 0.03787878787878788, 0.0681818181818177, 0.13636363636363635, 0.25, 0.19318181818181818, 0.03787878787878788, 0.03787878787878788, 0.03787878787878788],
    #      'TP53': [0.32142857142857145, 0.061224489795918366, 0.061224489795918366, 0.27040816326530615, 0.061224489795918366, 0.066326530612244902, 0.051020408163265307, 0.051020408163265307, 0.056122448979591837],

```

```

# 'EGFR': [0.0568181818181816, 0.215909090909091, 0.0625,
0.0681818181818177, 0.0681818181818177, 0.0625, 0.34659090909091
2, 0.0625, 0.0568181818181816],
# 'BRCA2': [0.133333333333333, 0.0606060606060608, 0.06060
60606060608, 0.0787878787878782, 0.13939393939394, 0.3454545454
54546, 0.0606060606060608, 0.0606060606060608, 0.0606060606060
8],
# 'PTEN': [0.069182389937106917, 0.062893081761006289, 0.06918
2389937106917, 0.46540880503144655, 0.075471698113207544, 0.06289308176
1006289, 0.069182389937106917, 0.062893081761006289, 0.0628930817610062
89],
# 'KIT': [0.066225165562913912, 0.25165562913907286, 0.0728476
82119205295, 0.072847682119205295, 0.066225165562913912, 0.066225165562
913912, 0.27152317880794702, 0.066225165562913912, 0.06622516556291391
2],
# 'BRAF': [0.06666666666666666, 0.179999999999999, 0.073333
33333333334, 0.0733333333333334, 0.0933333333333338, 0.08000000000
0000002, 0.299999999999999, 0.06666666666666666, 0.0666666666666666
6],
# ...
# }
gv_dict = get_gv_fea_dict(alpha, feature, df)
# value_count is similar in get_gv_fea_dict
value_count = train_df[feature].value_counts()

# gv_fea: Gene_variation feature, it will contain the feature for e
ach feature value in the data
gv_fea = []
# for every feature values in the given data frame we will check if
it is there in the train data then we will add the feature to gv_fea
# if not we will add [1/9,1/9,1/9,1/9,1/9,1/9,1/9,1/9,1/9] to gv_fe
a
for index, row in df.iterrows():
    if row[feature] in dict(value_count).keys():
        gv_fea.append(gv_dict[row[feature]])
    else:
        gv_fea.append([1/9,1/9,1/9,1/9,1/9,1/9,1/9,1/9,1/9])
# gv_fea.append([-1,-1,-1,-1,-1,-1,-1,-1,-1])
return gv_fea

```

when we calculate the probability of a feature belongs to any particular class, we apply laplace smoothing

- $(\text{numerator} + 10 \cdot \alpha) / (\text{denominator} + 90 \cdot \alpha)$

3.2.1 Univariate Analysis on Gene Feature

Q1. Gene, What type of feature it is ?

Ans. Gene is a categorical variable

Q2. How many categories are there and How they are distributed?

```
In [17]: unique_genes = train_df['Gene'].value_counts()
print('Number of Unique Genes :', unique_genes.shape[0])
# the top 10 genes that occurred most
print(unique_genes.head(10))
```

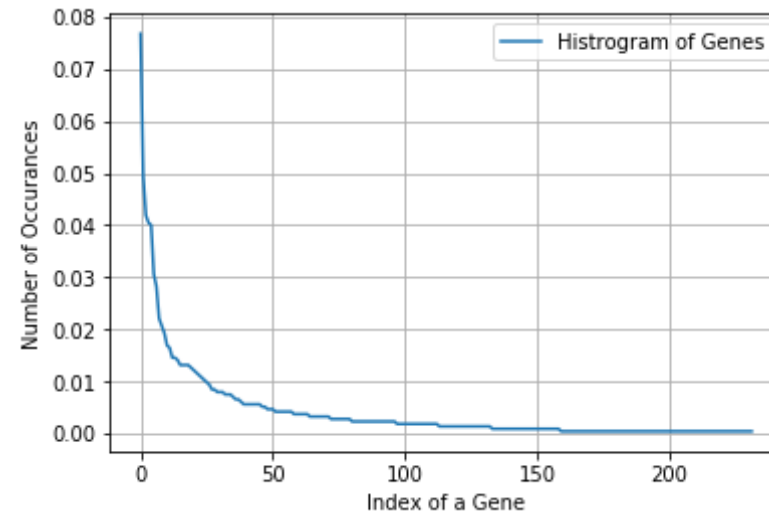
```
Number of Unique Genes : 232
BRCA1      163
TP53       105
EGFR        89
PTEN        86
BRCA2       85
KIT         65
BRAF        60
ALK         47
ERBB2       44
PDGFRA      41
Name: Gene, dtype: int64
```

```
In [18]: print("Ans: There are", unique_genes.shape[0], "different categories of
genes in the train data, and they are distributed as follows",)
```

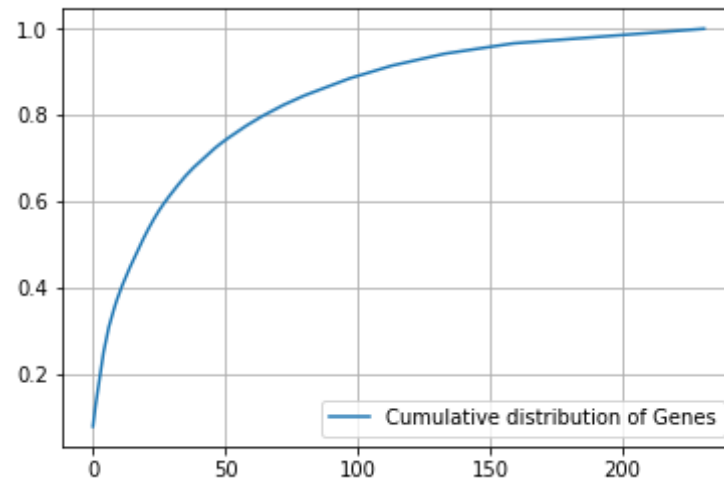
```
Ans: There are 232 different categories of genes in the train data, and
they are distributed as follows
```



```
In [19]: s = sum(unique_genes.values);  
h = unique_genes.values/s;  
plt.plot(h, label="Histogram of Genes")  
plt.xlabel('Index of a Gene')  
plt.ylabel('Number of Occurances')  
plt.legend()  
plt.grid()  
plt.show()
```



```
In [20]: c = np.cumsum(h)  
plt.plot(c, label='Cumulative distribution of Genes')  
plt.grid()  
plt.legend()  
plt.show()
```



Q3. How to featurize this Gene feature ?

Ans.there are two ways we can featurize this variable check out this video:
<https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/handling-categorical-and-numerical-features/>

1. One hot Encoding
2. Response coding

We will choose the appropriate featurization based on the ML model we use. For this problem of multi-class classification with categorical features, one-hot encoding is better for Logistic regression while response coding is better for Random Forests.

```
In [21]: #response-coding of the Gene feature
# alpha is used for laplace smoothing
alpha = 1
# train gene feature
train_gene_feature_responseCoding = np.array(get_gv_feature(alpha, "Gene", train_df))
# test gene feature
test_gene_feature_responseCoding = np.array(get_gv_feature(alpha, "Gene", test_df))
```

```
# cross validation gene feature
cv_gene_feature_responseCoding = np.array(get_gv_feature(alpha, "Gene",
cv_df))
```

```
In [22]: print("train_gene_feature_responseCoding is converted feature using res
pone coding method. The shape of gene feature:", train_gene_feature_res
ponseCoding.shape)
```

train_gene_feature_responseCoding is converted feature using response coding method. The shape of gene feature: (2124, 9)

```
In [23]: # one-hot encoding of Gene feature.
gene_vectorizer = CountVectorizer()
train_gene_feature_onehotCoding = gene_vectorizer.fit_transform(train_d
f['Gene'])
test_gene_feature_onehotCoding = gene_vectorizer.transform(test_df['Gen
e'])
cv_gene_feature_onehotCoding = gene_vectorizer.transform(cv_df['Gene'])
```

```
In [24]: train_df['Gene'].head()
```

```
Out[24]: 1296      HRAS
2594      BRCA1
3217      NTRK1
1432      SPOP
1369      AKT2
Name: Gene, dtype: object
```

```
In [25]: gene_features = gene_vectorizer.get_feature_names()
gene_features
```

```
Out[25]: ['abl1',
'acvr1',
'ago2',
'akt1',
'akt2',
'akt3',
'alk',
```

```
'apc',  
'ar',  
'araf',  
'arid1a',  
'arid2',  
'arid5b',  
'asxl1',  
'asxl2',  
'atm',  
'atrx',  
'aurka',  
'aurkb',  
'b2m',  
'bap1',  
'bard1',  
'bcl10',  
'bcl2',  
'bcl2l11',  
'bcor',  
'braf',  
'brca1',  
'brca2',  
'brd4',  
'brip1',  
'btk',  
'card11',  
'carm1',  
'casp8',  
'cbl',  
'ccnd1',  
'ccnd3',  
'ccne1',  
'cdh1',  
'cdk12',  
'cdk4',  
'cdk6',  
'cdkn1a',  
'cdkn1b',  
'cdkn2a',
```

'cdkn2b',
'cdkn2c',
'chek2',
'cic',
'crebbp',
'ctcf',
'ctla4',
'ctnnb1',
'ddr2',
'dicer1',
'dnmt3a',
'dnmt3b',
'dusp4',
'egfr',
'elf3',
'ep300',
'epas1',
'epcam',
'erbb2',
'erbb3',
'erbb4',
'ercc2',
'ercc3',
'ercc4',
'erg',
'errfi1',
'esr1',
'etv1',
'etv6',
'ewsr1',
'ezh2',
'fam58a',
'fanca',
'fat1',
'fbxw7',
'fgf3',
'fgf4',
'fgfr1',
'fgfr2',

```
'fgfr3',  
'fgfr4',  
'flt1',  
'flt3',  
'foxa1',  
'foxl2',  
'foxp1',  
'fubp1',  
'gata3',  
'gli1',  
'gnaq',  
'gnas',  
'h3f3a',  
'hist1h1c',  
'hla',  
'hnf1a',  
'hras',  
'idh1',  
'idh2',  
'igf1r',  
'il7r',  
'inpp4b',  
'jak1',  
'jak2',  
'kdm5a',  
'kdm5c',  
'kdm6a',  
'kdr',  
'keap1',  
'kit',  
'kmt2a',  
'kmt2b',  
'kmt2c',  
'kmt2d',  
'knstrn',  
'kras',  
'lats1',  
'map2k1',  
'map2k2',
```

```
'map2k4',  
'mapk1',  
'mdm4',  
'med12',  
'mef2b',  
'men1',  
'met',  
'mga',  
'mlh1',  
'mpl',  
'msh2',  
'msh6',  
'mtor',  
'myc',  
'mycn',  
'myd88',  
'ncor1',  
'nf1',  
'nf2',  
'nfe2l2',  
'nfkb1a',  
'nkx2',  
'notch1',  
'notch2',  
'npm1',  
'nras',  
'nsd1',  
'ntrk1',  
'ntrk2',  
'ntrk3',  
'nup93',  
'pbrm1',  
'pdgfra',  
'pdgfrb',  
'pik3ca',  
'pik3cb',  
'pik3cd',  
'pik3r1',  
'pik3r2',
```

'pim1',
'pms1',
'pms2',
'pole',
'ppm1d',
'ppp2r1a',
'ppp6c',
'prdm1',
'ptch1',
'pten',
'ptpn11',
'ptprd',
'ptprt',
'rab35',
'rac1',
'rad21',
'rad50',
'rad51b',
'rad51c',
'rad54l',
'raf1',
'rara',
'rasa1',
'rb1',
'rbm10',
'ret',
'rheb',
'rhoa',
'rit1',
'ros1',
'runx1',
'rxra',
'rybp',
'sdhb',
'setd2',
'sf3b1',
'shoc2',
'smad2',
'smad3',


```
'smad4',  
'smarca4',  
'smarcb1',  
'smo',  
'sos1',  
'sox9',  
'spop',  
'src',  
'srsf2',  
'stat3',  
'stk11',  
'tcf7l2',  
'tert',  
'tet1',  
'tet2',  
'tgfbr1',  
'tgfbr2',  
'tmprss2',  
'tp53',  
'tp53bp1',  
'tsc1',  
'tsc2',  
'u2af1',  
'vegfa',  
'vhl',  
'whsc1',  
'whsc1l1',  
'xpo1',  
'yap1']
```

```
In [26]: print("train_gene_feature_onehotCoding is converted feature using one-hot  
encoding method. The shape of gene feature:", train_gene_feature_one  
hotCoding.shape)
```

```
train_gene_feature_onehotCoding is converted feature using one-hot encoding  
method. The shape of gene feature: (2124, 231)
```

Q4. How good is this gene feature in predicting y_i ?

There are many ways to estimate how good a feature is, in predicting y_i . One of the good methods is to build a proper ML model using just this feature. In this case, we will build a logistic regression model using only Gene feature (one hot encoded) to predict y_i .

```
In [27]: alpha = [10 ** x for x in range(-5, 1)] # hyperparam for SGD classifier.

# read more about SGDClassifier() at http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html
# -----
# default parameters
# SGDClassifier(loss='hinge', penalty='l2', alpha=0.0001, l1_ratio=0.15, fit_intercept=True, max_iter=None, tol=None,
# shuffle=True, verbose=0, epsilon=0.1, n_jobs=1, random_state=None, learning_rate='optimal', eta0=0.0, power_t=0.5,
# class_weight=None, warm_start=False, average=False, n_iter=None)

# some of methods
# fit(X, y[, coef_init, intercept_init, ...])      Fit linear model with Stochastic Gradient Descent.
# predict(X)      Predict class labels for samples in X.

#-----
# video link:
#-----

cv_log_error_array=[]
for i in alpha:
    clf = SGDClassifier(alpha=i, penalty='l2', loss='log', random_state=42)
    clf.fit(train_gene_feature_onehotCoding, y_train)
    sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
    sig_clf.fit(train_gene_feature_onehotCoding, y_train)
    predict_y = sig_clf.predict_proba(cv_gene_feature_onehotCoding)
    cv_log_error_array.append(log_loss(y_cv, predict_y, labels=clf.classes_, eps=1e-15))
    print('For values of alpha = ', i, "The log loss is:", log_loss(y_cv
```

```

, predict_y, labels=clf.classes_, eps=1e-15))

fig, ax = plt.subplots()
ax.plot(alpha, cv_log_error_array, c='g')
for i, txt in enumerate(np.round(cv_log_error_array,3)):
    ax.annotate((alpha[i],np.round(txt,3)), (alpha[i],cv_log_error_array[i]))
plt.grid()
plt.title("Cross Validation Error for each alpha")
plt.xlabel("Alpha i's")
plt.ylabel("Error measure")
plt.show()

best_alpha = np.argmin(cv_log_error_array)
clf = SGDClassifier(alpha=alpha[best_alpha], penalty='l2', loss='log',
random_state=42)
clf.fit(train_gene_feature_onehotCoding, y_train)
sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
sig_clf.fit(train_gene_feature_onehotCoding, y_train)

# collecting data for summary
gene_best_alpha = best_alpha

predict_y = sig_clf.predict_proba(train_gene_feature_onehotCoding)
print('For values of best alpha = ', alpha[best_alpha], "The train log
loss is:", log_loss(y_train, predict_y, labels=clf.classes_, eps=1e-15
))
gene_train_log_loss = log_loss(y_train, predict_y, labels=clf.classes_,
eps=1e-15)

predict_y = sig_clf.predict_proba(cv_gene_feature_onehotCoding)
print('For values of best alpha = ', alpha[best_alpha], "The cross vali
dation log loss is:", log_loss(y_cv, predict_y, labels=clf.classes_, eps
=1e-15))
gene_cv_log_loss = log_loss(y_cv, predict_y, labels=clf.classes_, eps=1
e-15)

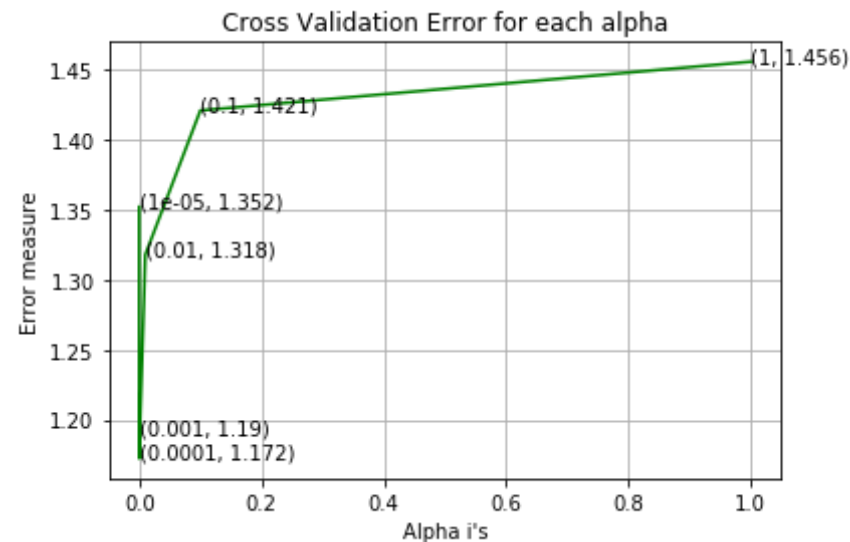
```

```

predict_y = sig_clf.predict_proba(test_gene_feature_onehotCoding)
print('For values of best alpha = ', alpha[best_alpha], "The test log loss is:", log_loss(y_test, predict_y, labels=clf.classes_, eps=1e-15))
gene_test_log_loss = log_loss(y_test, predict_y, labels=clf.classes_, eps=1e-15)

```

For values of alpha = 1e-05 The log loss is: 1.3522172807502544
 For values of alpha = 0.0001 The log loss is: 1.1722710197306832
 For values of alpha = 0.001 The log loss is: 1.1904507259828094
 For values of alpha = 0.01 The log loss is: 1.318004313302481
 For values of alpha = 0.1 The log loss is: 1.4211770942732322
 For values of alpha = 1 The log loss is: 1.4560434788253847



For values of best alpha = 0.0001 The train log loss is: 1.041718670770223
 For values of best alpha = 0.0001 The cross validation log loss is: 1.1722710197306832
 For values of best alpha = 0.0001 The test log loss is: 1.257956767116093

Q5. Is the Gene feature stable across all the data sets (Test, Train, Cross validation)?

Ans. Yes, it is. Otherwise, the CV and Test errors would be significantly more than train error.

```
In [28]: print("Q6. How many data points in Test and CV datasets are covered by  
the ", unique_genes.shape[0], " genes in train dataset?")  
  
test_coverage=test_df[test_df['Gene'].isin(list(set(train_df['Gene']  
)))]  
cv_coverage=cv_df[cv_df['Gene'].isin(list(set(train_df['Gene'])))]  
e[0]  
  
print('Ans\n1. In test data',test_coverage, 'out of',test_df.shape[0],  
":",(test_coverage/test_df.shape[0])*100)  
print('2. In cross validation data',cv_coverage, 'out of ',cv_df.shape[0],  
":", (cv_coverage/cv_df.shape[0])*100)  
  
gene_stability = (((test_coverage/test_df.shape[0])*100) + ((cv_coverage/cv_df.shape[0])*100))/2
```

Q6. How many data points in Test and CV datasets are covered by the 23
2 genes in train dataset?

Ans

1. In test data 644 out of 665 : 96.84210526315789

2. In cross validation data 516 out of 532 : 96.99248120300751

3.2.2 Univariate Analysis on Variation Feature

Q7. Variation, What type of feature is it ?

Ans. Variation is a categorical variable

Q8. How many categories are there?

```
In [29]: unique_variations = train_df['Variation'].value_counts()  
print('Number of Unique Variations :', unique_variations.shape[0])
```

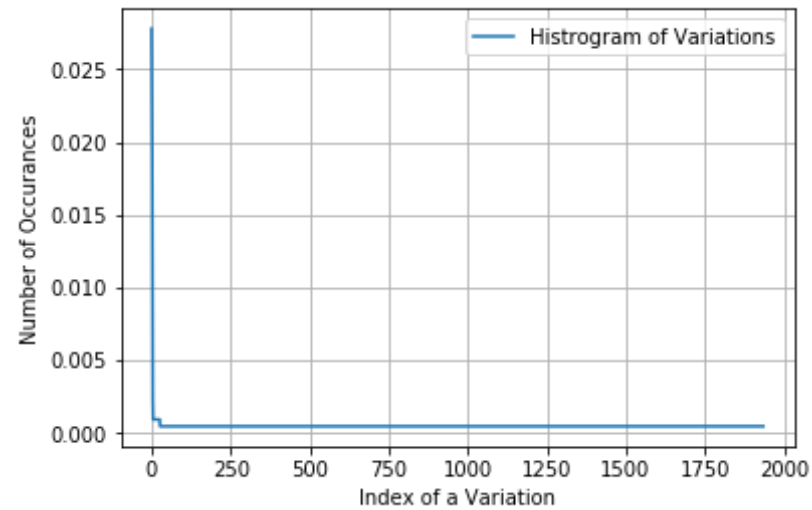
```
# the top 10 variations that occurred most
print(unique_variations.head(10))
```

```
Number of Unique Variations : 1935
Truncating_Mutations      59
Deletion                  49
Amplification             40
Fusions                   19
Overexpression            5
G13V                      2
T167A                     2
E330K                     2
Y42C                      2
TMPRSS2-ETV1_Fusion       2
Name: Variation, dtype: int64
```

```
In [30]: print("Ans: There are", unique_variations.shape[0] , "different categories of variations in the train data, and they are distributed as follows", )
```

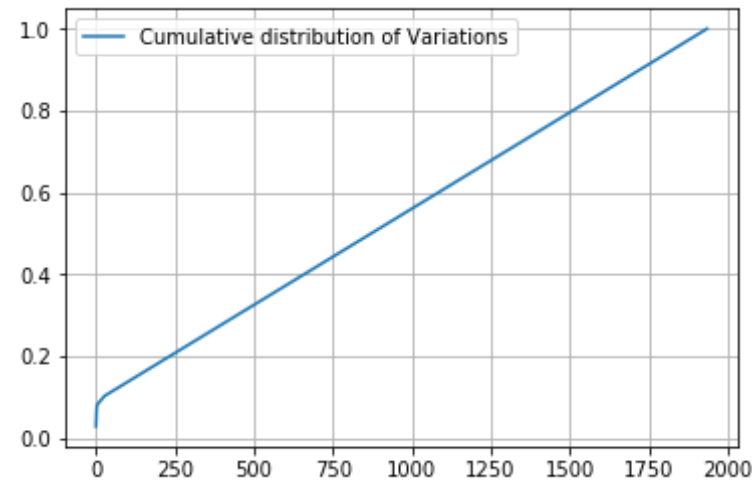
Ans: There are 1935 different categories of variations in the train data, and they are distributed as follows

```
In [31]: s = sum(unique_variations.values);
h = unique_variations.values/s;
plt.plot(h, label="Histogram of Variations")
plt.xlabel('Index of a Variation')
plt.ylabel('Number of Occurances')
plt.legend()
plt.grid()
plt.show()
```



```
In [32]: c = np.cumsum(h)
print(c)
plt.plot(c, label='Cumulative distribution of Variations')
plt.grid()
plt.legend()
plt.show()

[0.02777778 0.05084746 0.06967985 ... 0.99905838 0.99952919 1.          ]
```



Q9. How to featurize this Variation feature ?

Ans. There are two ways we can featurize this variable check out this video:
<https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/handling-categorical-and-numerical-features/>

1. One hot Encoding
2. Response coding

We will be using both these methods to featurize the Variation Feature

```
In [33]: # alpha is used for laplace smoothing
alpha = 1
# train gene feature
train_variation_feature_responseCoding = np.array(get_gv_feature(alpha,
"Variation", train_df))
# test gene feature
test_variation_feature_responseCoding = np.array(get_gv_feature(alpha,
"Variation", test_df))
# cross validation gene feature
cv_variation_feature_responseCoding = np.array(get_gv_feature(alpha, "V
ariation", cv_df))
```



```
In [34]: print("train_variation_feature_responseCoding is a converted feature using the response coding method. The shape of Variation feature:", train_variation_feature_responseCoding.shape)
```

train_variation_feature_responseCoding is a converted feature using the response coding method. The shape of Variation feature: (2124, 9)

```
In [35]: # one-hot encoding of variation feature.
variation_vectorizer = CountVectorizer()
train_variation_feature_onehotCoding = variation_vectorizer.fit_transform(train_df['Variation'])
test_variation_feature_onehotCoding = variation_vectorizer.transform(test_df['Variation'])
cv_variation_feature_onehotCoding = variation_vectorizer.transform(cv_df['Variation'])
```

```
In [36]: print("train_variation_feature_onehotEncoded is converted feature using the one-hot encoding method. The shape of Variation feature:", train_variation_feature_onehotCoding.shape)
```

train_variation_feature_onehotEncoded is converted feature using the one-hot encoding method. The shape of Variation feature: (2124, 1963)

```
In [37]: variation_features = variation_vectorizer.get_feature_names()
variation_features
```

```
Out[37]: ['126',
          '17',
          '1_2009trunc',
          '1_fusion',
          '2010_2471trunc',
          '256_286trunc',
          '385_418del',
          '422_605trunc',
          '534_536del',
          '560_561inser',
          '596_619splice',
          '981_1028splice',
```

```
'_baiap2l1_fusion',  
'_deletion',  
'a1022e',  
'a1066v',  
'a111p',  
'a1170v',  
'a11_g12insga',  
'a1200v',  
'a120s',  
'a121e',  
'a121p',  
'a1234t',  
'a126d',  
'a126g',  
'a126s',  
'a126v',  
'a134d',  
'a1374v',  
'a146t',  
'a146v',  
'a148t',  
'a149p',  
'a159t',  
'a161s',  
'a161t',  
'a1669s',  
'a1708v',  
'a1789t',  
'a1823t',  
'a1830t',  
'a1843p',  
'a1843t',  
'a197t',  
'a19v',  
'a2034v',  
'a211d',  
'a232v',  
'a2351g',  
'a2425t',
```

'a246p',
'a263v',
'a2643g',
'a272v',
'a2770t',
'a290t',
'a298t',
'a339v',
'a347t',
'a349p',
'a34d',
'a36p',
'a39p',
'a40e',
'a41p',
'a41t',
'a459v',
'a500t',
'a502_y503dup',
'a504_y505ins',
'a530t',
'a530v',
'a546d',
'a598t',
'a598v',
'a59g',
'a59t',
'a60v',
'a617t',
'a627t',
'a633t',
'a633v',
'a634d',
'a634v',
'a648t',
'a677g',
'a707t',
'a717g',
'a723d',

'a727v',
'a728v',
'a72s',
'a72v',
'a750_e758delinsp',
'a750p',
'a763_y764insfqa',
'a767_v769del',
'a77t',
'a829p',
'a864t',
'a883f',
'a8s',
'a95d',
'abl1_fusion',
'agk',
'ahcyll_fusion',
'akap9',
'alk_fusion',
'amplification',
'arv567es',
'atf1_fusion',
'atf7ip',
'bcl2_fusion',
'bcor',
'bcor_fusion',
'bcr',
'braf_fusion',
'brd4',
'c105f',
'c1156f',
'c1156y',
'c124r',
'c124s',
'c125s',
'c1265s',
'c134w',
'c135r',
'c135y',

'c136r',
'c1483r',
'c1483w',
'c1483y',
'c157y',
'c1697r',
'c1767s',
'c176f',
'c1787s',
'c2060g',
'c229r',
'c238f',
'c248t',
'c24r',
'c24y',
'c250t',
'c275s',
'c277q',
'c277r',
'c277w',
'c324y',
'c334s',
'c381a',
'c384r',
'c396r',
'c39r',
'c39s',
'c39y',
'c41y',
'c443y',
'c450_k451insmiewmi',
'c456_n468del',
'c47s',
'c49y',
'c569y',
'c582f',
'c609y',
'c611y',
'c618r',

'c61g',
'c620y',
'c630r',
'c634s',
'c634y',
'c696y',
'c706f',
'c712r',
'c77f',
'c91a',
'c91s',
'cad',
'ccdc6',
'ccdc6_fusion',
'ccnb3_fusion',
'ccnd1',
'cd74_fusion',
'cep110',
'cep85l',
'cic',
'cpeb1',
'cul1',
'd1010h',
'd1010n',
'd1010y',
'd101y',
'd1067a',
'd1067y',
'd106a',
'd107y',
'd108h',
'd1091n',
'd119n',
'd1270g',
'd1280v',
'd1344h',
'd1352y',
'd1384v',
'd140g',

'd153v',
'd1546n',
'd162g',
'd162h',
'd1709e',
'd171g',
'd1733g',
'd1739e',
'd1739v',
'd1739y',
'd1778g',
'd1778h',
'd1778n',
'd1818g',
'd1853n',
'd186a',
'd194y',
'd2033n',
'd2312v',
'd245v',
'd24y',
'd252g',
'd254n',
'd257n',
'd2665g',
'd2723g',
'd287h',
'd289_d292del',
'd289del',
'd29h',
'd300h',
'd300n',
'd3095e',
'd324n',
'd326n',
'd32h',
'd32n',
'd32y',
'd331g',

'd351h',
'd384n',
'd390y',
'd401n',
'd404g',
'd408y',
'd419del',
'd423n',
'd450e',
'd450h',
'd493a',
'd520n',
'd537e',
'd537y',
'd587h',
'd594v',
'd594y',
'd600_l601insfreyeyd',
'd603g',
'd603n',
'd609e',
'd60n',
'd617g',
'd61n',
'd61y',
'd631a',
'd631g',
'd641n',
'd661v',
'd67n',
'd717v',
'd737n',
'd74y',
'd761y',
'd769a',
'd769h',
'd770_p772dup',
'd806h',
'd808n',

'd814v',
'd816a',
'd816e',
'd816g',
'd816v',
'd816y',
'd820a',
'd820e',
'd820g',
'd821n',
'd835a',
'd835del',
'd835e',
'd835n',
'd835y',
'd837n',
'd83v',
'd842_h845del',
'd842_m844del',
'd842i',
'd842v',
'd842y',
'd846y',
'd84g',
'd84h',
'd84n',
'd84v',
'd887n',
'd927g',
'd92a',
'd92e',
'd92g',
'd96n',
'ddit3_fusion',
'deletion',
'delta',
'dnmt3b7',
'dux4_fusion',
'e1021k',

'e102_i103del',
'e1051k',
'e106g',
'e1071w',
'e1099k',
'e116k',
'e1210k',
'e124q',
'e1250k',
'e1346k',
'e1356g',
'e135k',
'e137k',
'e142d',
'e1660g',
'e1682k',
'e1682v',
'e168d',
'e1705k',
'e172k',
'e1799k',
'e17k',
'e1836k',
'e1935g',
'e1978',
'e2014k',
'e207k',
'e218',
'e221q',
'e23fs',
'e2419k',
'e258v',
'e2663v',
'e267g',
'e279k',
'e281k',
'e2856a',
'e285k',
'e285v',

'e286k',
'e311_k312del',
'e31k',
'e321g',
'e321k',
'e322k',
'e326l',
'e330k',
'e365k',
'e40l',
'e40q',
'e41a',
'e439del',
'e452k',
'e462g',
'e466k',
'e475k',
'e483',
'e490k',
'e49k',
'e501k',
'e525k',
'e541k',
'e542k',
'e542q',
'e545g',
'e545k',
'e552k',
'e554_k558del',
'e554_v559del',
'e571k',
'e579k',
'e586k',
'e598_y599insdvdfreye',
'e598_y599insglvqvtgssdneyfyvdfreye',
'e606g',
'e60l',
'e612_f613insgyvdfreyeydlkwefrprenlef',
'e622d',

'e626k',
'e627d',
'e636k',
'e685v',
'e69g',
'e709a',
'e709g',
'e709q',
'e709v',
'e70k',
'e717k',
'e719k',
'e731k',
'e746_s752delinsi',
'e746_t751delinsa',
'e746_t751delinsva',
'e746_t751linsip',
'e746q',
'e746v',
'e758g',
'e75g',
'e768d',
'e76a',
'e76k',
'e77k',
'e79k',
'e79q',
'e804g',
'e812k',
'e81k',
'e82d',
'e82g',
'e82v',
'e839k',
'e846k',
'e866k',
'e868g',
'e88k',
'e921k',

```
'e946',  
'e996k',  
'ebf1',  
'egfr',  
'egfrviii',  
'egfrvv',  
'ep300',  
'epigenetic_silencing',  
'erbb4_fusion',  
'erg_fusion',  
'erlin2',  
'esrp1',  
'etv1_fusion',  
'etv4_fusion',  
'etv6',  
'evil_fusion',  
'ewsr1',  
'exon_1_mutations',  
'exon_20_insertions',  
'exon_2_mutations',  
'ezr',  
'f1061w',  
'f1088sfs',  
'f1174i',  
'f1174l',  
'f119s',  
'f1200i',  
'f123i',  
'f1245c',  
'f1245v',  
'f133l',  
'f133v',  
'f134y',  
'f154l',  
'f156l',  
'f161l',  
'f1695l',  
'f1704s',  
'f1888i',
```

```
'f21a',  
'f241s',  
'f248s',  
'f317l',  
'f328v',  
'f341v',  
'f346v',  
'f347l',  
'f351l',  
'f460l',  
'f53l',  
'f53s',  
'f57c',  
'f57l',  
'f57v',  
'f590g',  
'f594_r595inssdneyfyvdf',  
'f594l',  
'f615s',  
'f691l',  
'f808l',  
'f81v',  
'f876l',  
'f877l',  
'f958v',  
'fam76a_fusion',  
'fev_fusion',  
'fgfr1',  
'fgfr1_fusion',  
'fgfr2',  
'fgfr3_',  
'fgfr3_fusion',  
'fig',  
'flt3_fusion',  
'flt3_internal_tandem_duplications',  
'fus',  
'fusions',  
'g106v',  
'g1123d',
```

'g1123s',
'g1125a',
'g114r',
'g1194d',
'g1201e',
'g1202r',
'g123r',
'g123s',
'g1269s',
'g127e',
'g127n',
'g128v',
'g129r',
'g12a',
'g12c',
'g12r',
'g12s',
'g12v',
'g13d',
'g13v',
'g14v',
'g1529r',
'g1567d',
'g161v',
'g163d',
'g1656d',
'g165e',
'g165r',
'g165v',
'g1706a',
'g1706e',
'g1738r',
'g1743r',
'g1763v',
'g1770v',
'g17e',
'g17v',
'g1803a',
'g1809k',

'g186r',
'g1971e',
'g199r',
'g2032r',
'g20e',
'g2101a',
'g216r',
'g2274v',
'g2430a',
'g244d',
'g244r',
'g244s',
'g245a',
'g245d',
'g245s',
'g248v',
'g251c',
'g253c',
'g266r',
'g271e',
'g2748d',
'g284r',
'g305r',
'g305w',
'g309a',
'g309e',
'g31a',
'g31r',
'g322s',
'g325a',
'g325r',
'g328e',
'g34e',
'g356a',
'g35a',
'g370c',
'g373r',
'g375c',
'g376r',

'g380r',
'g382d',
'g39e',
'g419v',
'g423r',
'g423v',
'g430c',
'g44s',
'g464a',
'g464r',
'g464v',
'g465e',
'g466a',
'g466v',
'g469a',
'g469del',
'g478c',
'g480w',
'g503v',
'g505s',
'g508s',
'g52r',
'g591v',
'g596c',
'g596r',
'g596v',
'g60e',
'g660d',
'g665a',
'g67r',
'g67s',
'g67w',
'g697c',
'g719a',
'g719c',
'g719d',
'g724s',
'g735s',
'g751r',

'g774v',
'g776delinslc',
'g776delinsvc',
'g776s',
'g778_p780dup',
'g785s',
'g796s',
'g810s',
'g81s',
'g829r',
'g853d',
'g85r',
'g863d',
'g87r',
'g881d',
'g909r',
'g936r',
'git2',
'golga4',
'gpiap1',
'h1047l',
'h1047r',
'h1047y',
'h1094l',
'h1094r',
'h1094y',
'h115n',
'h118p',
'h123d',
'h123q',
'h123y',
'h132y',
'h133q',
'h1382y',
'h1421y',
'h1686q',
'h168n',
'h168r',
'h1746n',

'h174r',
'h179l',
'h179n',
'h179q',
'h179r',
'h1904r',
'h1918y',
'h191d',
'h193n',
'h193p',
'h1966y',
'h206d',
'h2074n',
'h214q',
'h2428q',
'h355m',
'h396p',
'h396r',
'h398y',
'h410r',
'h412y',
'h41r',
'h570r',
'h61r',
'h643d',
'h65y',
'h662r',
'h68y',
'h694r',
'h697y',
'h701p',
'h773_v774insh',
'h773dup',
'h773l',
'h773y',
'h78q',
'h83y',
'h845_n848delinsp',
'h870r',

'h876q',
'h878y',
'h93d',
'h93q',
'h93r',
'h94y',
'hip1',
'hmga2',
'hypermethylation',
'i1018f',
'i1018w',
'i103n',
'i111a',
'i111r',
'i1170s',
'i1171n',
'i1171t',
'i122l',
'i122s',
'i122v',
'i1250t',
'i130m',
'i1349t',
'i157t',
'i15t',
'i1616n',
'i162m',
'i168f',
'i1766s',
'i18v',
'i195t',
'i204t',
'i219v',
'i21v',
'i2285v',
'i2500f',
'i251s',
'i2627f',
'i2675v',

'i26n',
'i279p',
'i289m',
'i28t',
'i290a',
'i290r',
'i326v',
'i32del',
'i33del',
'i347m',
'i391m',
'i408v',
'i42v',
'i448v',
'i463s',
'i47f',
'i491m',
'i49s',
'i538v',
'i559_d560insdkrmns',
'i562m',
'i638f',
'i642v',
'i68k',
'i744_k745delinskipvai',
'i767m',
'i836del',
'i843del',
'i867s',
'i89t',
'igh',
'igh_fusion',
'jak2_fusion',
'k101m',
'k1026e',
'k1062m',
'k111e',
'k111n',
'k117n',

'k117r',
'k11r',
'k120e',
'k120m',
'k120n',
'k125e',
'k125m',
'k125r',
'k128n',
'k1299e',
'k1434i',
'k1436q',
'k1452n',
'k147e',
'k162d',
'k1690n',
'k1702e',
'k179m',
'k181m',
'k218t',
'k2411t',
'k2729n',
'k288q',
'k289e',
'k28m',
'k291e',
'k292i',
'k292t',
'k320e',
'k335i',
'k373e',
'k375a',
'k376n',
'k382e',
'k38n',
'k398a',
'k420a',
'k428a',
'k442nfs',

'k45q',
'k45t',
'k467t',
'k483e',
'k483m',
'k499e',
'k508m',
'k513r',
'k525e',
'k526e',
'k52r',
'k539l',
'k550_w557del',
'k558_e562del',
'k558_v559del',
'k558n',
'k56n',
'k575m',
'k57e',
'k590r',
'k5n',
'k601n',
'k601q',
'k618t',
'k62r',
'k641n',
'k641r',
'k642e',
'k648n',
'k650e',
'k650m',
'k650q',
'k650t',
'k656e',
'k659n',
'k666m',
'k666t',
'k700e',
'k745_a750del',

'k745m',
'k753a',
'k79e',
'k806a',
'k830r',
'k83e',
'k83n',
'k975e',
'k97m',
'kank1',
'kdd',
'kdelr2',
'kiaa1549',
'kiaa1967_fusion',
'kif5b',
'l1026f',
'l1122v',
'l112p',
'l112r',
'l1152r',
'l115r',
'l117p',
'l118p',
'l118r',
'l1195v',
'l1198f',
'l1198p',
'l1224f',
'l1240v',
'l1267s',
'l128f',
'l1301r',
'l1407p',
'l1460p',
'l146r',
'l1574p',
'l158p',
'l1593p',
'l1596h',

'l1657p',
'l165p',
'l1678p',
'l1705p',
'l1780p',
'l180p',
'l1844r',
'l188q',
'l188v',
'l1904v',
'l1947r',
'l1951r',
'l19f',
'l209f',
'l2106p',
'l210r',
'l221r',
'l2230v',
'l230p',
'l232li',
'l234fs',
'l23f',
'l2431p',
'l251p',
'l274p',
'l2865v',
'l28p',
'l30f',
'l325f',
'l32p',
'l344p',
'l344r',
'l348s',
'l358r',
'l362r',
'l370fs',
'l37p',
'l387m',
'l388m',

'l390f',
'l399v',
'l412f',
'l424v',
'l43v',
'l448p',
'l455m',
'l46f',
'l46r',
'l481f',
'l485_p490del',
'l485_p490delinsf',
'l485_q494del',
'l485f',
'l493p',
'l493v',
'l49h',
'l507p',
'l52r',
'l535p',
'l536h',
'l536r',
'l559r',
'l576del',
'l576p',
'l585i',
'l597q',
'l597s',
'l597v',
'l607i',
'l611_e612inscssdneyfyvdfreyeydlkwefprenl',
'l611v',
'l617f',
'l617m',
'l622h',
'l63p',
'l67p',
'l692f',
'l702h',

```
'l703p',
'l704n',
'l708p',
'l726f',
'l726i',
'l747_e749del',
'l747_p753delinss',
'l747f',
'l747s',
'l749p',
'l755s',
'l833v',
'l838p',
...]
```

Q10. How good is this Variation feature in predicting y_i ?

Let's build a model just like the earlier!

```
In [38]: alpha = [10 ** x for x in range(-5, 1)]

# read more about SGDClassifier() at http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html
# -----
# default parameters
# SGDClassifier(loss='hinge', penalty='l2', alpha=0.0001, l1_ratio=0.15, fit_intercept=True, max_iter=None, tol=None,
# shuffle=True, verbose=0, epsilon=0.1, n_jobs=1, random_state=None, learning_rate='optimal', eta0=0.0, power_t=0.5,
# class_weight=None, warm_start=False, average=False, n_iter=None)

# some of methods
# fit(X, y[, coef_init, intercept_init, ...])    Fit linear model with Stochastic Gradient Descent.
# predict(X)    Predict class labels for samples in X.

#-----
# video link:
```

```

#-----

cv_log_error_array=[]
for i in alpha:
    clf = SGDClassifier(alpha=i, penalty='l2', loss='log', random_state
=42)
    clf.fit(train_variation_feature_onehotCoding, y_train)

    sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
    sig_clf.fit(train_variation_feature_onehotCoding, y_train)
    predict_y = sig_clf.predict_proba(cv_variation_feature_onehotCoding
)

    cv_log_error_array.append(log_loss(y_cv, predict_y, labels=clf.clas
ses_, eps=1e-15))
    print('For values of alpha = ', i, "The log loss is:",log_loss(y_cv
, predict_y, labels=clf.classes_, eps=1e-15))

fig, ax = plt.subplots()
ax.plot(alpha, cv_log_error_array,c='g')
for i, txt in enumerate(np.round(cv_log_error_array,3)):
    ax.annotate((alpha[i],np.round(txt,3)), (alpha[i],cv_log_error_arra
y[i]))
plt.grid()
plt.title("Cross Validation Error for each alpha")
plt.xlabel("Alpha i's")
plt.ylabel("Error measure")
plt.show()

best_alpha = np.argmin(cv_log_error_array)
clf = SGDClassifier(alpha=alpha[best_alpha], penalty='l2', loss='log',
random_state=42)
clf.fit(train_variation_feature_onehotCoding, y_train)
sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
sig_clf.fit(train_variation_feature_onehotCoding, y_train)

# collecting data for summary

```

```

variation_best_alpha = best_alpha

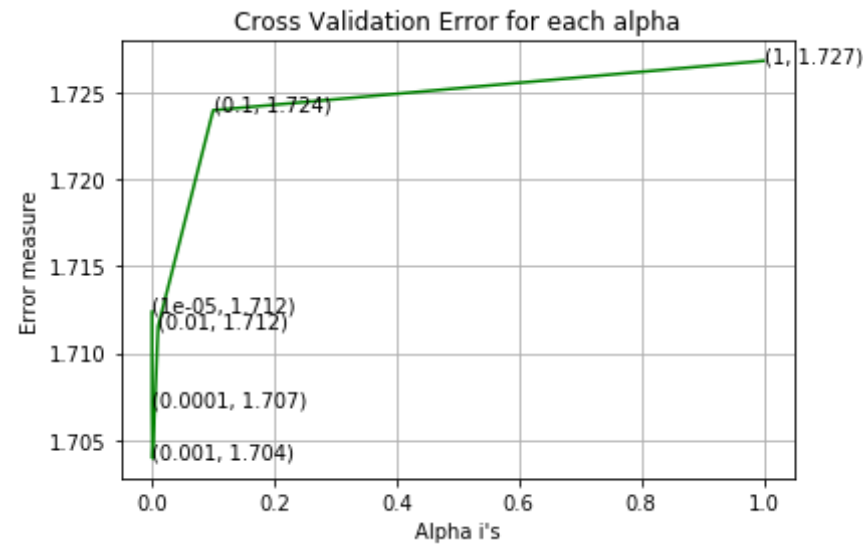
predict_y = sig_clf.predict_proba(train_variation_feature_onehotCoding)
variation_train_log_loss = log_loss(y_train, predict_y, labels=clf.classes_,
eps=1e-15)
print('For values of best alpha = ', alpha[best_alpha], "The train log
loss is:",log_loss(y_train, predict_y, labels=clf.classes_, eps=1e-15
))

predict_y = sig_clf.predict_proba(cv_variation_feature_onehotCoding)
variation_cv_log_loss = log_loss(y_cv, predict_y, labels=clf.classes_,
eps=1e-15)
print('For values of best alpha = ', alpha[best_alpha], "The cross vali
dation log loss is:",log_loss(y_cv, predict_y, labels=clf.classes_, eps
=1e-15))

predict_y = sig_clf.predict_proba(test_variation_feature_onehotCoding)
variation_test_log_loss = log_loss(y_test, predict_y, labels=clf.class
es_, eps=1e-15)
print('For values of best alpha = ', alpha[best_alpha], "The test log l
oss is:",log_loss(y_test, predict_y, labels=clf.classes_, eps=1e-15))

```

For values of alpha = 1e-05 The log loss is: 1.7123813814330526
 For values of alpha = 0.0001 The log loss is: 1.7070123834723545
 For values of alpha = 0.001 The log loss is: 1.703931262692979
 For values of alpha = 0.01 The log loss is: 1.7115335328403893
 For values of alpha = 0.1 The log loss is: 1.7239999715309355
 For values of alpha = 1 The log loss is: 1.7268510328334596



For values of best alpha = 0.001 The train log loss is: 1.0965469452519268

For values of best alpha = 0.001 The cross validation log loss is: 1.703931262692979

For values of best alpha = 0.001 The test log loss is: 1.6981929725933396

Q11. Is the Variation feature stable across all the data sets (Test, Train, Cross validation)?

Ans. Not sure! But lets be very sure using the below analysis.

```
In [39]: print("Q12. How many data points are covered by total ", unique_variations.shape[0], " genes in test and cross validation data sets?")
test_coverage=test_df[test_df['Variation'].isin(list(set(train_df['Variation'])))]
cv_coverage=cv_df[cv_df['Variation'].isin(list(set(train_df['Variation'])))]
print('Ans\n1. In test data',test_coverage, 'out of',test_df.shape[0],
":", (test_coverage/test_df.shape[0])*100)
print('2. In cross validation data',cv_coverage, 'out of ',cv_df.shape[0],
":", (cv_coverage/cv_df.shape[0])*100)
```

```
0], ":" , (cv_coverage/cv_df.shape[0])*100)
```

```
variation_stability = (((test_coverage/test_df.shape[0])*100) + ((cv_coverage/cv_df.shape[0])*100))/2
```

Q12. How many data points are covered by total 1935 genes in test and cross validation data sets?

Ans

1. In test data 76 out of 665 : 11.428571428571429

2. In cross validation data 52 out of 532 : 9.774436090225564

3.2.3 Univariate Analysis on Text Feature

1. How many unique words are present in train data?
2. How are word frequencies distributed?
3. How to featurize text field?
4. Is the text feature useful in predicting y_i ?
5. Is the text feature stable across train, test and CV datasets?

```
In [40]: # cls_text is a data frame
# for every row in data fram consider the 'TEXT'
# split the words by space
# make a dict with those words
# increment its count whenever we see that word
```

```
def extract_dictionary_paddle(cls_text):
    dictionary = defaultdict(int)
    for index, row in cls_text.iterrows():
        for word in row['TEXT'].split():
            dictionary[word] +=1
    return dictionary
```

```
In [41]: import math
#https://stackoverflow.com/a/1602964
def get_text_responsecoding(df):
```

```

text_feature_responseCoding = np.zeros((df.shape[0],9))
for i in range(0,9):
    row_index = 0
    for index, row in df.iterrows():
        sum_prob = 0
        for word in row['TEXT'].split():
            sum_prob += math.log(((dict_list[i].get(word,0)+10)/(total_dict.get(word,0)+90)))
        text_feature_responseCoding[row_index][i] = math.exp(sum_prob/len(row['TEXT'].split()))
        row_index += 1
    return text_feature_responseCoding

```

```

In [42]: # building a tfidfvectorizer with all the words that occurred minimum 3
         times in train data
text_vectorizer = TfidfVectorizer(min_df=5)
train_text_feature_onehotCoding = text_vectorizer.fit_transform(train_df['TEXT'])

# getting all the feature names (words)
tfidf_train_text_onehotencoding= text_vectorizer.get_feature_names()

# creating dictionary
dictionary = dict(zip(text_vectorizer.get_feature_names(), list(text_vectorizer.idf_)))
tfidf_features = set(text_vectorizer.get_feature_names())

# train_text_feature_onehotCoding.sum(axis=0).A1 will sum every row and
# returns (1*number of features) vector
train_text_fea_counts = train_text_feature_onehotCoding.sum(axis=0).A1

# zip(list(text_features),text_fea_counts) will zip a word with its number
# of times it occurred
#text_fea_dict = dict(zip(list(tfidf_train_text_onehotencodingfeatures),train_text_fea_counts))
text_fea_dict = dict(text_vectorizer.vocabulary_)

```



```
print("Total number of unique words in train data :", len(tfidf_train_t  
ext_onehotencoding))
```

Total number of unique words in train data : 38762

In [43]: text_fea_dict

```
Out[43]: {'ras': 30170,  
          'key': 20650,  
          'signaling': 32910,  
          'molecule': 23826,  
          'living': 21853,  
          'cells': 8692,  
          'mutations': 24270,  
          'involved': 19831,  
          '30': 1978,  
          'human': 18309,  
          'cancers': 8085,  
          'becoming': 6779,  
          'progressively': 28799,  
          'clear': 9319,  
          'spatial': 33399,  
          'arrangement': 5869,  
          'proteins': 28954,  
          'within': 38124,  
          'cell': 8683,  
          'chemical': 8990,  
          'structure': 33988,  
          'important': 19118,  
          'aspect': 6071,  
          'function': 15653,  
          'work': 38176,  
          'use': 37033,  
          'series': 32619,  
          'quantitative': 29554,  
          'physical': 27657,  
          'techniques': 35046,
```

'map': 22612,
'tendency': 35098,
'two': 36376,
'molecules': 23827,
'bind': 6992,
'together': 35659,
'form': 15404,
'dimer': 12126,
'membrane': 23073,
'surfaces': 34359,
'insights': 19501,
'well': 37990,
'technical': 35043,
'assays': 6091,
'developed': 11922,
'may': 22797,
'help': 17610,
'discover': 12241,
'new': 24981,
'therapeutic': 35295,
'drugs': 12720,
'capable': 8133,
'modulating': 23796,
'behavior': 6799,
'cancer': 8055,
'keywords': 20653,
'assay': 6082,
'go': 16748,
'abstract': 4220,
'lipid': 21797,
'anchored': 5304,
'small': 33163,
'gtpase': 17023,
'node': 25171,
'mammalian': 22546,
'number': 25574,
'observations': 25646,
'suggest': 34193,
'laterally': 21374,

'organized': 26049,
'play': 27903,
'regulatory': 30637,
'role': 31387,
'activation': 4417,
'anchors': 5306,
'composed': 9798,
'palmitoyl': 26649,
'farnesyl': 14747,
'moieties': 23813,
'widely': 38057,
'suspected': 34405,
'responsible': 31010,
'guiding': 17066,
'protein': 28945,
'organization': 26045,
'membranes': 23075,
'report': 30832,
'forms': 15427,
'binding': 6995,
'interface': 19628,
'y64a': 38506,
'point': 28055,
'mutation': 24258,
'switch': 34474,
'ii': 18827,
'region': 30604,
'known': 20825,
'prevent': 28589,
'son': 33306,
'sevenless': 32670,
'pi3k': 27681,
'effector': 13310,
'interactions': 19592,
'abolishes': 4194,
'formation': 15413,
'suggests': 34199,
'near': 24764,
'nucleotide': 25562,

'cleft': 9335,
'either': 13379,
'part': 26769,
'allosterically': 5050,
'coupled': 10405,
'tethering': 35164,
'bilayers': 6971,
'via': 37591,
'tail': 34843,
'show': 32830,
'mediated': 22951,
'require': 30903,
'anchor': 5301,
'clustering': 9404,
'quantitatively': 29555,
'characterize': 8942,
'dimerization': 12130,
'supported': 34323,
'using': 37043,
'combination': 9644,
'fluorescence': 15286,
'correlation': 10307,
'spectroscopy': 33462,
'photon': 27628,
'counting': 10400,
'histogram': 17947,
'analysis': 5260,
'time': 35535,
'resolved': 30977,
'anisotropy': 5411,
'single': 32987,
'tracking': 35809,
'step': 33794,
'photobleaching': 27617,
'2d': 1893,
'kd': 20594,
'measured': 22916,
'103': 332,
'm2': 22335,

'higher': 17839,
'order': 26018,
'oligomers': 25821,
'observed': 25649,
'occurs': 25701,
'surface': 34356,
'strictly': 33955,
'monomeric': 23872,
'comparable': 9714,
'densities': 11655,
'solution': 33282,
'constructs': 10072,
'including': 19198,
'changes': 8914,
'lipidation': 21798,
'pattern': 26895,
'hypervariable': 18496,
'general': 16277,
'property': 28889,
'native': 24677,
'signal': 32907,
'transduction': 35891,
'functions': 15663,
'binary': 6991,
'fundamental': 15665,
'processes': 28735,
'proliferation': 28821,
'differentiation': 12068,
'survival': 34392,
'network': 24891,
'hub': 18290,
'various': 37453,
'upstream': 36995,
'pathways': 26876,
'activate': 4412,
'gdp': 16219,
'gtp': 17020,
'subsequently': 34104,
'selects': 32450,

'multiple': 24163,
'downstream': 12645,
'effectors': 13311,
'elicit': 13460,
'varied': 37448,
'specific': 33424,
'biochemical': 7016,
'response': 31003,
'specificity': 33430,
'achieved': 4352,
'conformational': 9941,
'plasticity': 27878,
'dynamic': 12841,
'control': 10166,
'isoform': 19953,
'posttranslational': 28265,
'targets': 34930,
'main': 22482,
'isoforms': 19954,
'different': 12060,
'subdomains': 34059,
'plasma': 27865,
'10': 288,
'example': 14212,
'localizes': 21928,
'cholesterol': 9097,
'sensitive': 32494,
'domains': 12554,
'whereas': 38030,
'11': 436,
'common': 9689,
'terminal': 35126,
'moiety': 23814,
'operates': 25944,
'concert': 9861,
'one': 25895,
'groups': 16955,
'basic': 6677,
'sequence': 32540,

'six': 33037,
'lysines': 22263,
'ras4b': 30178,
'12': 551,
'provide': 29018,
'primary': 28627,
'anchorage': 5302,
'importantly': 19119,
'domain': 12545,
'residues': 30956,
'166': 960,
'hvr': 18352,
'167': 966,
'189': 1182,
'dynamically': 12843,
'modulate': 23793,
'localization': 21923,
'preference': 28474,
'distinct': 12380,
'populations': 28185,
'13': 666,
'away': 6455,
'necessary': 24776,
'efficient': 13319,
'raf': 30072,
'loading': 21906,
'promotes': 28855,
'redistribution': 30510,
'mechanism': 22927,
'requires': 30907,
'14': 746,
'however': 18189,
'molecular': 23822,
'details': 11876,
'coupling': 10407,
'partitioning': 26786,
'dependent': 11698,
'remain': 30732,
'unclear': 36685,

'addition': 4496,
'evidence': 14152,
'communication': 9698,
'pocket': 28050,
'nmr': 25152,
'ir': 19870,
'spectroscopic': 33461,
'insertion': 19490,
'affects': 4673,
'orientation': 26062,
'15': 810,
'17': 997,
'dynamics': 12844,
'md': 22873,
'modeling': 23762,
'bilayer': 6970,
'induced': 19303,
'conformations': 9943,
'identified': 18740,
'states': 33742,
'differ': 12053,
'conformation': 9938,
'contacts': 10087,
'18': 1109,
'vivo': 37696,
'fret': 15565,
'measurements': 22918,
'consistent': 10026,
'reorientation': 30784,
'respect': 30987,
'upon': 36983,
'19': 1197,
'20': 1413,
'showed': 32832,
'canonical': 8119,
'regions': 30607,
'communicate': 9695,
'across': 4388,
'long': 21972,

'range': 30135,
'side': 32883,
'chain': 8893,
'21': 1524,
'selection': 32442,
'22': 1570,
'allosteric': 5049,
'modes': 23777,
'likely': 21726,
'contribute': 10157,
'direct': 12179,
'functional': 15656,
'consequences': 10002,
'poorly': 28174,
'understood': 36755,
'members': 23069,
'superfamily': 34265,
'gtpases': 17026,
'considered': 10019,
'23': 1631,
'several': 32674,
'subfamilies': 34062,
'dimerize': 12131,
'24': 1676,
'28': 1819,
'class': 9289,
'instead': 19519,
'activating': 4416,
'gaps': 16076,
'activity': 4433,
'29': 1849,
'recently': 30395,
'shown': 32834,
'cluster': 9402,
'vitro': 37684,
'manner': 22586,
'broadly': 7519,
'mechanics': 22926,
'mm': 23700,

'dimers': 12135,
'could': 10369,
'recruited': 30475,
'soluble': 33280,
'31': 2020,
'artificial': 5920,
'gst': 16992,
'fused': 15690,
'leads': 21435,
'hypothesized': 18529,
'exist': 14287,
'32': 2066,
'presumed': 28574,
'detected': 11880,
'cross': 10537,
'linking': 21791,
'intrinsic': 19768,
'oligomeric': 25814,
'properties': 28888,
'unknown': 36856,
'microscopy': 23423,
'c118s': 7720,
'181': 1120,
'184': 1148,
'referred': 30547,
'eliminate': 13467,
'effects': 13312,
'preserving': 28557,
'palmitoylation': 26650,
'site': 33016,
'predicted': 28432,
'undergo': 36726,
'large': 21344,
'exchange': 14234,
'labeling': 21249,
'fluorescent': 15287,
'linked': 21788,
'fcs': 14817,
'density': 11660,

'clusters': 9405,
'reveal': 31120,
'formed': 15420,
'loop': 21989,
'beta': 6894,
'strand': 33900,
'alpha': 5069,
'helix': 17606,
'suggesting': 34195,
'corresponding': 10313,
'broad': 7511,
'33': 2102,
'35': 2153,
'structural': 33986,
'change': 8910,
'comparing': 9726,
'singly': 32993,
'doubly': 12620,
'reveals': 31124,
'insensitive': 19484,
'results': 31045,
'exhibits': 14285,
'reduced': 30514,
'translational': 35954,
'rotational': 31441,
'mobility': 23747,
'experiments': 14366,
'attached': 6243,
'cysteines': 10870,
'maleimide': 22516,
'sn': 33197,
'glycero': 16708,
'carboxamide': 8174,
'mcc': 22823,
'fig': 15000,
'la': 1306,
'fully': 15647,
'result': 31040,
'avoided': 6440,

'takes': 34867,
'place': 27834,
'cysteine': 10869,
'lateral': 21373,
'diffusion': 12084,
'possible': 28236,
'orientations': 26064,
'tethered': 35163,
'onto': 25917,
'modified': 23785,
'ref': 30537,
'secondary': 32368,
'aa': 4045,
'cartoon': 8266,
'mode': 23757,
'portion': 28203,
'color': 9621,
'allows': 5055,
'lipids': 21799,
'monitored': 23850,
'simultaneously': 32976,
'spot': 33562,
'lb': 1311,
'percentage': 27184,
'005': 87,
'mol': 23815,
'texas': 35198,
'red': 30496,
'tr': 35796,
'included': 19195,
'loaded': 21905,
'normalized': 25374,
'autocorrelation': 6355,
'fluctuations': 15277,
'channels': 8917,
'illustrated': 18902,
'lc': 1313,
'times': 35541,
'correspond': 10310,

'coefficients': 9491,
'39': 2263,
'04': 183,
'respectively': 30990,
'faster': 14771,
'providing': 29023,
'confirmation': 9924,
'sites': 33024,
'unrestricted': 36915,
'dominated': 12568,
'component': 9794,
'36': 2180,
'37': 2202,
'38': 2238,
'expected': 14344,
'40': 2357,
'pronounced': 28870,
'lower': 22029,
'observe': 25648,
'additional': 4497,
'originally': 26071,
'sos': 33334,
'interaction': 19586,
'blocking': 7210,
'41': 2390,
'mutant': 24250,
'diffuse': 12081,
'identical': 18734,
'rates': 30225,
'ld': 1319,
'situated': 33034,
'opposite': 25970,
'proximal': 29037,
'terminus': 35137,
'provides': 29022,
'average': 6420,
'value': 37405,
'probe': 28713,
'distribution': 12401,

```
'ensemble': 13745,  
'used': 37035,  
'field': 14994,  
'view': 37614,  
'movie': 24001,  
's1': 31626,  
'particles': 26781,  
'individually': 19295,  
'tracked': 35808,  
'42': 2418,  
's2': 31724,  
'size': 33045,  
'histograms': 17948,  
'le': 1325,  
'characterized': 8943,  
'shorter': 32822,  
'steps': 33798,  
'relative': 30686,  
'infer': 19345,  
'fitting': 15182,  
'distributions': 12402,  
'equation': 13897,  
'cylindrical': 10833,  
'coordinates': 10229,  
'si': 32866,  
'materials': 22740,  
'methods': 23232,  
'described': 11778,  
'species': 33421,  
'yielding': 38614,  
'54': 2825,  
'05': 202,  
'model': 23758,  
'cannot': 8118,  
'describe': 11777,  
'indicating': 19278,  
'contains': 10093,  
'fast': 14770,  
'similar': 32953,
```

'03': 162,
'slow': 33131,
'81': 3463,
'02': 140,
'seems': 32409,
'homogeneous': 18095,
'cases': 8282,
'moving': 24003,
'similarly': 32956,
'commonly': 9692,
'detect': 11877,
'associated': 6130,
'differences': 12058,
'environment': 13792,
'43': 2452,
'44': 2483,
'performed': 27200,
'linearly': 21771,
'polarized': 28071,
'pulsed': 29241,
'laser': 21357,
'excitation': 14245,
'splitting': 33536,
'polarization': 28069,
'2a': 1879,
'labeled': 21248,
'frequently': 15562,
'exponential': 14400,
'components': 9795,
'fluorophore': 15305,
'whole': 38047,
'45': 2507,
'46': 2532,
'decay': 11399,
's2a': 31772,
'always': 5110,
'exhibited': 14283,
'ns': 25481,
'less': 21534,

```
'atto': 6280,  
'488': 2598,  
'timescale': 35542,  
'attribute': 6293,  
'free': 15540,  
'rotation': 31440,  
'slower': 33133,  
'schematic': 32214,  
'fits': 15180,  
'values': 37406,  
'79': 3401,  
'mobilities': 23746,  
'along': 5064,  
'neighboring': 24826,  
'maximize': 22789,  
'precision': 28410,  
'measurement': 22917,  
'data': 11260,  
'plotted': 27959,  
'ratio': 30229,  
'trans': 35843,  
'3a': 2292,  
'transition': 35940,  
'increases': 19224,  
'averaged': 6421,  
'rot': 31433,  
'increase': 19222,  
'increasing': 19225,  
'3b': 2297,  
'conversely': 10191,  
'mutants': 24252,  
'constant': 10043,  
'entire': 13770,  
'divided': 12434,  
'location': 21935,  
'reducing': 30516,  
'sometimes': 33302,  
'assess': 6106,  
'47': 2554,
```


'scaling': 32165,
'degree': 11497,
'defined': 11467,
'paradox': 26706,
'assessment': 6112,
'state': 33737,
'made': 22443,
'brightness': 7489,
'analyses': 5258,
'stoichiometric': 33865,
'determined': 11898,
'stoichiometries': 33868,
'present': 28545,
'sample': 32041,
'overall': 26207,
'measure': 22914,
'absolute': 4215,
'labels': 21252,
'type': 36394,
'oligomer': 25813,
'stoichiometry': 33869,
'total': 35744,
'internal': 19662,
'reflection': 30560,
'analyzing': 5288,
'4a': 2631,
'illustrates': 18903,
'representative': 30864,
'monomers': 23873,
'90': 3650,
'demonstrating': 11624,
'smallest': 33165,
'numbers': 25577,
'negligible': 24813,
'compared': 9722,
'4b': 2636,
'confirm': 9923,
'existence': 14290,
'skewed': 33060,

'toward': 35756,
'ruled': 31575,
'showing': 32833,
'per': 27178,
'000': 1,
'analyzed': 5282,
'accurately': 4324,
'quantify': 29546,
'48': 2583,
'discussion': 12265,
's3': 31782,
'4c': 2639,
'samples': 32045,
'yields': 38615,
'factor': 14663,
'b2': 6506,
'mixture': 23643,
'see': 32397,
'self': 32456,
'quenching': 29579,
'homo': 18080,
'system': 34562,
's4': 31821,
'directly': 12188,
'enables': 13593,
'quantification': 29542,
'monomer': 23871,
'n1': 24407,
'n2': 24435,
'129': 634,
'16': 912,
'giving': 16519,
'containing': 10091,
'returns': 31115,
'composition': 9801,
'b1': 6503,
'purely': 29263,
'fidelity': 14990,
'method': 23225,

'ab': 4102,
's5': 31864,
'4d': 2640,
'shows': 32836,
'fitted': 15181,
'assigning': 6118,
'weight': 37968,
'eq': 13889,
's6': 31887,
'assuming': 6140,
'population': 28184,
'dimeric': 12128,
'agrees': 4778,
'elution': 13529,
'profiles': 28768,
'analytical': 5277,
'gel': 16235,
'filtration': 15118,
'chromatography': 9147,
'50': 2690,
'500': 2691,
'even': 14140,
'concentrations': 9848,
'exceed': 14216,
'equivalents': 13914,
'maximal': 22787,
'merely': 23124,
'local': 21919,
'concentration': 9847,
'equilibrium': 13904,
'dissociation': 12364,
'dependence': 11694,
'ability': 4168,
'resolve': 30976,
'external': 14448,
'file': 15104,
'holds': 18051,
'picture': 27701,
'illustration': 18905,

```
'etc': 14068,  
'object': 25631,  
'name': 24616,  
'pnas': 28035,  
'jpg': 20252,  
'characterization': 8940,  
'found': 15459,  
'universally': 36847,  
'throughout': 35469,  
'titration': 35578,  
'upper': 36984,  
'also': 5078,  
'quantifies': 29545,  
'points': 28060,  
'collected': 9588,  
'determine': 11897,  
'dependency': 11697,  
'top': 35702,  
'let': 21540,  
'xm': 38311,  
'represent': 30861,  
'given': 16517,  
'fraction': 15497,  
'simple': 32958,  
'reaction': 30300,  
'obtain': 25662,  
'terms': 35139,  
'expressed': 14416,  
'rearranges': 30351,  
'image': 18915,  
'021': 143,  
'105': 357,  
'significantly': 32928,  
'805': 3455,  
'135': 697,  
'demonstrate': 11621,  
'effect': 13301,  
'fine': 15133,  
'weak': 37931,
```

's7': 31916,
'suggested': 34194,
'polar': 28067,
'iii': 18835,
'comprising': 9815,
'interact': 19583,
'phosphatidylserine': 27532,
'ps': 29055,
'lead': 21431,
'stable': 33664,
'induce': 19302,
'phase': 27417,
'separation': 32523,
'affected': 4671,
's8': 31942,
'phosphatidylinositol': 27531,
'bisphosphate': 7129,
'pip2': 27771,
'replacing': 30814,
'egg': 13344,
'phosphatidylcholine': 27525,
'pc': 26954,
'phosphocholine': 27543,
'affect': 4670,
'studied': 34002,
'purification': 29266,
'epitope': 13877,
'tags': 34839,
'recombinant': 30426,
'extension': 14440,
'49': 2605,
'51': 2729,
'antibody': 5488,
'staining': 33683,
'52': 2755,
'generally': 16285,
'little': 21845,
'impact': 19067,
'biological': 7039,

'53': 2783,
'55': 2846,
'find': 15126,
'tag': 34834,
'slightly': 33119,
'shifts': 32788,
'344': 2141,
'without': 38126,
'changing': 8915,
'qualitative': 29532,
'three': 35446,
'ways': 37919,
'affinities': 4679,
'reduction': 30518,
'degrees': 11499,
'freedom': 15543,
'amounts': 5190,
'restriction': 31036,
'rearrangement': 30349,
'create': 10479,
'first': 15165,
'second': 32366,
'examined': 14209,
'calculating': 7973,
'differing': 12072,
'entropy': 13780,
'bound': 7331,
'56': 2876,
's9': 31960,
'accounting': 4302,
'alone': 5063,
'translation': 35952,
'owing': 26269,
'account': 4299,
'significant': 32927,
'constraints': 10065,
'relatively': 30687,
'reported': 30834,
'vary': 37459,

'tens': 35103,
'hundreds': 18331,
'34': 2133,
'square': 33609,
'micrometer': 23403,
'additionally': 4499,
'partially': 26772,
'exchanging': 14239,
'nano': 24623,
'nm': 25144,
'diameter': 11999,
'physiological': 27666,
'clearly': 9326,
'candidate': 8108,
'participate': 26775,
'observation': 25644,
'tyrosine': 36477,
'64': 3098,
'alanine': 4916,
'indicates': 19277,
'located': 21932,
'undergoes': 36727,
'recent': 30394,
'simulation': 32973,
'close': 9384,
'predictions': 28437,
'favor': 14792,
'influence': 19369,
'distance': 12372,
'modeled': 23760,
'simulations': 32974,
'revealing': 31123,
'might': 23455,
'transmit': 35966,
'information': 19379,
'routes': 31463,
'full': 15643,
'length': 21499,
'energetics': 13680,

'insufficient': 19537,
'drive': 12697,
'configuration': 9917,
'required': 30904,
'enzyme': 13805,
'apart': 5582,
'almost': 5062,
'exposed': 14407,
'interfaces': 19630,
'57': 2895,
'residue': 30955,
'58': 2918,
'dominant': 12564,
'negative': 24808,
'gl2v': 15764,
'a59t': 3994,
'oncogenicity': 25873,
'59': 2937,
'flexibility': 15233,
'allowing': 5054,
'engage': 13689,
'corollary': 10286,
'allostery': 5051,
'couple': 10404,
'nucleotides': 25563,
'antibodies': 5487,
'gives': 16518,
'purified': 29268,
'previously': 28598,
'histidine': 17934,
'affinity': 4680,
'remaining': 30735,
'removed': 30756,
'tobacco': 35649,
'etch': 14069,
'virus': 37660,
'protease': 28932,
'cleavage': 9328,
'indistinguishable': 19290,

'wt': 38223,
'60': 3019,
'biophysical': 7064,
'studies': 34003,
'nonhydrolyzable': 25256,
'analog': 5244,
'purchased': 29257,
'jena': 20158,
'bioscience': 7079,
'anti': 5477,
'pan': 26661,
'igg': 18797,
'emd': 13552,
'millipore': 23491,
'home': 18064,
'built': 7631,
'apparatus': 5618,
'integrated': 19553,
'nikon': 25084,
'inverted': 19808,
'microscope': 23420,
'based': 6668,
'previous': 28596,
'design': 11822,
'61': 3052,
'calculated': 7971,
'www': 38242,
'com': 9640,
'real': 30328,
'igor': 18814,
'pro': 28672,
'software': 33252,
'wavemetrics': 37914,
'theoretical': 35290,
'describing': 11810,
'recorded': 30460,
'correlated': 10304,
'card': 8210,
'counts': 10403,

```
'later': 21372,  
'package': 26578,  
'laboratory': 21259,  
'university': 36855,  
'illinois': 18892,  
'eclipse': 13235,  
'ti': 35496,  
'confocal': 9935,  
'emission': 13562,  
'separated': 32519,  
'imaging': 18924,  
'100': 289,  
'oil': 25776,  
'immersion': 18955,  
'objective': 25632,  
'camera': 8036,  
'technology': 35049,  
'561': 2880,  
'coherent': 9514,  
'illumination': 18900,  
'sources': 33348,  
'strongest': 33979,  
'power': 28300,  
'setting': 32662,  
'dark': 11248,  
'background': 6548,  
'ten': 35093,  
'seconds': 32372,  
'images': 18920,  
'acquired': 4382,  
'exposure': 14410,  
'ms': 24044,  
'quantified': 29544,  
'custom': 10761,  
...}
```

```
In [44]: # sorted features by idf_ values  
  
from collections import OrderedDict
```

```
sorted_by_value = OrderedDict(sorted(dictionary.items(), reverse= True,
key=lambda x: x[1]))
sorted_by_value
```

```
Out[44]: OrderedDict([('000011', 6.869767612130462),
                        ('000012', 6.869767612130462),
                        ('00034', 6.869767612130462),
                        ('00092', 6.869767612130462),
                        ('001635', 6.869767612130462),
                        ('001810', 6.869767612130462),
                        ('002880', 6.869767612130462),
                        ('0030485', 6.869767612130462),
                        ('0037', 6.869767612130462),
                        ('00433', 6.869767612130462),
                        ('0045', 6.869767612130462),
                        ('005188', 6.869767612130462),
                        ('005631', 6.869767612130462),
                        ('0059', 6.869767612130462),
                        ('0072', 6.869767612130462),
                        ('0086c', 6.869767612130462),
                        ('0088', 6.869767612130462),
                        ('0117', 6.869767612130462),
                        ('01228', 6.869767612130462),
                        ('0134', 6.869767612130462),
                        ('0169', 6.869767612130462),
                        ('02114', 6.869767612130462),
                        ('0222', 6.869767612130462),
                        ('02b', 6.869767612130462),
                        ('0314', 6.869767612130462),
                        ('0405', 6.869767612130462),
                        ('045a', 6.869767612130462),
                        ('0615b', 6.869767612130462),
                        ('07ee', 6.869767612130462),
                        ('083', 6.869767612130462),
                        ('0896', 6.869767612130462),
                        ('0917', 6.869767612130462),
                        ('0996', 6.869767612130462),
                        ('0b', 6.869767612130462),
                        ('0x', 6.869767612130462),
                        ('100kb', 6.869767612130462),
```

('1025', 6.869767612130462),
('1067', 6.869767612130462),
('1080', 6.869767612130462),
('10838', 6.869767612130462),
('1095', 6.869767612130462),
('10q21', 6.869767612130462),
('10t', 6.869767612130462),
('1103', 6.869767612130462),
('1134', 6.869767612130462),
('1139t', 6.869767612130462),
('1183l', 6.869767612130462),
('118k', 6.869767612130462),
('118mg', 6.869767612130462),
('118p', 6.869767612130462),
('1195', 6.869767612130462),
('11n', 6.869767612130462),
('11p13', 6.869767612130462),
('11th', 6.869767612130462),
('11val555', 6.869767612130462),
('11val560asp', 6.869767612130462),
('1212', 6.869767612130462),
('1218l', 6.869767612130462),
('1232', 6.869767612130462),
('123k', 6.869767612130462),
('1244', 6.869767612130462),
('1247', 6.869767612130462),
('1249c', 6.869767612130462),
('125b', 6.869767612130462),
('127k', 6.869767612130462),
('12911', 6.869767612130462),
('12h', 6.869767612130462),
('12v', 6.869767612130462),
('1303', 6.869767612130462),
('131g', 6.869767612130462),
('1350', 6.869767612130462),
('1354', 6.869767612130462),
('1370', 6.869767612130462),
('1373', 6.869767612130462),
('1377', 6.869767612130462),

('1381c', 6.869767612130462),
('1386', 6.869767612130462),
('1393', 6.869767612130462),
('139k', 6.869767612130462),
('13t', 6.869767612130462),
('13val654ala', 6.869767612130462),
('1406gright', 6.869767612130462),
('1414', 6.869767612130462),
('1421', 6.869767612130462),
('1431', 6.869767612130462),
('1440', 6.869767612130462),
('1455gright', 6.869767612130462),
('147171', 6.869767612130462),
('1480', 6.869767612130462),
('1484', 6.869767612130462),
('1492', 6.869767612130462),
('1495aright', 6.869767612130462),
('14department', 6.869767612130462),
('14q21', 6.869767612130462),
('14thr670ile', 6.869767612130462),
('1501gright', 6.869767612130462),
('1502aright', 6.869767612130462),
('1509', 6.869767612130462),
('150v', 6.869767612130462),
('1533', 6.869767612130462),
('153k', 6.869767612130462),
('154838u22', 6.869767612130462),
('154899u20', 6.869767612130462),
('155273l23', 6.869767612130462),
('155348l22', 6.869767612130462),
('1554', 6.869767612130462),
('155750u20', 6.869767612130462),
('156027l20', 6.869767612130462),
('156107l22', 6.869767612130462),
('1566', 6.869767612130462),
('1580', 6.869767612130462),
('1581', 6.869767612130462),
('15nh4', 6.869767612130462),
('15p', 6.869767612130462),

('15t', 6.869767612130462),
('15x', 6.869767612130462),
('1608', 6.869767612130462),
('162072007', 6.869767612130462),
('1621', 6.869767612130462),
('1634', 6.869767612130462),
('1636', 6.869767612130462),
('164k', 6.869767612130462),
('1658', 6.869767612130462),
('1677', 6.869767612130462),
('1696', 6.869767612130462),
('16p11', 6.869767612130462),
('1704deltt', 6.869767612130462),
('172656u22', 6.869767612130462),
('172747u19', 6.869767612130462),
('173076l19', 6.869767612130462),
('173160l22', 6.869767612130462),
('174laright', 6.869767612130462),
('1753', 6.869767612130462),
('1762', 6.869767612130462),
('1763', 6.869767612130462),
('1769', 6.869767612130462),
('1780g', 6.869767612130462),
('178gright', 6.869767612130462),
('1790', 6.869767612130462),
('1790ct', 6.869767612130462),
('1794', 6.869767612130462),
('1799gt', 6.869767612130462),
('17asn822lys', 6.869767612130462),
('17asp816val', 6.869767612130462),
('17asp820gly', 6.869767612130462),
('17n', 6.869767612130462),
('17r', 6.869767612130462),
('1800g', 6.869767612130462),
('1801a', 6.869767612130462),
('182a', 6.869767612130462),
('1833', 6.869767612130462),
('1837', 6.869767612130462),
('184352', 6.869767612130462),

```
('1860', 6.869767612130462),
('188k', 6.869767612130462),
('18ex', 6.869767612130462),
('18k', 6.869767612130462),
('18mm', 6.869767612130462),
('18mutantgenes', 6.869767612130462),
('190b', 6.869767612130462),
('1957', 6.869767612130462),
('1961', 6.869767612130462),
('1968', 6.869767612130462),
('1990a', 6.869767612130462),
('1990b', 6.869767612130462),
('1993b', 6.869767612130462),
('19ex', 6.869767612130462),
('19n', 6.869767612130462),
('1a4', 6.869767612130462),
('1c1c', 6.869767612130462),
('1d3', 6.869767612130462),
('1department', 6.869767612130462),
('1dscan', 6.869767612130462),
('1hn', 6.869767612130462),
('1lr3', 6.869767612130462),
('1mg', 6.869767612130462),
('1months', 6.869767612130462),
('1o', 6.869767612130462),
('1q31', 6.869767612130462),
('1r2d', 6.869767612130462),
('1tsr', 6.869767612130462),
('2016cancer', 6.869767612130462),
('2022', 6.869767612130462),
('2023', 6.869767612130462),
('2032', 6.869767612130462),
('2048', 6.869767612130462),
('2053', 6.869767612130462),
('209k', 6.869767612130462),
('20mb', 6.869767612130462),
('2105', 6.869767612130462),
('2111c3a', 6.869767612130462),
('2150c', 6.869767612130462),
```

('2161', 6.869767612130462),
('21b', 6.869767612130462),
('21ex', 6.869767612130462),
('21n', 6.869767612130462),
('2209', 6.869767612130462),
('2240', 6.869767612130462),
('2247', 6.869767612130462),
('224k', 6.869767612130462),
('2254', 6.869767612130462),
('2256', 6.869767612130462),
('2264', 6.869767612130462),
('227650', 6.869767612130462),
('22q34', 6.869767612130462),
('2316', 6.869767612130462),
('2328', 6.869767612130462),
('2392', 6.869767612130462),
('23948', 6.869767612130462),
('23t', 6.869767612130462),
('2413', 6.869767612130462),
('2430', 6.869767612130462),
('24k', 6.869767612130462),
('2562', 6.869767612130462),
('2578', 6.869767612130462),
('2628', 6.869767612130462),
('2666', 6.869767612130462),
('27b', 6.869767612130462),
('27months', 6.869767612130462),
('2800', 6.869767612130462),
('283c', 6.869767612130462),
('2868', 6.869767612130462),
('2870', 6.869767612130462),
('2878', 6.869767612130462),
('291x', 6.869767612130462),
('292a', 6.869767612130462),
('294k', 6.869767612130462),
('2982', 6.869767612130462),
('2997', 6.869767612130462),
('2h10', 6.869767612130462),
('2ito', 6.869767612130462),

('2itp', 6.869767612130462),
('2itt', 6.869767612130462),
('2ity', 6.869767612130462),
('2itz', 6.869767612130462),
('2jiu', 6.869767612130462),
('2months', 6.869767612130462),
('2p12', 6.869767612130462),
('2q12', 6.869767612130462),
('2r1', 6.869767612130462),
('2t202', 6.869767612130462),
('2tip', 6.869767612130462),
('2v5y', 6.869767612130462),
('2xb7', 6.869767612130462),
('3011s', 6.869767612130462),
('3027', 6.869767612130462),
('3031', 6.869767612130462),
('307k', 6.869767612130462),
('30k', 6.869767612130462),
('30th', 6.869767612130462),
('314k', 6.869767612130462),
('3217', 6.869767612130462),
('3229', 6.869767612130462),
('3247', 6.869767612130462),
('328k', 6.869767612130462),
('32ds', 6.869767612130462),
('32pi', 6.869767612130462),
('3340', 6.869767612130462),
('3341', 6.869767612130462),
('3350', 6.869767612130462),
('3493', 6.869767612130462),
('350031', 6.869767612130462),
('3506', 6.869767612130462),
('3522', 6.869767612130462),
('3555', 6.869767612130462),
('35th', 6.869767612130462),
('36aa', 6.869767612130462),
('36b4', 6.869767612130462),
('370c', 6.869767612130462),
('3719', 6.869767612130462),

```
('3743', 6.869767612130462),
('3752', 6.869767612130462),
('3762', 6.869767612130462),
('3770', 6.869767612130462),
('3787', 6.869767612130462),
('3c5', 6.869767612130462),
('3crv', 6.869767612130462),
('3diaminobenzidine', 6.869767612130462),
('3eyh', 6.869767612130462),
('3lcs', 6.869767612130462),
('4001', 6.869767612130462),
('40atg', 6.869767612130462),
('4100', 6.869767612130462),
('4129', 6.869767612130462),
('4161', 6.869767612130462),
('4185', 6.869767612130462),
('4187', 6.869767612130462),
('42193', 6.869767612130462),
('42236', 6.869767612130462),
('42251', 6.869767612130462),
('42272', 6.869767612130462),
('42290', 6.869767612130462),
('42302', 6.869767612130462),
('4236', 6.869767612130462),
('4292', 6.869767612130462),
('4363', 6.869767612130462),
('4366', 6.869767612130462),
('440k', 6.869767612130462),
('4490', 6.869767612130462),
('45122', 6.869767612130462),
('4549', 6.869767612130462),
('458aright', 6.869767612130462),
('46285', 6.869767612130462),
('4686', 6.869767612130462),
('4750', 6.869767612130462),
('490nm', 6.869767612130462),
('4982', 6.869767612130462),
('49atg', 6.869767612130462),
('4a15', 6.869767612130462),
```

('4g0n', 6.869767612130462),
('4q32', 6.869767612130462),
('4v', 6.869767612130462),
('50jc', 6.869767612130462),
('50mm3', 6.869767612130462),
('5161', 6.869767612130462),
('5174', 6.869767612130462),
('5241', 6.869767612130462),
('5261', 6.869767612130462),
('53bp143', 6.869767612130462),
('53bp244', 6.869767612130462),
('53k', 6.869767612130462),
('54atg', 6.869767612130462),
('54egfr', 6.869767612130462),
('5506', 6.869767612130462),
('5558', 6.869767612130462),
('5559c', 6.869767612130462),
('5559insa', 6.869767612130462),
('55634u24', 6.869767612130462),
('558del', 6.869767612130462),
('55b', 6.869767612130462),
('55mm', 6.869767612130462),
('5611', 6.869767612130462),
('5632', 6.869767612130462),
('5721', 6.869767612130462),
('576del', 6.869767612130462),
('578del', 6.869767612130462),
('579del', 6.869767612130462),
('59l', 6.869767612130462),
('5c9c', 6.869767612130462),
('5kb', 6.869767612130462),
('5mg', 6.869767612130462),
('5p12', 6.869767612130462),
('5q23', 6.869767612130462),
('5s', 6.869767612130462),
('5y', 6.869767612130462),
('600x', 6.869767612130462),
('601399', 6.869767612130462),
('604850', 6.869767612130462),

```
('60min', 6.869767612130462),
('6100', 6.869767612130462),
('610756', 6.869767612130462),
('6174delt', 6.869767612130462),
('621', 6.869767612130462),
('63k', 6.869767612130462),
('643c', 6.869767612130462),
('6652', 6.869767612130462),
('6718', 6.869767612130462),
('6783', 6.869767612130462),
('6814', 6.869767612130462),
('6917', 6.869767612130462),
('6a6a', 6.869767612130462),
('6b6b', 6.869767612130462),
('6figure', 6.869767612130462),
('6m', 6.869767612130462),
('7001', 6.869767612130462),
('7025', 6.869767612130462),
('7066', 6.869767612130462),
('711', 6.869767612130462),
('7176', 6.869767612130462),
('7215', 6.869767612130462),
('736gright', 6.869767612130462),
('73749', 6.869767612130462),
('738', 6.869767612130462),
('7383', 6.869767612130462),
('767eeyl770', 6.869767612130462),
('76k', 6.869767612130462),
('770aright', 6.869767612130462),
('773insertion', 6.869767612130462),
('77k', 6.869767612130462),
('78k', 6.869767612130462),
('7985', 6.869767612130462),
('7q11', 6.869767612130462),
('7q36', 6.869767612130462),
('7tm', 6.869767612130462),
('8002', 6.869767612130462),
('800cw', 6.869767612130462),
('81e', 6.869767612130462),
```

('8312', 6.869767612130462),
('840gs', 6.869767612130462),
('8593', 6.869767612130462),
('8644', 6.869767612130462),
('8801', 6.869767612130462),
('89991', 6.869767612130462),
('8g7g3', 6.869767612130462),
('8mgb', 6.869767612130462),
('8p22', 6.869767612130462),
('8p23', 6.869767612130462),
('8q11', 6.869767612130462),
('8x', 6.869767612130462),
('9138', 6.869767612130462),
('9171', 6.869767612130462),
('9275s', 6.869767612130462),
('9408', 6.869767612130462),
('9451s', 6.869767612130462),
('9459', 6.869767612130462),
('95051', 6.869767612130462),
('9ala502', 6.869767612130462),
('9months', 6.869767612130462),
('9p22', 6.869767612130462),
('9t', 6.869767612130462),
('a00n', 6.869767612130462),
('a127s', 6.869767612130462),
('a138v', 6.869767612130462),
('a1a', 6.869767612130462),
('a1b', 6.869767612130462),
('a2220', 6.869767612130462),
('a26v', 6.869767612130462),
('a2717s', 6.869767612130462),
('a289', 6.869767612130462),
('a289d', 6.869767612130462),
('a289t', 6.869767612130462),
('a2a', 6.869767612130462),
('a2bp1', 6.869767612130462),
('a42', 6.869767612130462),
('a428t', 6.869767612130462),
('a456p', 6.869767612130462),

```

('a489', 6.869767612130462),
('a491ins', 6.869767612130462),
('a496s', 6.869767612130462),
('a636p', 6.869767612130462),
('a66', 6.869767612130462),
('a707t', 6.869767612130462),
('a750insp', 6.869767612130462),
('a95d', 6.869767612130462),
('a96g', 6.869767612130462),
('aataatcagtggtgattcgtggag', 6.869767612130462),
('aacccccac', 6.869767612130462),
('aagccgaagggctcccactg', 6.869767612130462),
('aattctgggtggaggcaccggtggcaatccgagtgggtgac', 6.86976761213
0462),
('aav', 6.869767612130462),
('ab134175', 6.869767612130462),
('ab137537', 6.869767612130462),
('ab5443', 6.869767612130462),
('ab5452', 6.869767612130462),
('ab65258', 6.869767612130462),
('ab72999', 6.869767612130462),
('abate', 6.869767612130462),
('abcb7', 6.869767612130462),
('abdurashidova', 6.869767612130462),
('aberle', 6.869767612130462),
('aberrancies', 6.869767612130462),
('abi3130xl', 6.869767612130462),
('abiprism', 6.869767612130462),
('ablanck', 6.869767612130462),
('abovementioned', 6.869767612130462),
('abraham', 6.869767612130462),
('abridged', 6.869767612130462),
('abscess', 6.869767612130462),
('abyss', 6.869767612130462),
('acagacuucggaguaccug', 6.869767612130462),
('acaggcctcgcagggtgtag', 6.869767612130462),
('accaataacctattccgttacac', 6.869767612130462),
('accacagtccatgccatcac', 6.869767612130462),
('accccatgaactgcctgtc', 6.869767612130462),

```

('accommodation', 6.869767612130462),
('accorded', 6.869767612130462),
('accruing', 6.869767612130462),
('accumed', 6.869767612130462),
('accumulation44', 6.869767612130462),
('acetylase', 6.869767612130462),
('acetylatabe', 6.869767612130462),
('ach4', 6.869767612130462),
('acidophilum', 6.869767612130462),
('acii', 6.869767612130462),
('acth', 6.869767612130462),
('activatable', 6.869767612130462),
('activities6', 6.869767612130462),
('activity1', 6.869767612130462),
('activity16', 6.869767612130462),
('activity22', 6.869767612130462),
('activity9', 6.869767612130462),
('activityb', 6.869767612130462),
('actl6a', 6.869767612130462),
('acttcacagccctgcgtaaac', 6.869767612130462),
('acylamide', 6.869767612130462),
('adaptively', 6.869767612130462),
('adata', 6.869767612130462),
('additivity', 6.869767612130462),
('adenosines', 6.869767612130462),
('adenosylhomocysteine', 6.869767612130462),
('adjunctive', 6.869767612130462),
('adjuvants', 6.869767612130462),
('adomet', 6.869767612130462),
('advantage1', 6.869767612130462),
('advisable', 6.869767612130462),
('aee778', 6.869767612130462),
('af6', 6.869767612130462),
('affiniti', 6.869767612130462),
('affy', 6.869767612130462),
('afro', 6.869767612130462),
('agag', 6.869767612130462),
('agami', 6.869767612130462),
('agcatctacttccaggcaccagcaagccaatggagag', 6.869767612130

462),

('agcggatcccccggtccaaaatgccc', 6.869767612130462),
('agctcaacgtggacg', 6.869767612130462),
('agggacacacgatgggcttctg', 6.869767612130462),
('aggresomes', 6.869767612130462),
('agk', 6.869767612130462),
('agnosis', 6.869767612130462),
('agtcggaattcactgatactgctgggta', 6.869767612130462),
('agtcggaattcgtagtggctgtggggga', 6.869767612130462),
('agtcggaattctctgcagatagttctacc', 6.869767612130462),
('agvd', 6.869767612130462),
('ahmadie', 6.869767612130462),
('aida', 6.869767612130462),
('airdried', 6.869767612130462),
('airspace', 6.869767612130462),
('akap13', 6.869767612130462),
('akiyama', 6.869767612130462),
('akti', 6.869767612130462),
('al17', 6.869767612130462),
('al24', 6.869767612130462),
('al5', 6.869767612130462),
('ala1708glu', 6.869767612130462),
('ala489', 6.869767612130462),
('ala530', 6.869767612130462),
('ala530thr', 6.869767612130462),
('ala530val', 6.869767612130462),
('ala750', 6.869767612130462),
('ala763', 6.869767612130462),
('ala767insaia', 6.869767612130462),
('ala767insalaile', 6.869767612130462),
('ala775instyrvalmetalala', 6.869767612130462),
('aldolase', 6.869767612130462),
('alexafuor', 6.869767612130462),
('aliases', 6.869767612130462),
('aliex', 6.869767612130462),
('alimentary', 6.869767612130462),
('alizadeh', 6.869767612130462),
('alkali', 6.869767612130462),
('alky1604', 6.869767612130462),

('allele31', 6.869767612130462),
('allelism', 6.869767612130462),
('allis', 6.869767612130462),
('allostery', 6.869767612130462),
('almeida', 6.869767612130462),
('alpha5', 6.869767612130462),
('alphac', 6.869767612130462),
('alternation', 6.869767612130462),
('aluminium', 6.869767612130462),
('aml', 6.869767612130462),
('amazing', 6.869767612130462),
('ambiguously', 6.869767612130462),
('amelxy', 6.869767612130462),
('aminobutyric', 6.869767612130462),
('aminopurine', 6.869767612130462),
('aml1b', 6.869767612130462),
('aml2', 6.869767612130462),
('aml3', 6.869767612130462),
('amlcg', 6.869767612130462),
('amplifica', 6.869767612130462),
('amyloidoses1', 6.869767612130462),
('analysis14', 6.869767612130462),
('analysis9', 6.869767612130462),
('analysisall', 6.869767612130462),
('analysisithe', 6.869767612130462),
('analyzable', 6.869767612130462),
('anastomosis', 6.869767612130462),
('anastomotic', 6.869767612130462),
('ance', 6.869767612130462),
('and15', 6.869767612130462),
('and5a', 6.869767612130462),
('andbglii', 6.869767612130462),
('andreas', 6.869767612130462),
('androgendb', 6.869767612130462),
('anf', 6.869767612130462),
('ang', 6.869767612130462),
('angiomatoid', 6.869767612130462),
('angiomyolipomas', 6.869767612130462),
('angiopoietin', 6.869767612130462),

('ani', 6.869767612130462),
('animals29', 6.869767612130462),
('anionic', 6.869767612130462),
('anisomycin', 6.869767612130462),
('ankyrins', 6.869767612130462),
('annex', 6.869767612130462),
('anovas', 6.869767612130462),
('ans', 6.869767612130462),
('answering', 6.869767612130462),
('anteriorly', 6.869767612130462),
('antiangiogenetic', 6.869767612130462),
('anticoagulant', 6.869767612130462),
('anticorrelation', 6.869767612130462),
('antierk', 6.869767612130462),
('antihormone', 6.869767612130462),
('antimitogenic', 6.869767612130462),
('antioncogene', 6.869767612130462),
('antiphosphoserine', 6.869767612130462),
('antiphosphothreonine', 6.869767612130462),
('antireceptor', 6.869767612130462),
('antitubulin', 6.869767612130462),
('antitumorigenic', 6.869767612130462),
('aom', 6.869767612130462),
('aparicio', 6.869767612130462),
('apex', 6.869767612130462),
('apob', 6.869767612130462),
('apoptosis2', 6.869767612130462),
('apoptosis23', 6.869767612130462),
('apoptosis8', 6.869767612130462),
('apperley', 6.869767612130462),
('apposing', 6.869767612130462),
('appraisal', 6.869767612130462),
('approaches20', 6.869767612130462),
('approvals', 6.869767612130462),
('approx3', 6.869767612130462),
('approx35', 6.869767612130462),
('approx4', 6.869767612130462),
('approx45', 6.869767612130462),
('approx60', 6.869767612130462),

```
('approximations', 6.869767612130462),
('apt', 6.869767612130462),
('arabinoside', 6.869767612130462),
('arches', 6.869767612130462),
('arg112his', 6.869767612130462),
('arg506', 6.869767612130462),
('arg509', 6.869767612130462),
('arg601gln', 6.869767612130462),
('arg631cys', 6.869767612130462),
('arg683trp', 6.869767612130462),
('argiris', 6.869767612130462),
('argonne', 6.869767612130462),
('aricescu', 6.869767612130462),
('arntta', 6.869767612130462),
('arrayscan', 6.869767612130462),
('arrest21', 6.869767612130462),
('ars853', 6.869767612130462),
('as604850', 6.869767612130462),
('asc', 6.869767612130462),
('ascribes', 6.869767612130462),
('asn486', 6.869767612130462),
('asn770', 6.869767612130462),
('asn771', 6.869767612130462),
('asn771insalaprotpr', 6.869767612130462),
('asn771insasn', 6.869767612130462),
('asn771insasnprogly', 6.869767612130462),
('asn771insasnprohis', 6.869767612130462),
('asn771insasp', 6.869767612130462),
('asn771insaspgly', 6.869767612130462),
('asn771insgly', 6.869767612130462),
('asn771insglyleu', 6.869767612130462),
('asn771linmetalathrpro', 6.869767612130462),
('asn771linsservalasp', 6.869767612130462),
('asn771linsservalgln', 6.869767612130462),
('asn771linsservalpro', 6.869767612130462),
('asns', 6.869767612130462),
('asp153', 6.869767612130462),
('asp154', 6.869767612130462),
('asp158', 6.869767612130462),
```

```
('asp186', 6.869767612130462),
('asp214', 6.869767612130462),
('asp312asn', 6.869767612130462),
('asp423asn', 6.869767612130462),
('asp449', 6.869767612130462),
('asp513tyr', 6.869767612130462),
('asp536del', 6.869767612130462),
('asp539tyr', 6.869767612130462),
('asp761', 6.869767612130462),
('asp770', 6.869767612130462),
('asp770dupservalasp', 6.869767612130462),
('asp770insalaser', 6.869767612130462),
('asp770insaspasval', 6.869767612130462),
('asp770inscysval', 6.869767612130462),
('asp770insglyserval', 6.869767612130462),
('asp770insglyval', 6.869767612130462),
('asp770insglyvalval', 6.869767612130462),
('asp770insmetalaservalasp', 6.869767612130462),
('asp820ala', 6.869767612130462),
('asp835ala', 6.869767612130462),
('asp835asn', 6.869767612130462),
('asp835deletion', 6.869767612130462),
('asp835gly', 6.869767612130462),
('asp835his', 6.869767612130462),
('asp835val', 6.869767612130462),
('aspp2', 6.869767612130462),
('assay19', 6.869767612130462),
('assay40', 6.869767612130462),
('assent', 6.869767612130462),
('associatedwith', 6.869767612130462),
('asynchronized', 6.869767612130462),
('atcacagtgatgctg3', 6.869767612130462),
('atf1', 6.869767612130462),
('atgaattctagaggaaca', 6.869767612130462),
('atggagtggcatagggcatggg', 6.869767612130462),
('atgggacaggcactgatttgt', 6.869767612130462),
('atgtctgtctcattcaatgg', 6.869767612130462),
('atinine', 6.869767612130462),
('attattgagtgccttttc', 6.869767612130462),
```

('attendant', 6.869767612130462),
('attest', 6.869767612130462),
('attl', 6.869767612130462),
('attp', 6.869767612130462),
('audi', 6.869767612130462),
('aura', 6.869767612130462),
('auspices', 6.869767612130462),
('autoclave', 6.869767612130462),
('autonomy', 6.869767612130462),
('autophagy9', 6.869767612130462),
('autopsied', 6.869767612130462),
('auxotrophy', 6.869767612130462),
('avanti', 6.869767612130462),
('avert', 6.869767612130462),
('avizienyte', 6.869767612130462),
('avraham', 6.869767612130462),
('axel', 6.869767612130462),
('ay803272', 6.869767612130462),
('aylesbury', 6.869767612130462),
('azd9291', 6.869767612130462),
('b12', 6.869767612130462),
('b15', 6.869767612130462),
('baca', 6.869767612130462),
('baf73611', 6.869767612130462),
('balancing', 6.869767612130462),
('bams', 6.869767612130462),
('bani', 6.869767612130462),
('banin', 6.869767612130462),
('barber', 6.869767612130462),
('bard', 6.869767612130462),
('barletta', 6.869767612130462),
('bartkova', 6.869767612130462),
('barton', 6.869767612130462),
('basso', 6.869767612130462),
('bax48', 6.869767612130462),
('bbrc', 6.869767612130462),
('bcc1', 6.869767612130462),
('bcg', 6.869767612130462),
('bcl7', 6.869767612130462),

('beaming', 6.869767612130462),
('bebenek', 6.869767612130462),
('beds', 6.869767612130462),
('beerm', 6.869767612130462),
('belarussian', 6.869767612130462),
('bellostas', 6.869767612130462),
('benzoyl', 6.869767612130462),
('berap4', 6.869767612130462),
('bern', 6.869767612130462),
('bernoulli', 6.869767612130462),
('bernstein', 6.869767612130462),
('berry', 6.869767612130462),
('bfl1', 6.869767612130462),
('bgl', 6.869767612130462),
('bhatia', 6.869767612130462),
('bhgvs', 6.869767612130462),
('bicc', 6.869767612130462),
('bifid', 6.869767612130462),
('billaud', 6.869767612130462),
('bilobed', 6.869767612130462),
('bioanalyser', 6.869767612130462),
('biocomputing', 6.869767612130462),
('bioinfo', 6.869767612130462),
('biologist', 6.869767612130462),
('biomedicum', 6.869767612130462),
('bioprime', 6.869767612130462),
('biorobot', 6.869767612130462),
('biosym', 6.869767612130462),
('biphosphate', 6.869767612130462),
('birthday', 6.869767612130462),
('bisagni', 6.869767612130462),
('bitransgenic', 6.869767612130462),
('bl578', 6.869767612130462),
('blacks', 6.869767612130462),
('blackwell', 6.869767612130462),
('blanket', 6.869767612130462),
('blastomeres', 6.869767612130462),
('blc', 6.869767612130462),
('bleed', 6.869767612130462),

('bleomycin', 6.869767612130462),
('blimp', 6.869767612130462),
('blinding', 6.869767612130462),
('bloc', 6.869767612130462),
('bluesky', 6.869767612130462),
('bmax', 6.869767612130462),
('bnip3', 6.869767612130462),
('bodes', 6.869767612130462),
('bodrug', 6.869767612130462),
('bogden', 6.869767612130462),
('bohlander', 6.869767612130462),
('bonneau', 6.869767612130462),
('bootstrapping', 6.869767612130462),
('borohydrate', 6.869767612130462),
('boudewijn', 6.869767612130462),
('boulder', 6.869767612130462),
('boussard', 6.869767612130462),
('bph', 6.869767612130462),
('bracket', 6.869767612130462),
('brafengaged', 6.869767612130462),
('brafmutant', 6.869767612130462),
('branchpoint', 6.869767612130462),
('brb', 6.869767612130462),
('brcald', 6.869767612130462),
('bre', 6.869767612130462),
('breasts', 6.869767612130462),
('brest', 6.869767612130462),
('broke', 6.869767612130462),
('bronchiectasis', 6.869767612130462),
('bronchioid', 6.869767612130462),
('bronchiolar', 6.869767612130462),
('brunet', 6.869767612130462),
('bscher', 6.869767612130462),
('bsiwi', 6.869767612130462),
('bsphi', 6.869767612130462),
('bstxi', 6.869767612130462),
('bthe', 6.869767612130462),
('btl', 6.869767612130462),
('bubrla', 6.869767612130462),

('bucc', 6.869767612130462),
('buds', 6.869767612130462),
('bulb', 6.869767612130462),
('bulletin', 6.869767612130462),
('burlab', 6.869767612130462),
('burgeoning', 6.869767612130462),
('burger', 6.869767612130462),
('burgoyne', 6.869767612130462),
('burgt', 6.869767612130462),
('bursts', 6.869767612130462),
('busulfan', 6.869767612130462),
('buti', 6.869767612130462),
('butyrate', 6.869767612130462),
('buys', 6.869767612130462),
('bx61', 6.869767612130462),
('bxf', 6.869767612130462),
('byron', 6.869767612130462),
('c118s', 6.869767612130462),
('c13', 6.869767612130462),
('c15orf21', 6.869767612130462),
('c19', 6.869767612130462),
('c2060g', 6.869767612130462),
('c228t', 6.869767612130462),
('c242s', 6.869767612130462),
('c250t', 6.869767612130462),
('c26g7', 6.869767612130462),
('c277w', 6.869767612130462),
('c290r', 6.869767612130462),
('c33a', 6.869767612130462),
('c379', 6.869767612130462),
('c379a', 6.869767612130462),
('c381', 6.869767612130462),
('c393', 6.869767612130462),
('c44y', 6.869767612130462),
('c450ins', 6.869767612130462),
('c459g', 6.869767612130462),
('c47f', 6.869767612130462),
('c4g', 6.869767612130462),
('c540ins', 6.869767612130462),


```

('c57b1', 6.869767612130462),
('c609w', 6.869767612130462),
('c609y', 6.869767612130462),
('c61', 6.869767612130462),
('c66', 6.869767612130462),
('c6i8s', 6.869767612130462),
('c706f', 6.869767612130462),
('c712r', 6.869767612130462),
('c775inshv', 6.869767612130462),
('c84', 6.869767612130462),
('c843y', 6.869767612130462),
('ca630', 6.869767612130462),
('caaataccaacaggatcaagttgtcaaagataatattgaagctgtaggg',
6.869767612130462),
('caaccacatcaagtattggtctctctttctcagctggataaaggctctgg',
6.869767612130462),
('caagtattggtctctcgtctttcagctggataaaggctctggtttaatgc',
6.869767612130462),
('caatgggtcttgctcatccatgcagagagcctcccacgccccattg',
6.869767612130462),
('cacaatggattttgccagtctcc', 6.869767612130462),
('cacgtg', 6.869767612130462),
('cactcatgataactcgagcc', 6.869767612130462),
('cactcatgatatgtcgagcc', 6.869767612130462),
('cactcatgatatttcgagcc', 6.869767612130462),
('cactcatgatctctcgagcc', 6.869767612130462),
('cagctctggctcacactaccag', 6.869767612130462),
('cagctggataaaggctctggtttaatgctcttcatacgtttgtcaatttctcgata
c',
6.869767612130462),
462),
('cagtgaggagcccttcggcttcgcggtgacccgctgccat', 6.869767612130
462),
('cagttgatctgtgggcacgttggtgaagg', 6.869767612130462),
('cagttgatctgtgggcctgttggtgaagg', 6.869767612130462),
('cagttgatctgtgggcgtgttggtgaagg', 6.869767612130462),
('cagttgatctgtgggctcgttggtgaagg', 6.869767612130462),
('cagttgatctgtgggcttggttggtgaagg', 6.869767612130462),
('caki', 6.869767612130462),
('calame', 6.869767612130462),

```

('calbo', 6.869767612130462),
('calboplatin', 6.869767612130462),
('calculator', 6.869767612130462),
('calgary', 6.869767612130462),
('calgb', 6.869767612130462),
('caligiuri', 6.869767612130462),
('callers', 6.869767612130462),
('calvert', 6.869767612130462),
('calypso', 6.869767612130462),
('cam1', 6.869767612130462),
('campaign', 6.869767612130462),
('cancelled', 6.869767612130462),
('cancels', 6.869767612130462),
('cancer14', 6.869767612130462),
('cancer24', 6.869767612130462),
('cancer36', 6.869767612130462),
('cancer4', 6.869767612130462),
('cancer6', 6.869767612130462),
('cancers30', 6.869767612130462),
('cancers31', 6.869767612130462),
('canman', 6.869767612130462),
('cannon', 6.869767612130462),
('canonically', 6.869767612130462),
('canto', 6.869767612130462),
('capitulated', 6.869767612130462),
('capricious', 6.869767612130462),
('carbamidomethylation', 6.869767612130462),
('carbobenzoxyl', 6.869767612130462),
('carboxylates', 6.869767612130462),
('carcinoma17', 6.869767612130462),
('cards', 6.869767612130462),
('caretakers', 6.869767612130462),
('carlson', 6.869767612130462),
('carson', 6.869767612130462),
('carvajal', 6.869767612130462),
('cases4', 6.869767612130462),
('cases9', 6.869767612130462),
('casnellie', 6.869767612130462),
('casodex', 6.869767612130462),

```
('casola', 6.869767612130462),
('castilla', 6.869767612130462),
('catalouge', 6.869767612130462),
('catalyst', 6.869767612130462),
('catastrophe', 6.869767612130462),
('catataatctatcatattctcgattttcttgaaactgagtggttatattc',
6.869767612130462),
('categoric', 6.869767612130462),
('cathode', 6.869767612130462),
('cattle', 6.869767612130462),
('cauchi', 6.869767612130462),
('cbb', 6.869767612130462),
('cbfa2', 6.869767612130462),
('cbic', 6.869767612130462),
('cble10r', 6.869767612130462),
('cble7f', 6.869767612130462),
('cblwt', 6.869767612130462),
('cbt', 6.869767612130462),
('ccaccaccagctcctcgctc', 6.869767612130462),
('ccagaccttatccagctgagaaagagagaccaatacttgatgtggttg',
6.869767612130462),
('ccattgaatgagacagacatgacc', 6.869767612130462),
('cccatgcctatgccactccat', 6.869767612130462),
('cccattttcctccctcaattcctagaaaa', 6.869767612130462),
('cccgaaaagcactcaataat', 6.869767612130462),
('ccctacagcttcaatattatctttgacaacttgatcctggttgatttg',
6.869767612130462),
('ccctggagggttccatcattta', 6.869767612130462),
('cce', 6.869767612130462),
('ccgaacacctttttgagtcaacatcaagtattggtctctcgtctttc',
6.869767612130462),
('ccgccagggcctcccctcaccgaggaagacgttccaggag', 6.86976761213
0462),
2),
('ccggaattcggatcctggggcgctctgagcccg', 6.86976761213046
2),
('ccggactcggatcaaattct', 6.869767612130462),
('cci', 6.869767612130462),
('ccl2', 6.869767612130462),
('ccl3', 6.869767612130462),
```

```
('ccl4', 6.869767612130462),
('ccm', 6.869767612130462),
('ccna1', 6.869767612130462),
('ccttcaccaacaagcccacagatcaactg', 6.869767612130462),
('ccttcaccaacacgcccacagatcaactg', 6.869767612130462),
('ccttcaccaacaggcccacagatcaactg', 6.869767612130462),
('ccttcaccaacgagcccacagatcaactg', 6.869767612130462),
('ccttcaccaacgtgcccacagatcaactg', 6.869767612130462),
('cd14', 6.869767612130462),
('cd19low', 6.869767612130462),
('cd27', 6.869767612130462),
('cd58', 6.869767612130462),
('cd69', 6.869767612130462),
('cd8a', 6.869767612130462),
('cdc25b', 6.869767612130462),
('cdc7', 6.869767612130462),
('cdh11', 6.869767612130462),
('cdh13', 6.869767612130462),
('cdh2', 6.869767612130462),
('cdk4r24c', 6.869767612130462),
('cdnaclonecontainingtheentirecodingsequence', 6.869767612
130462),
('cebpe', 6.869767612130462),
('ced', 6.869767612130462),
('cediranib', 6.869767612130462),
('ceive', 6.869767612130462),
('cellomics', 6.869767612130462),
('cells33', 6.869767612130462),
('cells5', 6.869767612130462),
('cellsignal', 6.869767612130462),
('cellsto', 6.869767612130462),
('celltiter96aqueousone', 6.869767612130462),
('cen6', 6.869767612130462),
('cenp', 6.869767612130462),
('censoring', 6.869767612130462),
('centriprep', 6.869767612130462),
('centroblast', 6.869767612130462),
('centromerically', 6.869767612130462),
('centuries', 6.869767612130462),
```

```
        ('cep110', 6.869767612130462),  
        ('cep85l', 6.869767612130462),  
        ('cephalic', 6.869767612130462),  
        ('cerebrooculofacioskeletal', 6.869767612130462),  
        ('cerione', 6.869767612130462),  
        ...])
```

```
In [45]: # getting top 1k features using idf_values  
top_tfidf_features = list(sorted_by_value.keys())[:1000]
```

```
In [46]: dict_list = []  
# dict_list=[] contains 9 dictionaries each corresponds to a class  
for i in range(1,10):  
    cls_text = train_df[train_df['Class']==i]  
    # build a word dict based on the words in that class  
    dict_list.append(extract_dictionary_paddle(cls_text))  
    # append it to dict_list  
  
# dict_list[i] is build on i'th class text data  
# total_dict is build on whole training text data  
total_dict = extract_dictionary_paddle(train_df)  
  
confuse_array = []  
for i in tfidf_train_text_onehotencoding:  
    ratios = []  
    max_val = -1  
    for j in range(0,9):  
        ratios.append((dict_list[j][i]+10)/(total_dict[i]+90))  
    confuse_array.append(ratios)  
confuse_array = np.array(confuse_array)
```

```
In [47]: #response coding of text features  
train_text_feature_responseCoding = get_text_responsecoding(train_df)  
test_text_feature_responseCoding = get_text_responsecoding(test_df)  
cv_text_feature_responseCoding = get_text_responsecoding(cv_df)
```

```
In [48]: # https://stackoverflow.com/a/16202486
# we convert each row values such that they sum to 1
train_text_feature_responseCoding = (train_text_feature_responseCoding.T/train_text_feature_responseCoding.sum(axis=1)).T
test_text_feature_responseCoding = (test_text_feature_responseCoding.T/test_text_feature_responseCoding.sum(axis=1)).T
cv_text_feature_responseCoding = (cv_text_feature_responseCoding.T/cv_text_ext_feature_responseCoding.sum(axis=1)).T
```

```
In [49]: # don't forget to normalize every feature
train_text_feature_onehotCoding = normalize(train_text_feature_onehotCoding, axis=0)

# we use the same vectorizer that was trained on train data
test_text_feature_onehotCoding = text_vectorizer.transform(test_df['TEXT'])
# don't forget to normalize every feature
test_text_feature_onehotCoding = normalize(test_text_feature_onehotCoding, axis=0)

# we use the same vectorizer that was trained on train data
cv_text_feature_onehotCoding = text_vectorizer.transform(cv_df['TEXT'])
# don't forget to normalize every feature
cv_text_feature_onehotCoding = normalize(cv_text_feature_onehotCoding, axis=0)
```

```
In [50]: #https://stackoverflow.com/a/2258273/4084039
sorted_text_fea_dict = dict(sorted(text_fea_dict.items(), key=lambda x: x[1], reverse=True))
sorted_text_occur = np.array(list(sorted_text_fea_dict.values()))
```

```
In [51]: # Number of words for a given frequency.
print(Counter(sorted_text_occur))
```

```
Counter({38761: 1, 38760: 1, 38759: 1, 38758: 1, 38757: 1, 38756: 1, 38755: 1, 38754: 1, 38753: 1, 38752: 1, 38751: 1, 38750: 1, 38749: 1, 38748: 1, 38747: 1, 38746: 1, 38745: 1, 38744: 1, 38743: 1, 38742: 1, 38741: 1, 38740: 1, 38739: 1, 38738: 1, 38737: 1, 38736: 1, 38735: 1, 3873
```

4: 1, 38733: 1, 38732: 1, 38731: 1, 38730: 1, 38729: 1, 38728: 1, 3872
7: 1, 38726: 1, 38725: 1, 38724: 1, 38723: 1, 38722: 1, 38721: 1, 3872
0: 1, 38719: 1, 38718: 1, 38717: 1, 38716: 1, 38715: 1, 38714: 1, 3871
3: 1, 38712: 1, 38711: 1, 38710: 1, 38709: 1, 38708: 1, 38707: 1, 3870
6: 1, 38705: 1, 38704: 1, 38703: 1, 38702: 1, 38701: 1, 38700: 1, 3869
9: 1, 38698: 1, 38697: 1, 38696: 1, 38695: 1, 38694: 1, 38693: 1, 3869
2: 1, 38691: 1, 38690: 1, 38689: 1, 38688: 1, 38687: 1, 38686: 1, 3868
5: 1, 38684: 1, 38683: 1, 38682: 1, 38681: 1, 38680: 1, 38679: 1, 3867
8: 1, 38677: 1, 38676: 1, 38675: 1, 38674: 1, 38673: 1, 38672: 1, 3867
1: 1, 38670: 1, 38669: 1, 38668: 1, 38667: 1, 38666: 1, 38665: 1, 3866
4: 1, 38663: 1, 38662: 1, 38661: 1, 38660: 1, 38659: 1, 38658: 1, 3865
7: 1, 38656: 1, 38655: 1, 38654: 1, 38653: 1, 38652: 1, 38651: 1, 3865
0: 1, 38649: 1, 38648: 1, 38647: 1, 38646: 1, 38645: 1, 38644: 1, 3864
3: 1, 38642: 1, 38641: 1, 38640: 1, 38639: 1, 38638: 1, 38637: 1, 3863
6: 1, 38635: 1, 38634: 1, 38633: 1, 38632: 1, 38631: 1, 38630: 1, 3862
9: 1, 38628: 1, 38627: 1, 38626: 1, 38625: 1, 38624: 1, 38623: 1, 3862
2: 1, 38621: 1, 38620: 1, 38619: 1, 38618: 1, 38617: 1, 38616: 1, 3861
5: 1, 38614: 1, 38613: 1, 38612: 1, 38611: 1, 38610: 1, 38609: 1, 3860
8: 1, 38607: 1, 38606: 1, 38605: 1, 38604: 1, 38603: 1, 38602: 1, 3860
1: 1, 38600: 1, 38599: 1, 38598: 1, 38597: 1, 38596: 1, 38595: 1, 3859
4: 1, 38593: 1, 38592: 1, 38591: 1, 38590: 1, 38589: 1, 38588: 1, 3858
7: 1, 38586: 1, 38585: 1, 38584: 1, 38583: 1, 38582: 1, 38581: 1, 3858
0: 1, 38579: 1, 38578: 1, 38577: 1, 38576: 1, 38575: 1, 38574: 1, 3857
3: 1, 38572: 1, 38571: 1, 38570: 1, 38569: 1, 38568: 1, 38567: 1, 3856
6: 1, 38565: 1, 38564: 1, 38563: 1, 38562: 1, 38561: 1, 38560: 1, 3855
9: 1, 38558: 1, 38557: 1, 38556: 1, 38555: 1, 38554: 1, 38553: 1, 3855
2: 1, 38551: 1, 38550: 1, 38549: 1, 38548: 1, 38547: 1, 38546: 1, 3854
5: 1, 38544: 1, 38543: 1, 38542: 1, 38541: 1, 38540: 1, 38539: 1, 3853
8: 1, 38537: 1, 38536: 1, 38535: 1, 38534: 1, 38533: 1, 38532: 1, 3853
1: 1, 38530: 1, 38529: 1, 38528: 1, 38527: 1, 38526: 1, 38525: 1, 3852
4: 1, 38523: 1, 38522: 1, 38521: 1, 38520: 1, 38519: 1, 38518: 1, 3851
7: 1, 38516: 1, 38515: 1, 38514: 1, 38513: 1, 38512: 1, 38511: 1, 3851
0: 1, 38509: 1, 38508: 1, 38507: 1, 38506: 1, 38505: 1, 38504: 1, 3850
3: 1, 38502: 1, 38501: 1, 38500: 1, 38499: 1, 38498: 1, 38497: 1, 3849
6: 1, 38495: 1, 38494: 1, 38493: 1, 38492: 1, 38491: 1, 38490: 1, 3848
9: 1, 38488: 1, 38487: 1, 38486: 1, 38485: 1, 38484: 1, 38483: 1, 3848
2: 1, 38481: 1, 38480: 1, 38479: 1, 38478: 1, 38477: 1, 38476: 1, 3847
5: 1, 38474: 1, 38473: 1, 38472: 1, 38471: 1, 38470: 1, 38469: 1, 3846
8: 1, 38467: 1, 38466: 1, 38465: 1, 38464: 1, 38463: 1, 38462: 1, 3846

1: 1, 38460: 1, 38459: 1, 38458: 1, 38457: 1, 38456: 1, 38455: 1, 3845
4: 1, 38453: 1, 38452: 1, 38451: 1, 38450: 1, 38449: 1, 38448: 1, 3844
7: 1, 38446: 1, 38445: 1, 38444: 1, 38443: 1, 38442: 1, 38441: 1, 3844
0: 1, 38439: 1, 38438: 1, 38437: 1, 38436: 1, 38435: 1, 38434: 1, 3843
3: 1, 38432: 1, 38431: 1, 38430: 1, 38429: 1, 38428: 1, 38427: 1, 3842
6: 1, 38425: 1, 38424: 1, 38423: 1, 38422: 1, 38421: 1, 38420: 1, 3841
9: 1, 38418: 1, 38417: 1, 38416: 1, 38415: 1, 38414: 1, 38413: 1, 3841
2: 1, 38411: 1, 38410: 1, 38409: 1, 38408: 1, 38407: 1, 38406: 1, 3840
5: 1, 38404: 1, 38403: 1, 38402: 1, 38401: 1, 38400: 1, 38399: 1, 3839
8: 1, 38397: 1, 38396: 1, 38395: 1, 38394: 1, 38393: 1, 38392: 1, 3839
1: 1, 38390: 1, 38389: 1, 38388: 1, 38387: 1, 38386: 1, 38385: 1, 3838
4: 1, 38383: 1, 38382: 1, 38381: 1, 38380: 1, 38379: 1, 38378: 1, 3837
7: 1, 38376: 1, 38375: 1, 38374: 1, 38373: 1, 38372: 1, 38371: 1, 3837
0: 1, 38369: 1, 38368: 1, 38367: 1, 38366: 1, 38365: 1, 38364: 1, 3836
3: 1, 38362: 1, 38361: 1, 38360: 1, 38359: 1, 38358: 1, 38357: 1, 3835
6: 1, 38355: 1, 38354: 1, 38353: 1, 38352: 1, 38351: 1, 38350: 1, 3834
9: 1, 38348: 1, 38347: 1, 38346: 1, 38345: 1, 38344: 1, 38343: 1, 3834
2: 1, 38341: 1, 38340: 1, 38339: 1, 38338: 1, 38337: 1, 38336: 1, 3833
5: 1, 38334: 1, 38333: 1, 38332: 1, 38331: 1, 38330: 1, 38329: 1, 3832
8: 1, 38327: 1, 38326: 1, 38325: 1, 38324: 1, 38323: 1, 38322: 1, 3832
1: 1, 38320: 1, 38319: 1, 38318: 1, 38317: 1, 38316: 1, 38315: 1, 3831
4: 1, 38313: 1, 38312: 1, 38311: 1, 38310: 1, 38309: 1, 38308: 1, 3830
7: 1, 38306: 1, 38305: 1, 38304: 1, 38303: 1, 38302: 1, 38301: 1, 3830
0: 1, 38299: 1, 38298: 1, 38297: 1, 38296: 1, 38295: 1, 38294: 1, 3829
3: 1, 38292: 1, 38291: 1, 38290: 1, 38289: 1, 38288: 1, 38287: 1, 3828
6: 1, 38285: 1, 38284: 1, 38283: 1, 38282: 1, 38281: 1, 38280: 1, 3827
9: 1, 38278: 1, 38277: 1, 38276: 1, 38275: 1, 38274: 1, 38273: 1, 3827
2: 1, 38271: 1, 38270: 1, 38269: 1, 38268: 1, 38267: 1, 38266: 1, 3826
5: 1, 38264: 1, 38263: 1, 38262: 1, 38261: 1, 38260: 1, 38259: 1, 3825
8: 1, 38257: 1, 38256: 1, 38255: 1, 38254: 1, 38253: 1, 38252: 1, 3825
1: 1, 38250: 1, 38249: 1, 38248: 1, 38247: 1, 38246: 1, 38245: 1, 3824
4: 1, 38243: 1, 38242: 1, 38241: 1, 38240: 1, 38239: 1, 38238: 1, 3823
7: 1, 38236: 1, 38235: 1, 38234: 1, 38233: 1, 38232: 1, 38231: 1, 3823
0: 1, 38229: 1, 38228: 1, 38227: 1, 38226: 1, 38225: 1, 38224: 1, 3822
3: 1, 38222: 1, 38221: 1, 38220: 1, 38219: 1, 38218: 1, 38217: 1, 3821
6: 1, 38215: 1, 38214: 1, 38213: 1, 38212: 1, 38211: 1, 38210: 1, 3820
9: 1, 38208: 1, 38207: 1, 38206: 1, 38205: 1, 38204: 1, 38203: 1, 3820
2: 1, 38201: 1, 38200: 1, 38199: 1, 38198: 1, 38197: 1, 38196: 1, 3819
5: 1, 38194: 1, 38193: 1, 38192: 1, 38191: 1, 38190: 1, 38189: 1, 3818

8: 1, 38187: 1, 38186: 1, 38185: 1, 38184: 1, 38183: 1, 38182: 1, 3818
1: 1, 38180: 1, 38179: 1, 38178: 1, 38177: 1, 38176: 1, 38175: 1, 3817
4: 1, 38173: 1, 38172: 1, 38171: 1, 38170: 1, 38169: 1, 38168: 1, 3816
7: 1, 38166: 1, 38165: 1, 38164: 1, 38163: 1, 38162: 1, 38161: 1, 3816
0: 1, 38159: 1, 38158: 1, 38157: 1, 38156: 1, 38155: 1, 38154: 1, 3815
3: 1, 38152: 1, 38151: 1, 38150: 1, 38149: 1, 38148: 1, 38147: 1, 3814
6: 1, 38145: 1, 38144: 1, 38143: 1, 38142: 1, 38141: 1, 38140: 1, 3813
9: 1, 38138: 1, 38137: 1, 38136: 1, 38135: 1, 38134: 1, 38133: 1, 3813
2: 1, 38131: 1, 38130: 1, 38129: 1, 38128: 1, 38127: 1, 38126: 1, 3812
5: 1, 38124: 1, 38123: 1, 38122: 1, 38121: 1, 38120: 1, 38119: 1, 3811
8: 1, 38117: 1, 38116: 1, 38115: 1, 38114: 1, 38113: 1, 38112: 1, 3811
1: 1, 38110: 1, 38109: 1, 38108: 1, 38107: 1, 38106: 1, 38105: 1, 3810
4: 1, 38103: 1, 38102: 1, 38101: 1, 38100: 1, 38099: 1, 38098: 1, 3809
7: 1, 38096: 1, 38095: 1, 38094: 1, 38093: 1, 38092: 1, 38091: 1, 3809
0: 1, 38089: 1, 38088: 1, 38087: 1, 38086: 1, 38085: 1, 38084: 1, 3808
3: 1, 38082: 1, 38081: 1, 38080: 1, 38079: 1, 38078: 1, 38077: 1, 3807
6: 1, 38075: 1, 38074: 1, 38073: 1, 38072: 1, 38071: 1, 38070: 1, 3806
9: 1, 38068: 1, 38067: 1, 38066: 1, 38065: 1, 38064: 1, 38063: 1, 3806
2: 1, 38061: 1, 38060: 1, 38059: 1, 38058: 1, 38057: 1, 38056: 1, 3805
5: 1, 38054: 1, 38053: 1, 38052: 1, 38051: 1, 38050: 1, 38049: 1, 3804
8: 1, 38047: 1, 38046: 1, 38045: 1, 38044: 1, 38043: 1, 38042: 1, 3804
1: 1, 38040: 1, 38039: 1, 38038: 1, 38037: 1, 38036: 1, 38035: 1, 3803
4: 1, 38033: 1, 38032: 1, 38031: 1, 38030: 1, 38029: 1, 38028: 1, 3802
7: 1, 38026: 1, 38025: 1, 38024: 1, 38023: 1, 38022: 1, 38021: 1, 3802
0: 1, 38019: 1, 38018: 1, 38017: 1, 38016: 1, 38015: 1, 38014: 1, 3801
3: 1, 38012: 1, 38011: 1, 38010: 1, 38009: 1, 38008: 1, 38007: 1, 3800
6: 1, 38005: 1, 38004: 1, 38003: 1, 38002: 1, 38001: 1, 38000: 1, 3799
9: 1, 37998: 1, 37997: 1, 37996: 1, 37995: 1, 37994: 1, 37993: 1, 3799
2: 1, 37991: 1, 37990: 1, 37989: 1, 37988: 1, 37987: 1, 37986: 1, 3798
5: 1, 37984: 1, 37983: 1, 37982: 1, 37981: 1, 37980: 1, 37979: 1, 3797
8: 1, 37977: 1, 37976: 1, 37975: 1, 37974: 1, 37973: 1, 37972: 1, 3797
1: 1, 37970: 1, 37969: 1, 37968: 1, 37967: 1, 37966: 1, 37965: 1, 3796
4: 1, 37963: 1, 37962: 1, 37961: 1, 37960: 1, 37959: 1, 37958: 1, 3795
7: 1, 37956: 1, 37955: 1, 37954: 1, 37953: 1, 37952: 1, 37951: 1, 3795
0: 1, 37949: 1, 37948: 1, 37947: 1, 37946: 1, 37945: 1, 37944: 1, 3794
3: 1, 37942: 1, 37941: 1, 37940: 1, 37939: 1, 37938: 1, 37937: 1, 3793
6: 1, 37935: 1, 37934: 1, 37933: 1, 37932: 1, 37931: 1, 37930: 1, 3792
9: 1, 37928: 1, 37927: 1, 37926: 1, 37925: 1, 37924: 1, 37923: 1, 3792
2: 1, 37921: 1, 37920: 1, 37919: 1, 37918: 1, 37917: 1, 37916: 1, 3791

5: 1, 37914: 1, 37913: 1, 37912: 1, 37911: 1, 37910: 1, 37909: 1, 3790
8: 1, 37907: 1, 37906: 1, 37905: 1, 37904: 1, 37903: 1, 37902: 1, 3790
1: 1, 37900: 1, 37899: 1, 37898: 1, 37897: 1, 37896: 1, 37895: 1, 3789
4: 1, 37893: 1, 37892: 1, 37891: 1, 37890: 1, 37889: 1, 37888: 1, 3788
7: 1, 37886: 1, 37885: 1, 37884: 1, 37883: 1, 37882: 1, 37881: 1, 3788
0: 1, 37879: 1, 37878: 1, 37877: 1, 37876: 1, 37875: 1, 37874: 1, 3787
3: 1, 37872: 1, 37871: 1, 37870: 1, 37869: 1, 37868: 1, 37867: 1, 3786
6: 1, 37865: 1, 37864: 1, 37863: 1, 37862: 1, 37861: 1, 37860: 1, 3785
9: 1, 37858: 1, 37857: 1, 37856: 1, 37855: 1, 37854: 1, 37853: 1, 3785
2: 1, 37851: 1, 37850: 1, 37849: 1, 37848: 1, 37847: 1, 37846: 1, 3784
5: 1, 37844: 1, 37843: 1, 37842: 1, 37841: 1, 37840: 1, 37839: 1, 3783
8: 1, 37837: 1, 37836: 1, 37835: 1, 37834: 1, 37833: 1, 37832: 1, 3783
1: 1, 37830: 1, 37829: 1, 37828: 1, 37827: 1, 37826: 1, 37825: 1, 3782
4: 1, 37823: 1, 37822: 1, 37821: 1, 37820: 1, 37819: 1, 37818: 1, 3781
7: 1, 37816: 1, 37815: 1, 37814: 1, 37813: 1, 37812: 1, 37811: 1, 3781
0: 1, 37809: 1, 37808: 1, 37807: 1, 37806: 1, 37805: 1, 37804: 1, 3780
3: 1, 37802: 1, 37801: 1, 37800: 1, 37799: 1, 37798: 1, 37797: 1, 3779
6: 1, 37795: 1, 37794: 1, 37793: 1, 37792: 1, 37791: 1, 37790: 1, 3778
9: 1, 37788: 1, 37787: 1, 37786: 1, 37785: 1, 37784: 1, 37783: 1, 3778
2: 1, 37781: 1, 37780: 1, 37779: 1, 37778: 1, 37777: 1, 37776: 1, 3777
5: 1, 37774: 1, 37773: 1, 37772: 1, 37771: 1, 37770: 1, 37769: 1, 3776
8: 1, 37767: 1, 37766: 1, 37765: 1, 37764: 1, 37763: 1, 37762: 1, 3776
1: 1, 37760: 1, 37759: 1, 37758: 1, 37757: 1, 37756: 1, 37755: 1, 3775
4: 1, 37753: 1, 37752: 1, 37751: 1, 37750: 1, 37749: 1, 37748: 1, 3774
7: 1, 37746: 1, 37745: 1, 37744: 1, 37743: 1, 37742: 1, 37741: 1, 3774
0: 1, 37739: 1, 37738: 1, 37737: 1, 37736: 1, 37735: 1, 37734: 1, 3773
3: 1, 37732: 1, 37731: 1, 37730: 1, 37729: 1, 37728: 1, 37727: 1, 3772
6: 1, 37725: 1, 37724: 1, 37723: 1, 37722: 1, 37721: 1, 37720: 1, 3771
9: 1, 37718: 1, 37717: 1, 37716: 1, 37715: 1, 37714: 1, 37713: 1, 3771
2: 1, 37711: 1, 37710: 1, 37709: 1, 37708: 1, 37707: 1, 37706: 1, 3770
5: 1, 37704: 1, 37703: 1, 37702: 1, 37701: 1, 37700: 1, 37699: 1, 3769
8: 1, 37697: 1, 37696: 1, 37695: 1, 37694: 1, 37693: 1, 37692: 1, 3769
1: 1, 37690: 1, 37689: 1, 37688: 1, 37687: 1, 37686: 1, 37685: 1, 3768
4: 1, 37683: 1, 37682: 1, 37681: 1, 37680: 1, 37679: 1, 37678: 1, 3767
7: 1, 37676: 1, 37675: 1, 37674: 1, 37673: 1, 37672: 1, 37671: 1, 3767
0: 1, 37669: 1, 37668: 1, 37667: 1, 37666: 1, 37665: 1, 37664: 1, 3766
3: 1, 37662: 1, 37661: 1, 37660: 1, 37659: 1, 37658: 1, 37657: 1, 3765
6: 1, 37655: 1, 37654: 1, 37653: 1, 37652: 1, 37651: 1, 37650: 1, 3764
9: 1, 37648: 1, 37647: 1, 37646: 1, 37645: 1, 37644: 1, 37643: 1, 3764

2: 1, 37641: 1, 37640: 1, 37639: 1, 37638: 1, 37637: 1, 37636: 1, 3763
5: 1, 37634: 1, 37633: 1, 37632: 1, 37631: 1, 37630: 1, 37629: 1, 3762
8: 1, 37627: 1, 37626: 1, 37625: 1, 37624: 1, 37623: 1, 37622: 1, 3762
1: 1, 37620: 1, 37619: 1, 37618: 1, 37617: 1, 37616: 1, 37615: 1, 3761
4: 1, 37613: 1, 37612: 1, 37611: 1, 37610: 1, 37609: 1, 37608: 1, 3760
7: 1, 37606: 1, 37605: 1, 37604: 1, 37603: 1, 37602: 1, 37601: 1, 3760
0: 1, 37599: 1, 37598: 1, 37597: 1, 37596: 1, 37595: 1, 37594: 1, 3759
3: 1, 37592: 1, 37591: 1, 37590: 1, 37589: 1, 37588: 1, 37587: 1, 3758
6: 1, 37585: 1, 37584: 1, 37583: 1, 37582: 1, 37581: 1, 37580: 1, 3757
9: 1, 37578: 1, 37577: 1, 37576: 1, 37575: 1, 37574: 1, 37573: 1, 3757
2: 1, 37571: 1, 37570: 1, 37569: 1, 37568: 1, 37567: 1, 37566: 1, 3756
5: 1, 37564: 1, 37563: 1, 37562: 1, 37561: 1, 37560: 1, 37559: 1, 3755
8: 1, 37557: 1, 37556: 1, 37555: 1, 37554: 1, 37553: 1, 37552: 1, 3755
1: 1, 37550: 1, 37549: 1, 37548: 1, 37547: 1, 37546: 1, 37545: 1, 3754
4: 1, 37543: 1, 37542: 1, 37541: 1, 37540: 1, 37539: 1, 37538: 1, 3753
7: 1, 37536: 1, 37535: 1, 37534: 1, 37533: 1, 37532: 1, 37531: 1, 3753
0: 1, 37529: 1, 37528: 1, 37527: 1, 37526: 1, 37525: 1, 37524: 1, 3752
3: 1, 37522: 1, 37521: 1, 37520: 1, 37519: 1, 37518: 1, 37517: 1, 3751
6: 1, 37515: 1, 37514: 1, 37513: 1, 37512: 1, 37511: 1, 37510: 1, 3750
9: 1, 37508: 1, 37507: 1, 37506: 1, 37505: 1, 37504: 1, 37503: 1, 3750
2: 1, 37501: 1, 37500: 1, 37499: 1, 37498: 1, 37497: 1, 37496: 1, 3749
5: 1, 37494: 1, 37493: 1, 37492: 1, 37491: 1, 37490: 1, 37489: 1, 3748
8: 1, 37487: 1, 37486: 1, 37485: 1, 37484: 1, 37483: 1, 37482: 1, 3748
1: 1, 37480: 1, 37479: 1, 37478: 1, 37477: 1, 37476: 1, 37475: 1, 3747
4: 1, 37473: 1, 37472: 1, 37471: 1, 37470: 1, 37469: 1, 37468: 1, 3746
7: 1, 37466: 1, 37465: 1, 37464: 1, 37463: 1, 37462: 1, 37461: 1, 3746
0: 1, 37459: 1, 37458: 1, 37457: 1, 37456: 1, 37455: 1, 37454: 1, 3745
3: 1, 37452: 1, 37451: 1, 37450: 1, 37449: 1, 37448: 1, 37447: 1, 3744
6: 1, 37445: 1, 37444: 1, 37443: 1, 37442: 1, 37441: 1, 37440: 1, 3743
9: 1, 37438: 1, 37437: 1, 37436: 1, 37435: 1, 37434: 1, 37433: 1, 3743
2: 1, 37431: 1, 37430: 1, 37429: 1, 37428: 1, 37427: 1, 37426: 1, 3742
5: 1, 37424: 1, 37423: 1, 37422: 1, 37421: 1, 37420: 1, 37419: 1, 3741
8: 1, 37417: 1, 37416: 1, 37415: 1, 37414: 1, 37413: 1, 37412: 1, 3741
1: 1, 37410: 1, 37409: 1, 37408: 1, 37407: 1, 37406: 1, 37405: 1, 3740
4: 1, 37403: 1, 37402: 1, 37401: 1, 37400: 1, 37399: 1, 37398: 1, 3739
7: 1, 37396: 1, 37395: 1, 37394: 1, 37393: 1, 37392: 1, 37391: 1, 3739
0: 1, 37389: 1, 37388: 1, 37387: 1, 37386: 1, 37385: 1, 37384: 1, 3738
3: 1, 37382: 1, 37381: 1, 37380: 1, 37379: 1, 37378: 1, 37377: 1, 3737
6: 1, 37375: 1, 37374: 1, 37373: 1, 37372: 1, 37371: 1, 37370: 1, 3736

9: 1, 37368: 1, 37367: 1, 37366: 1, 37365: 1, 37364: 1, 37363: 1, 3736
2: 1, 37361: 1, 37360: 1, 37359: 1, 37358: 1, 37357: 1, 37356: 1, 3735
5: 1, 37354: 1, 37353: 1, 37352: 1, 37351: 1, 37350: 1, 37349: 1, 3734
8: 1, 37347: 1, 37346: 1, 37345: 1, 37344: 1, 37343: 1, 37342: 1, 3734
1: 1, 37340: 1, 37339: 1, 37338: 1, 37337: 1, 37336: 1, 37335: 1, 3733
4: 1, 37333: 1, 37332: 1, 37331: 1, 37330: 1, 37329: 1, 37328: 1, 3732
7: 1, 37326: 1, 37325: 1, 37324: 1, 37323: 1, 37322: 1, 37321: 1, 3732
0: 1, 37319: 1, 37318: 1, 37317: 1, 37316: 1, 37315: 1, 37314: 1, 3731
3: 1, 37312: 1, 37311: 1, 37310: 1, 37309: 1, 37308: 1, 37307: 1, 3730
6: 1, 37305: 1, 37304: 1, 37303: 1, 37302: 1, 37301: 1, 37300: 1, 3729
9: 1, 37298: 1, 37297: 1, 37296: 1, 37295: 1, 37294: 1, 37293: 1, 3729
2: 1, 37291: 1, 37290: 1, 37289: 1, 37288: 1, 37287: 1, 37286: 1, 3728
5: 1, 37284: 1, 37283: 1, 37282: 1, 37281: 1, 37280: 1, 37279: 1, 3727
8: 1, 37277: 1, 37276: 1, 37275: 1, 37274: 1, 37273: 1, 37272: 1, 3727
1: 1, 37270: 1, 37269: 1, 37268: 1, 37267: 1, 37266: 1, 37265: 1, 3726
4: 1, 37263: 1, 37262: 1, 37261: 1, 37260: 1, 37259: 1, 37258: 1, 3725
7: 1, 37256: 1, 37255: 1, 37254: 1, 37253: 1, 37252: 1, 37251: 1, 3725
0: 1, 37249: 1, 37248: 1, 37247: 1, 37246: 1, 37245: 1, 37244: 1, 3724
3: 1, 37242: 1, 37241: 1, 37240: 1, 37239: 1, 37238: 1, 37237: 1, 3723
6: 1, 37235: 1, 37234: 1, 37233: 1, 37232: 1, 37231: 1, 37230: 1, 3722
9: 1, 37228: 1, 37227: 1, 37226: 1, 37225: 1, 37224: 1, 37223: 1, 3722
2: 1, 37221: 1, 37220: 1, 37219: 1, 37218: 1, 37217: 1, 37216: 1, 3721
5: 1, 37214: 1, 37213: 1, 37212: 1, 37211: 1, 37210: 1, 37209: 1, 3720
8: 1, 37207: 1, 37206: 1, 37205: 1, 37204: 1, 37203: 1, 37202: 1, 3720
1: 1, 37200: 1, 37199: 1, 37198: 1, 37197: 1, 37196: 1, 37195: 1, 3719
4: 1, 37193: 1, 37192: 1, 37191: 1, 37190: 1, 37189: 1, 37188: 1, 3718
7: 1, 37186: 1, 37185: 1, 37184: 1, 37183: 1, 37182: 1, 37181: 1, 3718
0: 1, 37179: 1, 37178: 1, 37177: 1, 37176: 1, 37175: 1, 37174: 1, 3717
3: 1, 37172: 1, 37171: 1, 37170: 1, 37169: 1, 37168: 1, 37167: 1, 3716
6: 1, 37165: 1, 37164: 1, 37163: 1, 37162: 1, 37161: 1, 37160: 1, 3715
9: 1, 37158: 1, 37157: 1, 37156: 1, 37155: 1, 37154: 1, 37153: 1, 3715
2: 1, 37151: 1, 37150: 1, 37149: 1, 37148: 1, 37147: 1, 37146: 1, 3714
5: 1, 37144: 1, 37143: 1, 37142: 1, 37141: 1, 37140: 1, 37139: 1, 3713
8: 1, 37137: 1, 37136: 1, 37135: 1, 37134: 1, 37133: 1, 37132: 1, 3713
1: 1, 37130: 1, 37129: 1, 37128: 1, 37127: 1, 37126: 1, 37125: 1, 3712
4: 1, 37123: 1, 37122: 1, 37121: 1, 37120: 1, 37119: 1, 37118: 1, 3711
7: 1, 37116: 1, 37115: 1, 37114: 1, 37113: 1, 37112: 1, 37111: 1, 3711
0: 1, 37109: 1, 37108: 1, 37107: 1, 37106: 1, 37105: 1, 37104: 1, 3710
3: 1, 37102: 1, 37101: 1, 37100: 1, 37099: 1, 37098: 1, 37097: 1, 3709

6: 1, 37095: 1, 37094: 1, 37093: 1, 37092: 1, 37091: 1, 37090: 1, 3708
9: 1, 37088: 1, 37087: 1, 37086: 1, 37085: 1, 37084: 1, 37083: 1, 3708
2: 1, 37081: 1, 37080: 1, 37079: 1, 37078: 1, 37077: 1, 37076: 1, 3707
5: 1, 37074: 1, 37073: 1, 37072: 1, 37071: 1, 37070: 1, 37069: 1, 3706
8: 1, 37067: 1, 37066: 1, 37065: 1, 37064: 1, 37063: 1, 37062: 1, 3706
1: 1, 37060: 1, 37059: 1, 37058: 1, 37057: 1, 37056: 1, 37055: 1, 3705
4: 1, 37053: 1, 37052: 1, 37051: 1, 37050: 1, 37049: 1, 37048: 1, 3704
7: 1, 37046: 1, 37045: 1, 37044: 1, 37043: 1, 37042: 1, 37041: 1, 3704
0: 1, 37039: 1, 37038: 1, 37037: 1, 37036: 1, 37035: 1, 37034: 1, 3703
3: 1, 37032: 1, 37031: 1, 37030: 1, 37029: 1, 37028: 1, 37027: 1, 3702
6: 1, 37025: 1, 37024: 1, 37023: 1, 37022: 1, 37021: 1, 37020: 1, 3701
9: 1, 37018: 1, 37017: 1, 37016: 1, 37015: 1, 37014: 1, 37013: 1, 3701
2: 1, 37011: 1, 37010: 1, 37009: 1, 37008: 1, 37007: 1, 37006: 1, 3700
5: 1, 37004: 1, 37003: 1, 37002: 1, 37001: 1, 37000: 1, 36999: 1, 3699
8: 1, 36997: 1, 36996: 1, 36995: 1, 36994: 1, 36993: 1, 36992: 1, 3699
1: 1, 36990: 1, 36989: 1, 36988: 1, 36987: 1, 36986: 1, 36985: 1, 3698
4: 1, 36983: 1, 36982: 1, 36981: 1, 36980: 1, 36979: 1, 36978: 1, 3697
7: 1, 36976: 1, 36975: 1, 36974: 1, 36973: 1, 36972: 1, 36971: 1, 3697
0: 1, 36969: 1, 36968: 1, 36967: 1, 36966: 1, 36965: 1, 36964: 1, 3696
3: 1, 36962: 1, 36961: 1, 36960: 1, 36959: 1, 36958: 1, 36957: 1, 3695
6: 1, 36955: 1, 36954: 1, 36953: 1, 36952: 1, 36951: 1, 36950: 1, 3694
9: 1, 36948: 1, 36947: 1, 36946: 1, 36945: 1, 36944: 1, 36943: 1, 3694
2: 1, 36941: 1, 36940: 1, 36939: 1, 36938: 1, 36937: 1, 36936: 1, 3693
5: 1, 36934: 1, 36933: 1, 36932: 1, 36931: 1, 36930: 1, 36929: 1, 3692
8: 1, 36927: 1, 36926: 1, 36925: 1, 36924: 1, 36923: 1, 36922: 1, 3692
1: 1, 36920: 1, 36919: 1, 36918: 1, 36917: 1, 36916: 1, 36915: 1, 3691
4: 1, 36913: 1, 36912: 1, 36911: 1, 36910: 1, 36909: 1, 36908: 1, 3690
7: 1, 36906: 1, 36905: 1, 36904: 1, 36903: 1, 36902: 1, 36901: 1, 3690
0: 1, 36899: 1, 36898: 1, 36897: 1, 36896: 1, 36895: 1, 36894: 1, 3689
3: 1, 36892: 1, 36891: 1, 36890: 1, 36889: 1, 36888: 1, 36887: 1, 3688
6: 1, 36885: 1, 36884: 1, 36883: 1, 36882: 1, 36881: 1, 36880: 1, 3687
9: 1, 36878: 1, 36877: 1, 36876: 1, 36875: 1, 36874: 1, 36873: 1, 3687
2: 1, 36871: 1, 36870: 1, 36869: 1, 36868: 1, 36867: 1, 36866: 1, 3686
5: 1, 36864: 1, 36863: 1, 36862: 1, 36861: 1, 36860: 1, 36859: 1, 3685
8: 1, 36857: 1, 36856: 1, 36855: 1, 36854: 1, 36853: 1, 36852: 1, 3685
1: 1, 36850: 1, 36849: 1, 36848: 1, 36847: 1, 36846: 1, 36845: 1, 3684
4: 1, 36843: 1, 36842: 1, 36841: 1, 36840: 1, 36839: 1, 36838: 1, 3683
7: 1, 36836: 1, 36835: 1, 36834: 1, 36833: 1, 36832: 1, 36831: 1, 3683
0: 1, 36829: 1, 36828: 1, 36827: 1, 36826: 1, 36825: 1, 36824: 1, 3682

3: 1, 36822: 1, 36821: 1, 36820: 1, 36819: 1, 36818: 1, 36817: 1, 3681
6: 1, 36815: 1, 36814: 1, 36813: 1, 36812: 1, 36811: 1, 36810: 1, 3680
9: 1, 36808: 1, 36807: 1, 36806: 1, 36805: 1, 36804: 1, 36803: 1, 3680
2: 1, 36801: 1, 36800: 1, 36799: 1, 36798: 1, 36797: 1, 36796: 1, 3679
5: 1, 36794: 1, 36793: 1, 36792: 1, 36791: 1, 36790: 1, 36789: 1, 3678
8: 1, 36787: 1, 36786: 1, 36785: 1, 36784: 1, 36783: 1, 36782: 1, 3678
1: 1, 36780: 1, 36779: 1, 36778: 1, 36777: 1, 36776: 1, 36775: 1, 3677
4: 1, 36773: 1, 36772: 1, 36771: 1, 36770: 1, 36769: 1, 36768: 1, 3676
7: 1, 36766: 1, 36765: 1, 36764: 1, 36763: 1, 36762: 1, 36761: 1, 3676
0: 1, 36759: 1, 36758: 1, 36757: 1, 36756: 1, 36755: 1, 36754: 1, 3675
3: 1, 36752: 1, 36751: 1, 36750: 1, 36749: 1, 36748: 1, 36747: 1, 3674
6: 1, 36745: 1, 36744: 1, 36743: 1, 36742: 1, 36741: 1, 36740: 1, 3673
9: 1, 36738: 1, 36737: 1, 36736: 1, 36735: 1, 36734: 1, 36733: 1, 3673
2: 1, 36731: 1, 36730: 1, 36729: 1, 36728: 1, 36727: 1, 36726: 1, 3672
5: 1, 36724: 1, 36723: 1, 36722: 1, 36721: 1, 36720: 1, 36719: 1, 3671
8: 1, 36717: 1, 36716: 1, 36715: 1, 36714: 1, 36713: 1, 36712: 1, 3671
1: 1, 36710: 1, 36709: 1, 36708: 1, 36707: 1, 36706: 1, 36705: 1, 3670
4: 1, 36703: 1, 36702: 1, 36701: 1, 36700: 1, 36699: 1, 36698: 1, 3669
7: 1, 36696: 1, 36695: 1, 36694: 1, 36693: 1, 36692: 1, 36691: 1, 3669
0: 1, 36689: 1, 36688: 1, 36687: 1, 36686: 1, 36685: 1, 36684: 1, 3668
3: 1, 36682: 1, 36681: 1, 36680: 1, 36679: 1, 36678: 1, 36677: 1, 3667
6: 1, 36675: 1, 36674: 1, 36673: 1, 36672: 1, 36671: 1, 36670: 1, 3666
9: 1, 36668: 1, 36667: 1, 36666: 1, 36665: 1, 36664: 1, 36663: 1, 3666
2: 1, 36661: 1, 36660: 1, 36659: 1, 36658: 1, 36657: 1, 36656: 1, 3665
5: 1, 36654: 1, 36653: 1, 36652: 1, 36651: 1, 36650: 1, 36649: 1, 3664
8: 1, 36647: 1, 36646: 1, 36645: 1, 36644: 1, 36643: 1, 36642: 1, 3664
1: 1, 36640: 1, 36639: 1, 36638: 1, 36637: 1, 36636: 1, 36635: 1, 3663
4: 1, 36633: 1, 36632: 1, 36631: 1, 36630: 1, 36629: 1, 36628: 1, 3662
7: 1, 36626: 1, 36625: 1, 36624: 1, 36623: 1, 36622: 1, 36621: 1, 3662
0: 1, 36619: 1, 36618: 1, 36617: 1, 36616: 1, 36615: 1, 36614: 1, 3661
3: 1, 36612: 1, 36611: 1, 36610: 1, 36609: 1, 36608: 1, 36607: 1, 3660
6: 1, 36605: 1, 36604: 1, 36603: 1, 36602: 1, 36601: 1, 36600: 1, 3659
9: 1, 36598: 1, 36597: 1, 36596: 1, 36595: 1, 36594: 1, 36593: 1, 3659
2: 1, 36591: 1, 36590: 1, 36589: 1, 36588: 1, 36587: 1, 36586: 1, 3658
5: 1, 36584: 1, 36583: 1, 36582: 1, 36581: 1, 36580: 1, 36579: 1, 3657
8: 1, 36577: 1, 36576: 1, 36575: 1, 36574: 1, 36573: 1, 36572: 1, 3657
1: 1, 36570: 1, 36569: 1, 36568: 1, 36567: 1, 36566: 1, 36565: 1, 3656
4: 1, 36563: 1, 36562: 1, 36561: 1, 36560: 1, 36559: 1, 36558: 1, 3655
7: 1, 36556: 1, 36555: 1, 36554: 1, 36553: 1, 36552: 1, 36551: 1, 3655

0: 1, 36549: 1, 36548: 1, 36547: 1, 36546: 1, 36545: 1, 36544: 1, 3654
3: 1, 36542: 1, 36541: 1, 36540: 1, 36539: 1, 36538: 1, 36537: 1, 3653
6: 1, 36535: 1, 36534: 1, 36533: 1, 36532: 1, 36531: 1, 36530: 1, 3652
9: 1, 36528: 1, 36527: 1, 36526: 1, 36525: 1, 36524: 1, 36523: 1, 3652
2: 1, 36521: 1, 36520: 1, 36519: 1, 36518: 1, 36517: 1, 36516: 1, 3651
5: 1, 36514: 1, 36513: 1, 36512: 1, 36511: 1, 36510: 1, 36509: 1, 3650
8: 1, 36507: 1, 36506: 1, 36505: 1, 36504: 1, 36503: 1, 36502: 1, 3650
1: 1, 36500: 1, 36499: 1, 36498: 1, 36497: 1, 36496: 1, 36495: 1, 3649
4: 1, 36493: 1, 36492: 1, 36491: 1, 36490: 1, 36489: 1, 36488: 1, 3648
7: 1, 36486: 1, 36485: 1, 36484: 1, 36483: 1, 36482: 1, 36481: 1, 3648
0: 1, 36479: 1, 36478: 1, 36477: 1, 36476: 1, 36475: 1, 36474: 1, 3647
3: 1, 36472: 1, 36471: 1, 36470: 1, 36469: 1, 36468: 1, 36467: 1, 3646
6: 1, 36465: 1, 36464: 1, 36463: 1, 36462: 1, 36461: 1, 36460: 1, 3645
9: 1, 36458: 1, 36457: 1, 36456: 1, 36455: 1, 36454: 1, 36453: 1, 3645
2: 1, 36451: 1, 36450: 1, 36449: 1, 36448: 1, 36447: 1, 36446: 1, 3644
5: 1, 36444: 1, 36443: 1, 36442: 1, 36441: 1, 36440: 1, 36439: 1, 3643
8: 1, 36437: 1, 36436: 1, 36435: 1, 36434: 1, 36433: 1, 36432: 1, 3643
1: 1, 36430: 1, 36429: 1, 36428: 1, 36427: 1, 36426: 1, 36425: 1, 3642
4: 1, 36423: 1, 36422: 1, 36421: 1, 36420: 1, 36419: 1, 36418: 1, 3641
7: 1, 36416: 1, 36415: 1, 36414: 1, 36413: 1, 36412: 1, 36411: 1, 3641
0: 1, 36409: 1, 36408: 1, 36407: 1, 36406: 1, 36405: 1, 36404: 1, 3640
3: 1, 36402: 1, 36401: 1, 36400: 1, 36399: 1, 36398: 1, 36397: 1, 3639
6: 1, 36395: 1, 36394: 1, 36393: 1, 36392: 1, 36391: 1, 36390: 1, 3638
9: 1, 36388: 1, 36387: 1, 36386: 1, 36385: 1, 36384: 1, 36383: 1, 3638
2: 1, 36381: 1, 36380: 1, 36379: 1, 36378: 1, 36377: 1, 36376: 1, 3637
5: 1, 36374: 1, 36373: 1, 36372: 1, 36371: 1, 36370: 1, 36369: 1, 3636
8: 1, 36367: 1, 36366: 1, 36365: 1, 36364: 1, 36363: 1, 36362: 1, 3636
1: 1, 36360: 1, 36359: 1, 36358: 1, 36357: 1, 36356: 1, 36355: 1, 3635
4: 1, 36353: 1, 36352: 1, 36351: 1, 36350: 1, 36349: 1, 36348: 1, 3634
7: 1, 36346: 1, 36345: 1, 36344: 1, 36343: 1, 36342: 1, 36341: 1, 3634
0: 1, 36339: 1, 36338: 1, 36337: 1, 36336: 1, 36335: 1, 36334: 1, 3633
3: 1, 36332: 1, 36331: 1, 36330: 1, 36329: 1, 36328: 1, 36327: 1, 3632
6: 1, 36325: 1, 36324: 1, 36323: 1, 36322: 1, 36321: 1, 36320: 1, 3631
9: 1, 36318: 1, 36317: 1, 36316: 1, 36315: 1, 36314: 1, 36313: 1, 3631
2: 1, 36311: 1, 36310: 1, 36309: 1, 36308: 1, 36307: 1, 36306: 1, 3630
5: 1, 36304: 1, 36303: 1, 36302: 1, 36301: 1, 36300: 1, 36299: 1, 3629
8: 1, 36297: 1, 36296: 1, 36295: 1, 36294: 1, 36293: 1, 36292: 1, 3629
1: 1, 36290: 1, 36289: 1, 36288: 1, 36287: 1, 36286: 1, 36285: 1, 3628
4: 1, 36283: 1, 36282: 1, 36281: 1, 36280: 1, 36279: 1, 36278: 1, 3627

7: 1, 36276: 1, 36275: 1, 36274: 1, 36273: 1, 36272: 1, 36271: 1, 3627
0: 1, 36269: 1, 36268: 1, 36267: 1, 36266: 1, 36265: 1, 36264: 1, 3626
3: 1, 36262: 1, 36261: 1, 36260: 1, 36259: 1, 36258: 1, 36257: 1, 3625
6: 1, 36255: 1, 36254: 1, 36253: 1, 36252: 1, 36251: 1, 36250: 1, 3624
9: 1, 36248: 1, 36247: 1, 36246: 1, 36245: 1, 36244: 1, 36243: 1, 3624
2: 1, 36241: 1, 36240: 1, 36239: 1, 36238: 1, 36237: 1, 36236: 1, 3623
5: 1, 36234: 1, 36233: 1, 36232: 1, 36231: 1, 36230: 1, 36229: 1, 3622
8: 1, 36227: 1, 36226: 1, 36225: 1, 36224: 1, 36223: 1, 36222: 1, 3622
1: 1, 36220: 1, 36219: 1, 36218: 1, 36217: 1, 36216: 1, 36215: 1, 3621
4: 1, 36213: 1, 36212: 1, 36211: 1, 36210: 1, 36209: 1, 36208: 1, 3620
7: 1, 36206: 1, 36205: 1, 36204: 1, 36203: 1, 36202: 1, 36201: 1, 3620
0: 1, 36199: 1, 36198: 1, 36197: 1, 36196: 1, 36195: 1, 36194: 1, 3619
3: 1, 36192: 1, 36191: 1, 36190: 1, 36189: 1, 36188: 1, 36187: 1, 3618
6: 1, 36185: 1, 36184: 1, 36183: 1, 36182: 1, 36181: 1, 36180: 1, 3617
9: 1, 36178: 1, 36177: 1, 36176: 1, 36175: 1, 36174: 1, 36173: 1, 3617
2: 1, 36171: 1, 36170: 1, 36169: 1, 36168: 1, 36167: 1, 36166: 1, 3616
5: 1, 36164: 1, 36163: 1, 36162: 1, 36161: 1, 36160: 1, 36159: 1, 3615
8: 1, 36157: 1, 36156: 1, 36155: 1, 36154: 1, 36153: 1, 36152: 1, 3615
1: 1, 36150: 1, 36149: 1, 36148: 1, 36147: 1, 36146: 1, 36145: 1, 3614
4: 1, 36143: 1, 36142: 1, 36141: 1, 36140: 1, 36139: 1, 36138: 1, 3613
7: 1, 36136: 1, 36135: 1, 36134: 1, 36133: 1, 36132: 1, 36131: 1, 3613
0: 1, 36129: 1, 36128: 1, 36127: 1, 36126: 1, 36125: 1, 36124: 1, 3612
3: 1, 36122: 1, 36121: 1, 36120: 1, 36119: 1, 36118: 1, 36117: 1, 3611
6: 1, 36115: 1, 36114: 1, 36113: 1, 36112: 1, 36111: 1, 36110: 1, 3610
9: 1, 36108: 1, 36107: 1, 36106: 1, 36105: 1, 36104: 1, 36103: 1, 3610
2: 1, 36101: 1, 36100: 1, 36099: 1, 36098: 1, 36097: 1, 36096: 1, 3609
5: 1, 36094: 1, 36093: 1, 36092: 1, 36091: 1, 36090: 1, 36089: 1, 3608
8: 1, 36087: 1, 36086: 1, 36085: 1, 36084: 1, 36083: 1, 36082: 1, 3608
1: 1, 36080: 1, 36079: 1, 36078: 1, 36077: 1, 36076: 1, 36075: 1, 3607
4: 1, 36073: 1, 36072: 1, 36071: 1, 36070: 1, 36069: 1, 36068: 1, 3606
7: 1, 36066: 1, 36065: 1, 36064: 1, 36063: 1, 36062: 1, 36061: 1, 3606
0: 1, 36059: 1, 36058: 1, 36057: 1, 36056: 1, 36055: 1, 36054: 1, 3605
3: 1, 36052: 1, 36051: 1, 36050: 1, 36049: 1, 36048: 1, 36047: 1, 3604
6: 1, 36045: 1, 36044: 1, 36043: 1, 36042: 1, 36041: 1, 36040: 1, 3603
9: 1, 36038: 1, 36037: 1, 36036: 1, 36035: 1, 36034: 1, 36033: 1, 3603
2: 1, 36031: 1, 36030: 1, 36029: 1, 36028: 1, 36027: 1, 36026: 1, 3602
5: 1, 36024: 1, 36023: 1, 36022: 1, 36021: 1, 36020: 1, 36019: 1, 3601
8: 1, 36017: 1, 36016: 1, 36015: 1, 36014: 1, 36013: 1, 36012: 1, 3601
1: 1, 36010: 1, 36009: 1, 36008: 1, 36007: 1, 36006: 1, 36005: 1, 3600

4: 1, 36003: 1, 36002: 1, 36001: 1, 36000: 1, 35999: 1, 35998: 1, 3599
7: 1, 35996: 1, 35995: 1, 35994: 1, 35993: 1, 35992: 1, 35991: 1, 3599
0: 1, 35989: 1, 35988: 1, 35987: 1, 35986: 1, 35985: 1, 35984: 1, 3598
3: 1, 35982: 1, 35981: 1, 35980: 1, 35979: 1, 35978: 1, 35977: 1, 3597
6: 1, 35975: 1, 35974: 1, 35973: 1, 35972: 1, 35971: 1, 35970: 1, 3596
9: 1, 35968: 1, 35967: 1, 35966: 1, 35965: 1, 35964: 1, 35963: 1, 3596
2: 1, 35961: 1, 35960: 1, 35959: 1, 35958: 1, 35957: 1, 35956: 1, 3595
5: 1, 35954: 1, 35953: 1, 35952: 1, 35951: 1, 35950: 1, 35949: 1, 3594
8: 1, 35947: 1, 35946: 1, 35945: 1, 35944: 1, 35943: 1, 35942: 1, 3594
1: 1, 35940: 1, 35939: 1, 35938: 1, 35937: 1, 35936: 1, 35935: 1, 3593
4: 1, 35933: 1, 35932: 1, 35931: 1, 35930: 1, 35929: 1, 35928: 1, 3592
7: 1, 35926: 1, 35925: 1, 35924: 1, 35923: 1, 35922: 1, 35921: 1, 3592
0: 1, 35919: 1, 35918: 1, 35917: 1, 35916: 1, 35915: 1, 35914: 1, 3591
3: 1, 35912: 1, 35911: 1, 35910: 1, 35909: 1, 35908: 1, 35907: 1, 3590
6: 1, 35905: 1, 35904: 1, 35903: 1, 35902: 1, 35901: 1, 35900: 1, 3589
9: 1, 35898: 1, 35897: 1, 35896: 1, 35895: 1, 35894: 1, 35893: 1, 3589
2: 1, 35891: 1, 35890: 1, 35889: 1, 35888: 1, 35887: 1, 35886: 1, 3588
5: 1, 35884: 1, 35883: 1, 35882: 1, 35881: 1, 35880: 1, 35879: 1, 3587
8: 1, 35877: 1, 35876: 1, 35875: 1, 35874: 1, 35873: 1, 35872: 1, 3587
1: 1, 35870: 1, 35869: 1, 35868: 1, 35867: 1, 35866: 1, 35865: 1, 3586
4: 1, 35863: 1, 35862: 1, 35861: 1, 35860: 1, 35859: 1, 35858: 1, 3585
7: 1, 35856: 1, 35855: 1, 35854: 1, 35853: 1, 35852: 1, 35851: 1, 3585
0: 1, 35849: 1, 35848: 1, 35847: 1, 35846: 1, 35845: 1, 35844: 1, 3584
3: 1, 35842: 1, 35841: 1, 35840: 1, 35839: 1, 35838: 1, 35837: 1, 3583
6: 1, 35835: 1, 35834: 1, 35833: 1, 35832: 1, 35831: 1, 35830: 1, 3582
9: 1, 35828: 1, 35827: 1, 35826: 1, 35825: 1, 35824: 1, 35823: 1, 3582
2: 1, 35821: 1, 35820: 1, 35819: 1, 35818: 1, 35817: 1, 35816: 1, 3581
5: 1, 35814: 1, 35813: 1, 35812: 1, 35811: 1, 35810: 1, 35809: 1, 3580
8: 1, 35807: 1, 35806: 1, 35805: 1, 35804: 1, 35803: 1, 35802: 1, 3580
1: 1, 35800: 1, 35799: 1, 35798: 1, 35797: 1, 35796: 1, 35795: 1, 3579
4: 1, 35793: 1, 35792: 1, 35791: 1, 35790: 1, 35789: 1, 35788: 1, 3578
7: 1, 35786: 1, 35785: 1, 35784: 1, 35783: 1, 35782: 1, 35781: 1, 3578
0: 1, 35779: 1, 35778: 1, 35777: 1, 35776: 1, 35775: 1, 35774: 1, 3577
3: 1, 35772: 1, 35771: 1, 35770: 1, 35769: 1, 35768: 1, 35767: 1, 3576
6: 1, 35765: 1, 35764: 1, 35763: 1, 35762: 1, 35761: 1, 35760: 1, 3575
9: 1, 35758: 1, 35757: 1, 35756: 1, 35755: 1, 35754: 1, 35753: 1, 3575
2: 1, 35751: 1, 35750: 1, 35749: 1, 35748: 1, 35747: 1, 35746: 1, 3574
5: 1, 35744: 1, 35743: 1, 35742: 1, 35741: 1, 35740: 1, 35739: 1, 3573
8: 1, 35737: 1, 35736: 1, 35735: 1, 35734: 1, 35733: 1, 35732: 1, 3573

1: 1, 35730: 1, 35729: 1, 35728: 1, 35727: 1, 35726: 1, 35725: 1, 3572
4: 1, 35723: 1, 35722: 1, 35721: 1, 35720: 1, 35719: 1, 35718: 1, 3571
7: 1, 35716: 1, 35715: 1, 35714: 1, 35713: 1, 35712: 1, 35711: 1, 3571
0: 1, 35709: 1, 35708: 1, 35707: 1, 35706: 1, 35705: 1, 35704: 1, 3570
3: 1, 35702: 1, 35701: 1, 35700: 1, 35699: 1, 35698: 1, 35697: 1, 3569
6: 1, 35695: 1, 35694: 1, 35693: 1, 35692: 1, 35691: 1, 35690: 1, 3568
9: 1, 35688: 1, 35687: 1, 35686: 1, 35685: 1, 35684: 1, 35683: 1, 3568
2: 1, 35681: 1, 35680: 1, 35679: 1, 35678: 1, 35677: 1, 35676: 1, 3567
5: 1, 35674: 1, 35673: 1, 35672: 1, 35671: 1, 35670: 1, 35669: 1, 3566
8: 1, 35667: 1, 35666: 1, 35665: 1, 35664: 1, 35663: 1, 35662: 1, 3566
1: 1, 35660: 1, 35659: 1, 35658: 1, 35657: 1, 35656: 1, 35655: 1, 3565
4: 1, 35653: 1, 35652: 1, 35651: 1, 35650: 1, 35649: 1, 35648: 1, 3564
7: 1, 35646: 1, 35645: 1, 35644: 1, 35643: 1, 35642: 1, 35641: 1, 3564
0: 1, 35639: 1, 35638: 1, 35637: 1, 35636: 1, 35635: 1, 35634: 1, 3563
3: 1, 35632: 1, 35631: 1, 35630: 1, 35629: 1, 35628: 1, 35627: 1, 3562
6: 1, 35625: 1, 35624: 1, 35623: 1, 35622: 1, 35621: 1, 35620: 1, 3561
9: 1, 35618: 1, 35617: 1, 35616: 1, 35615: 1, 35614: 1, 35613: 1, 3561
2: 1, 35611: 1, 35610: 1, 35609: 1, 35608: 1, 35607: 1, 35606: 1, 3560
5: 1, 35604: 1, 35603: 1, 35602: 1, 35601: 1, 35600: 1, 35599: 1, 3559
8: 1, 35597: 1, 35596: 1, 35595: 1, 35594: 1, 35593: 1, 35592: 1, 3559
1: 1, 35590: 1, 35589: 1, 35588: 1, 35587: 1, 35586: 1, 35585: 1, 3558
4: 1, 35583: 1, 35582: 1, 35581: 1, 35580: 1, 35579: 1, 35578: 1, 3557
7: 1, 35576: 1, 35575: 1, 35574: 1, 35573: 1, 35572: 1, 35571: 1, 3557
0: 1, 35569: 1, 35568: 1, 35567: 1, 35566: 1, 35565: 1, 35564: 1, 3556
3: 1, 35562: 1, 35561: 1, 35560: 1, 35559: 1, 35558: 1, 35557: 1, 3555
6: 1, 35555: 1, 35554: 1, 35553: 1, 35552: 1, 35551: 1, 35550: 1, 3554
9: 1, 35548: 1, 35547: 1, 35546: 1, 35545: 1, 35544: 1, 35543: 1, 3554
2: 1, 35541: 1, 35540: 1, 35539: 1, 35538: 1, 35537: 1, 35536: 1, 3553
5: 1, 35534: 1, 35533: 1, 35532: 1, 35531: 1, 35530: 1, 35529: 1, 3552
8: 1, 35527: 1, 35526: 1, 35525: 1, 35524: 1, 35523: 1, 35522: 1, 3552
1: 1, 35520: 1, 35519: 1, 35518: 1, 35517: 1, 35516: 1, 35515: 1, 3551
4: 1, 35513: 1, 35512: 1, 35511: 1, 35510: 1, 35509: 1, 35508: 1, 3550
7: 1, 35506: 1, 35505: 1, 35504: 1, 35503: 1, 35502: 1, 35501: 1, 3550
0: 1, 35499: 1, 35498: 1, 35497: 1, 35496: 1, 35495: 1, 35494: 1, 3549
3: 1, 35492: 1, 35491: 1, 35490: 1, 35489: 1, 35488: 1, 35487: 1, 3548
6: 1, 35485: 1, 35484: 1, 35483: 1, 35482: 1, 35481: 1, 35480: 1, 3547
9: 1, 35478: 1, 35477: 1, 35476: 1, 35475: 1, 35474: 1, 35473: 1, 3547
2: 1, 35471: 1, 35470: 1, 35469: 1, 35468: 1, 35467: 1, 35466: 1, 3546
5: 1, 35464: 1, 35463: 1, 35462: 1, 35461: 1, 35460: 1, 35459: 1, 3545

8: 1, 35457: 1, 35456: 1, 35455: 1, 35454: 1, 35453: 1, 35452: 1, 3545
1: 1, 35450: 1, 35449: 1, 35448: 1, 35447: 1, 35446: 1, 35445: 1, 3544
4: 1, 35443: 1, 35442: 1, 35441: 1, 35440: 1, 35439: 1, 35438: 1, 3543
7: 1, 35436: 1, 35435: 1, 35434: 1, 35433: 1, 35432: 1, 35431: 1, 3543
0: 1, 35429: 1, 35428: 1, 35427: 1, 35426: 1, 35425: 1, 35424: 1, 3542
3: 1, 35422: 1, 35421: 1, 35420: 1, 35419: 1, 35418: 1, 35417: 1, 3541
6: 1, 35415: 1, 35414: 1, 35413: 1, 35412: 1, 35411: 1, 35410: 1, 3540
9: 1, 35408: 1, 35407: 1, 35406: 1, 35405: 1, 35404: 1, 35403: 1, 3540
2: 1, 35401: 1, 35400: 1, 35399: 1, 35398: 1, 35397: 1, 35396: 1, 3539
5: 1, 35394: 1, 35393: 1, 35392: 1, 35391: 1, 35390: 1, 35389: 1, 3538
8: 1, 35387: 1, 35386: 1, 35385: 1, 35384: 1, 35383: 1, 35382: 1, 3538
1: 1, 35380: 1, 35379: 1, 35378: 1, 35377: 1, 35376: 1, 35375: 1, 3537
4: 1, 35373: 1, 35372: 1, 35371: 1, 35370: 1, 35369: 1, 35368: 1, 3536
7: 1, 35366: 1, 35365: 1, 35364: 1, 35363: 1, 35362: 1, 35361: 1, 3536
0: 1, 35359: 1, 35358: 1, 35357: 1, 35356: 1, 35355: 1, 35354: 1, 3535
3: 1, 35352: 1, 35351: 1, 35350: 1, 35349: 1, 35348: 1, 35347: 1, 3534
6: 1, 35345: 1, 35344: 1, 35343: 1, 35342: 1, 35341: 1, 35340: 1, 3533
9: 1, 35338: 1, 35337: 1, 35336: 1, 35335: 1, 35334: 1, 35333: 1, 3533
2: 1, 35331: 1, 35330: 1, 35329: 1, 35328: 1, 35327: 1, 35326: 1, 3532
5: 1, 35324: 1, 35323: 1, 35322: 1, 35321: 1, 35320: 1, 35319: 1, 3531
8: 1, 35317: 1, 35316: 1, 35315: 1, 35314: 1, 35313: 1, 35312: 1, 3531
1: 1, 35310: 1, 35309: 1, 35308: 1, 35307: 1, 35306: 1, 35305: 1, 3530
4: 1, 35303: 1, 35302: 1, 35301: 1, 35300: 1, 35299: 1, 35298: 1, 3529
7: 1, 35296: 1, 35295: 1, 35294: 1, 35293: 1, 35292: 1, 35291: 1, 3529
0: 1, 35289: 1, 35288: 1, 35287: 1, 35286: 1, 35285: 1, 35284: 1, 3528
3: 1, 35282: 1, 35281: 1, 35280: 1, 35279: 1, 35278: 1, 35277: 1, 3527
6: 1, 35275: 1, 35274: 1, 35273: 1, 35272: 1, 35271: 1, 35270: 1, 3526
9: 1, 35268: 1, 35267: 1, 35266: 1, 35265: 1, 35264: 1, 35263: 1, 3526
2: 1, 35261: 1, 35260: 1, 35259: 1, 35258: 1, 35257: 1, 35256: 1, 3525
5: 1, 35254: 1, 35253: 1, 35252: 1, 35251: 1, 35250: 1, 35249: 1, 3524
8: 1, 35247: 1, 35246: 1, 35245: 1, 35244: 1, 35243: 1, 35242: 1, 3524
1: 1, 35240: 1, 35239: 1, 35238: 1, 35237: 1, 35236: 1, 35235: 1, 3523
4: 1, 35233: 1, 35232: 1, 35231: 1, 35230: 1, 35229: 1, 35228: 1, 3522
7: 1, 35226: 1, 35225: 1, 35224: 1, 35223: 1, 35222: 1, 35221: 1, 3522
0: 1, 35219: 1, 35218: 1, 35217: 1, 35216: 1, 35215: 1, 35214: 1, 3521
3: 1, 35212: 1, 35211: 1, 35210: 1, 35209: 1, 35208: 1, 35207: 1, 3520
6: 1, 35205: 1, 35204: 1, 35203: 1, 35202: 1, 35201: 1, 35200: 1, 3519
9: 1, 35198: 1, 35197: 1, 35196: 1, 35195: 1, 35194: 1, 35193: 1, 3519
2: 1, 35191: 1, 35190: 1, 35189: 1, 35188: 1, 35187: 1, 35186: 1, 3518

5: 1, 35184: 1, 35183: 1, 35182: 1, 35181: 1, 35180: 1, 35179: 1, 3517
8: 1, 35177: 1, 35176: 1, 35175: 1, 35174: 1, 35173: 1, 35172: 1, 3517
1: 1, 35170: 1, 35169: 1, 35168: 1, 35167: 1, 35166: 1, 35165: 1, 3516
4: 1, 35163: 1, 35162: 1, 35161: 1, 35160: 1, 35159: 1, 35158: 1, 3515
7: 1, 35156: 1, 35155: 1, 35154: 1, 35153: 1, 35152: 1, 35151: 1, 3515
0: 1, 35149: 1, 35148: 1, 35147: 1, 35146: 1, 35145: 1, 35144: 1, 3514
3: 1, 35142: 1, 35141: 1, 35140: 1, 35139: 1, 35138: 1, 35137: 1, 3513
6: 1, 35135: 1, 35134: 1, 35133: 1, 35132: 1, 35131: 1, 35130: 1, 3512
9: 1, 35128: 1, 35127: 1, 35126: 1, 35125: 1, 35124: 1, 35123: 1, 3512
2: 1, 35121: 1, 35120: 1, 35119: 1, 35118: 1, 35117: 1, 35116: 1, 3511
5: 1, 35114: 1, 35113: 1, 35112: 1, 35111: 1, 35110: 1, 35109: 1, 3510
8: 1, 35107: 1, 35106: 1, 35105: 1, 35104: 1, 35103: 1, 35102: 1, 3510
1: 1, 35100: 1, 35099: 1, 35098: 1, 35097: 1, 35096: 1, 35095: 1, 3509
4: 1, 35093: 1, 35092: 1, 35091: 1, 35090: 1, 35089: 1, 35088: 1, 3508
7: 1, 35086: 1, 35085: 1, 35084: 1, 35083: 1, 35082: 1, 35081: 1, 3508
0: 1, 35079: 1, 35078: 1, 35077: 1, 35076: 1, 35075: 1, 35074: 1, 3507
3: 1, 35072: 1, 35071: 1, 35070: 1, 35069: 1, 35068: 1, 35067: 1, 3506
6: 1, 35065: 1, 35064: 1, 35063: 1, 35062: 1, 35061: 1, 35060: 1, 3505
9: 1, 35058: 1, 35057: 1, 35056: 1, 35055: 1, 35054: 1, 35053: 1, 3505
2: 1, 35051: 1, 35050: 1, 35049: 1, 35048: 1, 35047: 1, 35046: 1, 3504
5: 1, 35044: 1, 35043: 1, 35042: 1, 35041: 1, 35040: 1, 35039: 1, 3503
8: 1, 35037: 1, 35036: 1, 35035: 1, 35034: 1, 35033: 1, 35032: 1, 3503
1: 1, 35030: 1, 35029: 1, 35028: 1, 35027: 1, 35026: 1, 35025: 1, 3502
4: 1, 35023: 1, 35022: 1, 35021: 1, 35020: 1, 35019: 1, 35018: 1, 3501
7: 1, 35016: 1, 35015: 1, 35014: 1, 35013: 1, 35012: 1, 35011: 1, 3501
0: 1, 35009: 1, 35008: 1, 35007: 1, 35006: 1, 35005: 1, 35004: 1, 3500
3: 1, 35002: 1, 35001: 1, 35000: 1, 34999: 1, 34998: 1, 34997: 1, 3499
6: 1, 34995: 1, 34994: 1, 34993: 1, 34992: 1, 34991: 1, 34990: 1, 3498
9: 1, 34988: 1, 34987: 1, 34986: 1, 34985: 1, 34984: 1, 34983: 1, 3498
2: 1, 34981: 1, 34980: 1, 34979: 1, 34978: 1, 34977: 1, 34976: 1, 3497
5: 1, 34974: 1, 34973: 1, 34972: 1, 34971: 1, 34970: 1, 34969: 1, 3496
8: 1, 34967: 1, 34966: 1, 34965: 1, 34964: 1, 34963: 1, 34962: 1, 3496
1: 1, 34960: 1, 34959: 1, 34958: 1, 34957: 1, 34956: 1, 34955: 1, 3495
4: 1, 34953: 1, 34952: 1, 34951: 1, 34950: 1, 34949: 1, 34948: 1, 3494
7: 1, 34946: 1, 34945: 1, 34944: 1, 34943: 1, 34942: 1, 34941: 1, 3494
0: 1, 34939: 1, 34938: 1, 34937: 1, 34936: 1, 34935: 1, 34934: 1, 3493
3: 1, 34932: 1, 34931: 1, 34930: 1, 34929: 1, 34928: 1, 34927: 1, 3492
6: 1, 34925: 1, 34924: 1, 34923: 1, 34922: 1, 34921: 1, 34920: 1, 3491
9: 1, 34918: 1, 34917: 1, 34916: 1, 34915: 1, 34914: 1, 34913: 1, 3491

2: 1, 34911: 1, 34910: 1, 34909: 1, 34908: 1, 34907: 1, 34906: 1, 3490
5: 1, 34904: 1, 34903: 1, 34902: 1, 34901: 1, 34900: 1, 34899: 1, 3489
8: 1, 34897: 1, 34896: 1, 34895: 1, 34894: 1, 34893: 1, 34892: 1, 3489
1: 1, 34890: 1, 34889: 1, 34888: 1, 34887: 1, 34886: 1, 34885: 1, 3488
4: 1, 34883: 1, 34882: 1, 34881: 1, 34880: 1, 34879: 1, 34878: 1, 3487
7: 1, 34876: 1, 34875: 1, 34874: 1, 34873: 1, 34872: 1, 34871: 1, 3487
0: 1, 34869: 1, 34868: 1, 34867: 1, 34866: 1, 34865: 1, 34864: 1, 3486
3: 1, 34862: 1, 34861: 1, 34860: 1, 34859: 1, 34858: 1, 34857: 1, 3485
6: 1, 34855: 1, 34854: 1, 34853: 1, 34852: 1, 34851: 1, 34850: 1, 3484
9: 1, 34848: 1, 34847: 1, 34846: 1, 34845: 1, 34844: 1, 34843: 1, 3484
2: 1, 34841: 1, 34840: 1, 34839: 1, 34838: 1, 34837: 1, 34836: 1, 3483
5: 1, 34834: 1, 34833: 1, 34832: 1, 34831: 1, 34830: 1, 34829: 1, 3482
8: 1, 34827: 1, 34826: 1, 34825: 1, 34824: 1, 34823: 1, 34822: 1, 3482
1: 1, 34820: 1, 34819: 1, 34818: 1, 34817: 1, 34816: 1, 34815: 1, 3481
4: 1, 34813: 1, 34812: 1, 34811: 1, 34810: 1, 34809: 1, 34808: 1, 3480
7: 1, 34806: 1, 34805: 1, 34804: 1, 34803: 1, 34802: 1, 34801: 1, 3480
0: 1, 34799: 1, 34798: 1, 34797: 1, 34796: 1, 34795: 1, 34794: 1, 3479
3: 1, 34792: 1, 34791: 1, 34790: 1, 34789: 1, 34788: 1, 34787: 1, 3478
6: 1, 34785: 1, 34784: 1, 34783: 1, 34782: 1, 34781: 1, 34780: 1, 3477
9: 1, 34778: 1, 34777: 1, 34776: 1, 34775: 1, 34774: 1, 34773: 1, 3477
2: 1, 34771: 1, 34770: 1, 34769: 1, 34768: 1, 34767: 1, 34766: 1, 3476
5: 1, 34764: 1, 34763: 1, 34762: 1, 34761: 1, 34760: 1, 34759: 1, 3475
8: 1, 34757: 1, 34756: 1, 34755: 1, 34754: 1, 34753: 1, 34752: 1, 3475
1: 1, 34750: 1, 34749: 1, 34748: 1, 34747: 1, 34746: 1, 34745: 1, 3474
4: 1, 34743: 1, 34742: 1, 34741: 1, 34740: 1, 34739: 1, 34738: 1, 3473
7: 1, 34736: 1, 34735: 1, 34734: 1, 34733: 1, 34732: 1, 34731: 1, 3473
0: 1, 34729: 1, 34728: 1, 34727: 1, 34726: 1, 34725: 1, 34724: 1, 3472
3: 1, 34722: 1, 34721: 1, 34720: 1, 34719: 1, 34718: 1, 34717: 1, 3471
6: 1, 34715: 1, 34714: 1, 34713: 1, 34712: 1, 34711: 1, 34710: 1, 3470
9: 1, 34708: 1, 34707: 1, 34706: 1, 34705: 1, 34704: 1, 34703: 1, 3470
2: 1, 34701: 1, 34700: 1, 34699: 1, 34698: 1, 34697: 1, 34696: 1, 3469
5: 1, 34694: 1, 34693: 1, 34692: 1, 34691: 1, 34690: 1, 34689: 1, 3468
8: 1, 34687: 1, 34686: 1, 34685: 1, 34684: 1, 34683: 1, 34682: 1, 3468
1: 1, 34680: 1, 34679: 1, 34678: 1, 34677: 1, 34676: 1, 34675: 1, 3467
4: 1, 34673: 1, 34672: 1, 34671: 1, 34670: 1, 34669: 1, 34668: 1, 3466
7: 1, 34666: 1, 34665: 1, 34664: 1, 34663: 1, 34662: 1, 34661: 1, 3466
0: 1, 34659: 1, 34658: 1, 34657: 1, 34656: 1, 34655: 1, 34654: 1, 3465
3: 1, 34652: 1, 34651: 1, 34650: 1, 34649: 1, 34648: 1, 34647: 1, 3464
6: 1, 34645: 1, 34644: 1, 34643: 1, 34642: 1, 34641: 1, 34640: 1, 3463

9: 1, 34638: 1, 34637: 1, 34636: 1, 34635: 1, 34634: 1, 34633: 1, 3463
2: 1, 34631: 1, 34630: 1, 34629: 1, 34628: 1, 34627: 1, 34626: 1, 3462
5: 1, 34624: 1, 34623: 1, 34622: 1, 34621: 1, 34620: 1, 34619: 1, 3461
8: 1, 34617: 1, 34616: 1, 34615: 1, 34614: 1, 34613: 1, 34612: 1, 3461
1: 1, 34610: 1, 34609: 1, 34608: 1, 34607: 1, 34606: 1, 34605: 1, 3460
4: 1, 34603: 1, 34602: 1, 34601: 1, 34600: 1, 34599: 1, 34598: 1, 3459
7: 1, 34596: 1, 34595: 1, 34594: 1, 34593: 1, 34592: 1, 34591: 1, 3459
0: 1, 34589: 1, 34588: 1, 34587: 1, 34586: 1, 34585: 1, 34584: 1, 3458
3: 1, 34582: 1, 34581: 1, 34580: 1, 34579: 1, 34578: 1, 34577: 1, 3457
6: 1, 34575: 1, 34574: 1, 34573: 1, 34572: 1, 34571: 1, 34570: 1, 3456
9: 1, 34568: 1, 34567: 1, 34566: 1, 34565: 1, 34564: 1, 34563: 1, 3456
2: 1, 34561: 1, 34560: 1, 34559: 1, 34558: 1, 34557: 1, 34556: 1, 3455
5: 1, 34554: 1, 34553: 1, 34552: 1, 34551: 1, 34550: 1, 34549: 1, 3454
8: 1, 34547: 1, 34546: 1, 34545: 1, 34544: 1, 34543: 1, 34542: 1, 3454
1: 1, 34540: 1, 34539: 1, 34538: 1, 34537: 1, 34536: 1, 34535: 1, 3453
4: 1, 34533: 1, 34532: 1, 34531: 1, 34530: 1, 34529: 1, 34528: 1, 3452
7: 1, 34526: 1, 34525: 1, 34524: 1, 34523: 1, 34522: 1, 34521: 1, 3452
0: 1, 34519: 1, 34518: 1, 34517: 1, 34516: 1, 34515: 1, 34514: 1, 3451
3: 1, 34512: 1, 34511: 1, 34510: 1, 34509: 1, 34508: 1, 34507: 1, 3450
6: 1, 34505: 1, 34504: 1, 34503: 1, 34502: 1, 34501: 1, 34500: 1, 3449
9: 1, 34498: 1, 34497: 1, 34496: 1, 34495: 1, 34494: 1, 34493: 1, 3449
2: 1, 34491: 1, 34490: 1, 34489: 1, 34488: 1, 34487: 1, 34486: 1, 3448
5: 1, 34484: 1, 34483: 1, 34482: 1, 34481: 1, 34480: 1, 34479: 1, 3447
8: 1, 34477: 1, 34476: 1, 34475: 1, 34474: 1, 34473: 1, 34472: 1, 3447
1: 1, 34470: 1, 34469: 1, 34468: 1, 34467: 1, 34466: 1, 34465: 1, 3446
4: 1, 34463: 1, 34462: 1, 34461: 1, 34460: 1, 34459: 1, 34458: 1, 3445
7: 1, 34456: 1, 34455: 1, 34454: 1, 34453: 1, 34452: 1, 34451: 1, 3445
0: 1, 34449: 1, 34448: 1, 34447: 1, 34446: 1, 34445: 1, 34444: 1, 3444
3: 1, 34442: 1, 34441: 1, 34440: 1, 34439: 1, 34438: 1, 34437: 1, 3443
6: 1, 34435: 1, 34434: 1, 34433: 1, 34432: 1, 34431: 1, 34430: 1, 3442
9: 1, 34428: 1, 34427: 1, 34426: 1, 34425: 1, 34424: 1, 34423: 1, 3442
2: 1, 34421: 1, 34420: 1, 34419: 1, 34418: 1, 34417: 1, 34416: 1, 3441
5: 1, 34414: 1, 34413: 1, 34412: 1, 34411: 1, 34410: 1, 34409: 1, 3440
8: 1, 34407: 1, 34406: 1, 34405: 1, 34404: 1, 34403: 1, 34402: 1, 3440
1: 1, 34400: 1, 34399: 1, 34398: 1, 34397: 1, 34396: 1, 34395: 1, 3439
4: 1, 34393: 1, 34392: 1, 34391: 1, 34390: 1, 34389: 1, 34388: 1, 3438
7: 1, 34386: 1, 34385: 1, 34384: 1, 34383: 1, 34382: 1, 34381: 1, 3438
0: 1, 34379: 1, 34378: 1, 34377: 1, 34376: 1, 34375: 1, 34374: 1, 3437
3: 1, 34372: 1, 34371: 1, 34370: 1, 34369: 1, 34368: 1, 34367: 1, 3436

6: 1, 34365: 1, 34364: 1, 34363: 1, 34362: 1, 34361: 1, 34360: 1, 3435
9: 1, 34358: 1, 34357: 1, 34356: 1, 34355: 1, 34354: 1, 34353: 1, 3435
2: 1, 34351: 1, 34350: 1, 34349: 1, 34348: 1, 34347: 1, 34346: 1, 3434
5: 1, 34344: 1, 34343: 1, 34342: 1, 34341: 1, 34340: 1, 34339: 1, 3433
8: 1, 34337: 1, 34336: 1, 34335: 1, 34334: 1, 34333: 1, 34332: 1, 3433
1: 1, 34330: 1, 34329: 1, 34328: 1, 34327: 1, 34326: 1, 34325: 1, 3432
4: 1, 34323: 1, 34322: 1, 34321: 1, 34320: 1, 34319: 1, 34318: 1, 3431
7: 1, 34316: 1, 34315: 1, 34314: 1, 34313: 1, 34312: 1, 34311: 1, 3431
0: 1, 34309: 1, 34308: 1, 34307: 1, 34306: 1, 34305: 1, 34304: 1, 3430
3: 1, 34302: 1, 34301: 1, 34300: 1, 34299: 1, 34298: 1, 34297: 1, 3429
6: 1, 34295: 1, 34294: 1, 34293: 1, 34292: 1, 34291: 1, 34290: 1, 3428
9: 1, 34288: 1, 34287: 1, 34286: 1, 34285: 1, 34284: 1, 34283: 1, 3428
2: 1, 34281: 1, 34280: 1, 34279: 1, 34278: 1, 34277: 1, 34276: 1, 3427
5: 1, 34274: 1, 34273: 1, 34272: 1, 34271: 1, 34270: 1, 34269: 1, 3426
8: 1, 34267: 1, 34266: 1, 34265: 1, 34264: 1, 34263: 1, 34262: 1, 3426
1: 1, 34260: 1, 34259: 1, 34258: 1, 34257: 1, 34256: 1, 34255: 1, 3425
4: 1, 34253: 1, 34252: 1, 34251: 1, 34250: 1, 34249: 1, 34248: 1, 3424
7: 1, 34246: 1, 34245: 1, 34244: 1, 34243: 1, 34242: 1, 34241: 1, 3424
0: 1, 34239: 1, 34238: 1, 34237: 1, 34236: 1, 34235: 1, 34234: 1, 3423
3: 1, 34232: 1, 34231: 1, 34230: 1, 34229: 1, 34228: 1, 34227: 1, 3422
6: 1, 34225: 1, 34224: 1, 34223: 1, 34222: 1, 34221: 1, 34220: 1, 3421
9: 1, 34218: 1, 34217: 1, 34216: 1, 34215: 1, 34214: 1, 34213: 1, 3421
2: 1, 34211: 1, 34210: 1, 34209: 1, 34208: 1, 34207: 1, 34206: 1, 3420
5: 1, 34204: 1, 34203: 1, 34202: 1, 34201: 1, 34200: 1, 34199: 1, 3419
8: 1, 34197: 1, 34196: 1, 34195: 1, 34194: 1, 34193: 1, 34192: 1, 3419
1: 1, 34190: 1, 34189: 1, 34188: 1, 34187: 1, 34186: 1, 34185: 1, 3418
4: 1, 34183: 1, 34182: 1, 34181: 1, 34180: 1, 34179: 1, 34178: 1, 3417
7: 1, 34176: 1, 34175: 1, 34174: 1, 34173: 1, 34172: 1, 34171: 1, 3417
0: 1, 34169: 1, 34168: 1, 34167: 1, 34166: 1, 34165: 1, 34164: 1, 3416
3: 1, 34162: 1, 34161: 1, 34160: 1, 34159: 1, 34158: 1, 34157: 1, 3415
6: 1, 34155: 1, 34154: 1, 34153: 1, 34152: 1, 34151: 1, 34150: 1, 3414
9: 1, 34148: 1, 34147: 1, 34146: 1, 34145: 1, 34144: 1, 34143: 1, 3414
2: 1, 34141: 1, 34140: 1, 34139: 1, 34138: 1, 34137: 1, 34136: 1, 3413
5: 1, 34134: 1, 34133: 1, 34132: 1, 34131: 1, 34130: 1, 34129: 1, 3412
8: 1, 34127: 1, 34126: 1, 34125: 1, 34124: 1, 34123: 1, 34122: 1, 3412
1: 1, 34120: 1, 34119: 1, 34118: 1, 34117: 1, 34116: 1, 34115: 1, 3411
4: 1, 34113: 1, 34112: 1, 34111: 1, 34110: 1, 34109: 1, 34108: 1, 3410
7: 1, 34106: 1, 34105: 1, 34104: 1, 34103: 1, 34102: 1, 34101: 1, 3410
0: 1, 34099: 1, 34098: 1, 34097: 1, 34096: 1, 34095: 1, 34094: 1, 3409

3: 1, 34092: 1, 34091: 1, 34090: 1, 34089: 1, 34088: 1, 34087: 1, 3408
6: 1, 34085: 1, 34084: 1, 34083: 1, 34082: 1, 34081: 1, 34080: 1, 3407
9: 1, 34078: 1, 34077: 1, 34076: 1, 34075: 1, 34074: 1, 34073: 1, 3407
2: 1, 34071: 1, 34070: 1, 34069: 1, 34068: 1, 34067: 1, 34066: 1, 3406
5: 1, 34064: 1, 34063: 1, 34062: 1, 34061: 1, 34060: 1, 34059: 1, 3405
8: 1, 34057: 1, 34056: 1, 34055: 1, 34054: 1, 34053: 1, 34052: 1, 3405
1: 1, 34050: 1, 34049: 1, 34048: 1, 34047: 1, 34046: 1, 34045: 1, 3404
4: 1, 34043: 1, 34042: 1, 34041: 1, 34040: 1, 34039: 1, 34038: 1, 3403
7: 1, 34036: 1, 34035: 1, 34034: 1, 34033: 1, 34032: 1, 34031: 1, 3403
0: 1, 34029: 1, 34028: 1, 34027: 1, 34026: 1, 34025: 1, 34024: 1, 3402
3: 1, 34022: 1, 34021: 1, 34020: 1, 34019: 1, 34018: 1, 34017: 1, 3401
6: 1, 34015: 1, 34014: 1, 34013: 1, 34012: 1, 34011: 1, 34010: 1, 3400
9: 1, 34008: 1, 34007: 1, 34006: 1, 34005: 1, 34004: 1, 34003: 1, 3400
2: 1, 34001: 1, 34000: 1, 33999: 1, 33998: 1, 33997: 1, 33996: 1, 3399
5: 1, 33994: 1, 33993: 1, 33992: 1, 33991: 1, 33990: 1, 33989: 1, 3398
8: 1, 33987: 1, 33986: 1, 33985: 1, 33984: 1, 33983: 1, 33982: 1, 3398
1: 1, 33980: 1, 33979: 1, 33978: 1, 33977: 1, 33976: 1, 33975: 1, 3397
4: 1, 33973: 1, 33972: 1, 33971: 1, 33970: 1, 33969: 1, 33968: 1, 3396
7: 1, 33966: 1, 33965: 1, 33964: 1, 33963: 1, 33962: 1, 33961: 1, 3396
0: 1, 33959: 1, 33958: 1, 33957: 1, 33956: 1, 33955: 1, 33954: 1, 3395
3: 1, 33952: 1, 33951: 1, 33950: 1, 33949: 1, 33948: 1, 33947: 1, 3394
6: 1, 33945: 1, 33944: 1, 33943: 1, 33942: 1, 33941: 1, 33940: 1, 3393
9: 1, 33938: 1, 33937: 1, 33936: 1, 33935: 1, 33934: 1, 33933: 1, 3393
2: 1, 33931: 1, 33930: 1, 33929: 1, 33928: 1, 33927: 1, 33926: 1, 3392
5: 1, 33924: 1, 33923: 1, 33922: 1, 33921: 1, 33920: 1, 33919: 1, 3391
8: 1, 33917: 1, 33916: 1, 33915: 1, 33914: 1, 33913: 1, 33912: 1, 3391
1: 1, 33910: 1, 33909: 1, 33908: 1, 33907: 1, 33906: 1, 33905: 1, 3390
4: 1, 33903: 1, 33902: 1, 33901: 1, 33900: 1, 33899: 1, 33898: 1, 3389
7: 1, 33896: 1, 33895: 1, 33894: 1, 33893: 1, 33892: 1, 33891: 1, 3389
0: 1, 33889: 1, 33888: 1, 33887: 1, 33886: 1, 33885: 1, 33884: 1, 3388
3: 1, 33882: 1, 33881: 1, 33880: 1, 33879: 1, 33878: 1, 33877: 1, 3387
6: 1, 33875: 1, 33874: 1, 33873: 1, 33872: 1, 33871: 1, 33870: 1, 3386
9: 1, 33868: 1, 33867: 1, 33866: 1, 33865: 1, 33864: 1, 33863: 1, 3386
2: 1, 33861: 1, 33860: 1, 33859: 1, 33858: 1, 33857: 1, 33856: 1, 3385
5: 1, 33854: 1, 33853: 1, 33852: 1, 33851: 1, 33850: 1, 33849: 1, 3384
8: 1, 33847: 1, 33846: 1, 33845: 1, 33844: 1, 33843: 1, 33842: 1, 3384
1: 1, 33840: 1, 33839: 1, 33838: 1, 33837: 1, 33836: 1, 33835: 1, 3383
4: 1, 33833: 1, 33832: 1, 33831: 1, 33830: 1, 33829: 1, 33828: 1, 3382
7: 1, 33826: 1, 33825: 1, 33824: 1, 33823: 1, 33822: 1, 33821: 1, 3382

0: 1, 33819: 1, 33818: 1, 33817: 1, 33816: 1, 33815: 1, 33814: 1, 3381
3: 1, 33812: 1, 33811: 1, 33810: 1, 33809: 1, 33808: 1, 33807: 1, 3380
6: 1, 33805: 1, 33804: 1, 33803: 1, 33802: 1, 33801: 1, 33800: 1, 3379
9: 1, 33798: 1, 33797: 1, 33796: 1, 33795: 1, 33794: 1, 33793: 1, 3379
2: 1, 33791: 1, 33790: 1, 33789: 1, 33788: 1, 33787: 1, 33786: 1, 3378
5: 1, 33784: 1, 33783: 1, 33782: 1, 33781: 1, 33780: 1, 33779: 1, 3377
8: 1, 33777: 1, 33776: 1, 33775: 1, 33774: 1, 33773: 1, 33772: 1, 3377
1: 1, 33770: 1, 33769: 1, 33768: 1, 33767: 1, 33766: 1, 33765: 1, 3376
4: 1, 33763: 1, 33762: 1, 33761: 1, 33760: 1, 33759: 1, 33758: 1, 3375
7: 1, 33756: 1, 33755: 1, 33754: 1, 33753: 1, 33752: 1, 33751: 1, 3375
0: 1, 33749: 1, 33748: 1, 33747: 1, 33746: 1, 33745: 1, 33744: 1, 3374
3: 1, 33742: 1, 33741: 1, 33740: 1, 33739: 1, 33738: 1, 33737: 1, 3373
6: 1, 33735: 1, 33734: 1, 33733: 1, 33732: 1, 33731: 1, 33730: 1, 3372
9: 1, 33728: 1, 33727: 1, 33726: 1, 33725: 1, 33724: 1, 33723: 1, 3372
2: 1, 33721: 1, 33720: 1, 33719: 1, 33718: 1, 33717: 1, 33716: 1, 3371
5: 1, 33714: 1, 33713: 1, 33712: 1, 33711: 1, 33710: 1, 33709: 1, 3370
8: 1, 33707: 1, 33706: 1, 33705: 1, 33704: 1, 33703: 1, 33702: 1, 3370
1: 1, 33700: 1, 33699: 1, 33698: 1, 33697: 1, 33696: 1, 33695: 1, 3369
4: 1, 33693: 1, 33692: 1, 33691: 1, 33690: 1, 33689: 1, 33688: 1, 3368
7: 1, 33686: 1, 33685: 1, 33684: 1, 33683: 1, 33682: 1, 33681: 1, 3368
0: 1, 33679: 1, 33678: 1, 33677: 1, 33676: 1, 33675: 1, 33674: 1, 3367
3: 1, 33672: 1, 33671: 1, 33670: 1, 33669: 1, 33668: 1, 33667: 1, 3366
6: 1, 33665: 1, 33664: 1, 33663: 1, 33662: 1, 33661: 1, 33660: 1, 3365
9: 1, 33658: 1, 33657: 1, 33656: 1, 33655: 1, 33654: 1, 33653: 1, 3365
2: 1, 33651: 1, 33650: 1, 33649: 1, 33648: 1, 33647: 1, 33646: 1, 3364
5: 1, 33644: 1, 33643: 1, 33642: 1, 33641: 1, 33640: 1, 33639: 1, 3363
8: 1, 33637: 1, 33636: 1, 33635: 1, 33634: 1, 33633: 1, 33632: 1, 3363
1: 1, 33630: 1, 33629: 1, 33628: 1, 33627: 1, 33626: 1, 33625: 1, 3362
4: 1, 33623: 1, 33622: 1, 33621: 1, 33620: 1, 33619: 1, 33618: 1, 3361
7: 1, 33616: 1, 33615: 1, 33614: 1, 33613: 1, 33612: 1, 33611: 1, 3361
0: 1, 33609: 1, 33608: 1, 33607: 1, 33606: 1, 33605: 1, 33604: 1, 3360
3: 1, 33602: 1, 33601: 1, 33600: 1, 33599: 1, 33598: 1, 33597: 1, 3359
6: 1, 33595: 1, 33594: 1, 33593: 1, 33592: 1, 33591: 1, 33590: 1, 3358
9: 1, 33588: 1, 33587: 1, 33586: 1, 33585: 1, 33584: 1, 33583: 1, 3358
2: 1, 33581: 1, 33580: 1, 33579: 1, 33578: 1, 33577: 1, 33576: 1, 3357
5: 1, 33574: 1, 33573: 1, 33572: 1, 33571: 1, 33570: 1, 33569: 1, 3356
8: 1, 33567: 1, 33566: 1, 33565: 1, 33564: 1, 33563: 1, 33562: 1, 3356
1: 1, 33560: 1, 33559: 1, 33558: 1, 33557: 1, 33556: 1, 33555: 1, 3355
4: 1, 33553: 1, 33552: 1, 33551: 1, 33550: 1, 33549: 1, 33548: 1, 3354

7: 1, 33546: 1, 33545: 1, 33544: 1, 33543: 1, 33542: 1, 33541: 1, 3354
0: 1, 33539: 1, 33538: 1, 33537: 1, 33536: 1, 33535: 1, 33534: 1, 3353
3: 1, 33532: 1, 33531: 1, 33530: 1, 33529: 1, 33528: 1, 33527: 1, 3352
6: 1, 33525: 1, 33524: 1, 33523: 1, 33522: 1, 33521: 1, 33520: 1, 3351
9: 1, 33518: 1, 33517: 1, 33516: 1, 33515: 1, 33514: 1, 33513: 1, 3351
2: 1, 33511: 1, 33510: 1, 33509: 1, 33508: 1, 33507: 1, 33506: 1, 3350
5: 1, 33504: 1, 33503: 1, 33502: 1, 33501: 1, 33500: 1, 33499: 1, 3349
8: 1, 33497: 1, 33496: 1, 33495: 1, 33494: 1, 33493: 1, 33492: 1, 3349
1: 1, 33490: 1, 33489: 1, 33488: 1, 33487: 1, 33486: 1, 33485: 1, 3348
4: 1, 33483: 1, 33482: 1, 33481: 1, 33480: 1, 33479: 1, 33478: 1, 3347
7: 1, 33476: 1, 33475: 1, 33474: 1, 33473: 1, 33472: 1, 33471: 1, 3347
0: 1, 33469: 1, 33468: 1, 33467: 1, 33466: 1, 33465: 1, 33464: 1, 3346
3: 1, 33462: 1, 33461: 1, 33460: 1, 33459: 1, 33458: 1, 33457: 1, 3345
6: 1, 33455: 1, 33454: 1, 33453: 1, 33452: 1, 33451: 1, 33450: 1, 3344
9: 1, 33448: 1, 33447: 1, 33446: 1, 33445: 1, 33444: 1, 33443: 1, 3344
2: 1, 33441: 1, 33440: 1, 33439: 1, 33438: 1, 33437: 1, 33436: 1, 3343
5: 1, 33434: 1, 33433: 1, 33432: 1, 33431: 1, 33430: 1, 33429: 1, 3342
8: 1, 33427: 1, 33426: 1, 33425: 1, 33424: 1, 33423: 1, 33422: 1, 3342
1: 1, 33420: 1, 33419: 1, 33418: 1, 33417: 1, 33416: 1, 33415: 1, 3341
4: 1, 33413: 1, 33412: 1, 33411: 1, 33410: 1, 33409: 1, 33408: 1, 3340
7: 1, 33406: 1, 33405: 1, 33404: 1, 33403: 1, 33402: 1, 33401: 1, 3340
0: 1, 33399: 1, 33398: 1, 33397: 1, 33396: 1, 33395: 1, 33394: 1, 3339
3: 1, 33392: 1, 33391: 1, 33390: 1, 33389: 1, 33388: 1, 33387: 1, 3338
6: 1, 33385: 1, 33384: 1, 33383: 1, 33382: 1, 33381: 1, 33380: 1, 3337
9: 1, 33378: 1, 33377: 1, 33376: 1, 33375: 1, 33374: 1, 33373: 1, 3337
2: 1, 33371: 1, 33370: 1, 33369: 1, 33368: 1, 33367: 1, 33366: 1, 3336
5: 1, 33364: 1, 33363: 1, 33362: 1, 33361: 1, 33360: 1, 33359: 1, 3335
8: 1, 33357: 1, 33356: 1, 33355: 1, 33354: 1, 33353: 1, 33352: 1, 3335
1: 1, 33350: 1, 33349: 1, 33348: 1, 33347: 1, 33346: 1, 33345: 1, 3334
4: 1, 33343: 1, 33342: 1, 33341: 1, 33340: 1, 33339: 1, 33338: 1, 3333
7: 1, 33336: 1, 33335: 1, 33334: 1, 33333: 1, 33332: 1, 33331: 1, 3333
0: 1, 33329: 1, 33328: 1, 33327: 1, 33326: 1, 33325: 1, 33324: 1, 3332
3: 1, 33322: 1, 33321: 1, 33320: 1, 33319: 1, 33318: 1, 33317: 1, 3331
6: 1, 33315: 1, 33314: 1, 33313: 1, 33312: 1, 33311: 1, 33310: 1, 3330
9: 1, 33308: 1, 33307: 1, 33306: 1, 33305: 1, 33304: 1, 33303: 1, 3330
2: 1, 33301: 1, 33300: 1, 33299: 1, 33298: 1, 33297: 1, 33296: 1, 3329
5: 1, 33294: 1, 33293: 1, 33292: 1, 33291: 1, 33290: 1, 33289: 1, 3328
8: 1, 33287: 1, 33286: 1, 33285: 1, 33284: 1, 33283: 1, 33282: 1, 3328
1: 1, 33280: 1, 33279: 1, 33278: 1, 33277: 1, 33276: 1, 33275: 1, 3327

4: 1, 33273: 1, 33272: 1, 33271: 1, 33270: 1, 33269: 1, 33268: 1, 3326
7: 1, 33266: 1, 33265: 1, 33264: 1, 33263: 1, 33262: 1, 33261: 1, 3326
0: 1, 33259: 1, 33258: 1, 33257: 1, 33256: 1, 33255: 1, 33254: 1, 3325
3: 1, 33252: 1, 33251: 1, 33250: 1, 33249: 1, 33248: 1, 33247: 1, 3324
6: 1, 33245: 1, 33244: 1, 33243: 1, 33242: 1, 33241: 1, 33240: 1, 3323
9: 1, 33238: 1, 33237: 1, 33236: 1, 33235: 1, 33234: 1, 33233: 1, 3323
2: 1, 33231: 1, 33230: 1, 33229: 1, 33228: 1, 33227: 1, 33226: 1, 3322
5: 1, 33224: 1, 33223: 1, 33222: 1, 33221: 1, 33220: 1, 33219: 1, 3321
8: 1, 33217: 1, 33216: 1, 33215: 1, 33214: 1, 33213: 1, 33212: 1, 3321
1: 1, 33210: 1, 33209: 1, 33208: 1, 33207: 1, 33206: 1, 33205: 1, 3320
4: 1, 33203: 1, 33202: 1, 33201: 1, 33200: 1, 33199: 1, 33198: 1, 3319
7: 1, 33196: 1, 33195: 1, 33194: 1, 33193: 1, 33192: 1, 33191: 1, 3319
0: 1, 33189: 1, 33188: 1, 33187: 1, 33186: 1, 33185: 1, 33184: 1, 3318
3: 1, 33182: 1, 33181: 1, 33180: 1, 33179: 1, 33178: 1, 33177: 1, 3317
6: 1, 33175: 1, 33174: 1, 33173: 1, 33172: 1, 33171: 1, 33170: 1, 3316
9: 1, 33168: 1, 33167: 1, 33166: 1, 33165: 1, 33164: 1, 33163: 1, 3316
2: 1, 33161: 1, 33160: 1, 33159: 1, 33158: 1, 33157: 1, 33156: 1, 3315
5: 1, 33154: 1, 33153: 1, 33152: 1, 33151: 1, 33150: 1, 33149: 1, 3314
8: 1, 33147: 1, 33146: 1, 33145: 1, 33144: 1, 33143: 1, 33142: 1, 3314
1: 1, 33140: 1, 33139: 1, 33138: 1, 33137: 1, 33136: 1, 33135: 1, 3313
4: 1, 33133: 1, 33132: 1, 33131: 1, 33130: 1, 33129: 1, 33128: 1, 3312
7: 1, 33126: 1, 33125: 1, 33124: 1, 33123: 1, 33122: 1, 33121: 1, 3312
0: 1, 33119: 1, 33118: 1, 33117: 1, 33116: 1, 33115: 1, 33114: 1, 3311
3: 1, 33112: 1, 33111: 1, 33110: 1, 33109: 1, 33108: 1, 33107: 1, 3310
6: 1, 33105: 1, 33104: 1, 33103: 1, 33102: 1, 33101: 1, 33100: 1, 3309
9: 1, 33098: 1, 33097: 1, 33096: 1, 33095: 1, 33094: 1, 33093: 1, 3309
2: 1, 33091: 1, 33090: 1, 33089: 1, 33088: 1, 33087: 1, 33086: 1, 3308
5: 1, 33084: 1, 33083: 1, 33082: 1, 33081: 1, 33080: 1, 33079: 1, 3307
8: 1, 33077: 1, 33076: 1, 33075: 1, 33074: 1, 33073: 1, 33072: 1, 3307
1: 1, 33070: 1, 33069: 1, 33068: 1, 33067: 1, 33066: 1, 33065: 1, 3306
4: 1, 33063: 1, 33062: 1, 33061: 1, 33060: 1, 33059: 1, 33058: 1, 3305
7: 1, 33056: 1, 33055: 1, 33054: 1, 33053: 1, 33052: 1, 33051: 1, 3305
0: 1, 33049: 1, 33048: 1, 33047: 1, 33046: 1, 33045: 1, 33044: 1, 3304
3: 1, 33042: 1, 33041: 1, 33040: 1, 33039: 1, 33038: 1, 33037: 1, 3303
6: 1, 33035: 1, 33034: 1, 33033: 1, 33032: 1, 33031: 1, 33030: 1, 3302
9: 1, 33028: 1, 33027: 1, 33026: 1, 33025: 1, 33024: 1, 33023: 1, 3302
2: 1, 33021: 1, 33020: 1, 33019: 1, 33018: 1, 33017: 1, 33016: 1, 3301
5: 1, 33014: 1, 33013: 1, 33012: 1, 33011: 1, 33010: 1, 33009: 1, 3300
8: 1, 33007: 1, 33006: 1, 33005: 1, 33004: 1, 33003: 1, 33002: 1, 3300

1: 1, 33000: 1, 32999: 1, 32998: 1, 32997: 1, 32996: 1, 32995: 1, 3299
4: 1, 32993: 1, 32992: 1, 32991: 1, 32990: 1, 32989: 1, 32988: 1, 3298
7: 1, 32986: 1, 32985: 1, 32984: 1, 32983: 1, 32982: 1, 32981: 1, 3298
0: 1, 32979: 1, 32978: 1, 32977: 1, 32976: 1, 32975: 1, 32974: 1, 3297
3: 1, 32972: 1, 32971: 1, 32970: 1, 32969: 1, 32968: 1, 32967: 1, 3296
6: 1, 32965: 1, 32964: 1, 32963: 1, 32962: 1, 32961: 1, 32960: 1, 3295
9: 1, 32958: 1, 32957: 1, 32956: 1, 32955: 1, 32954: 1, 32953: 1, 3295
2: 1, 32951: 1, 32950: 1, 32949: 1, 32948: 1, 32947: 1, 32946: 1, 3294
5: 1, 32944: 1, 32943: 1, 32942: 1, 32941: 1, 32940: 1, 32939: 1, 3293
8: 1, 32937: 1, 32936: 1, 32935: 1, 32934: 1, 32933: 1, 32932: 1, 3293
1: 1, 32930: 1, 32929: 1, 32928: 1, 32927: 1, 32926: 1, 32925: 1, 3292
4: 1, 32923: 1, 32922: 1, 32921: 1, 32920: 1, 32919: 1, 32918: 1, 3291
7: 1, 32916: 1, 32915: 1, 32914: 1, 32913: 1, 32912: 1, 32911: 1, 3291
0: 1, 32909: 1, 32908: 1, 32907: 1, 32906: 1, 32905: 1, 32904: 1, 3290
3: 1, 32902: 1, 32901: 1, 32900: 1, 32899: 1, 32898: 1, 32897: 1, 3289
6: 1, 32895: 1, 32894: 1, 32893: 1, 32892: 1, 32891: 1, 32890: 1, 3288
9: 1, 32888: 1, 32887: 1, 32886: 1, 32885: 1, 32884: 1, 32883: 1, 3288
2: 1, 32881: 1, 32880: 1, 32879: 1, 32878: 1, 32877: 1, 32876: 1, 3287
5: 1, 32874: 1, 32873: 1, 32872: 1, 32871: 1, 32870: 1, 32869: 1, 3286
8: 1, 32867: 1, 32866: 1, 32865: 1, 32864: 1, 32863: 1, 32862: 1, 3286
1: 1, 32860: 1, 32859: 1, 32858: 1, 32857: 1, 32856: 1, 32855: 1, 3285
4: 1, 32853: 1, 32852: 1, 32851: 1, 32850: 1, 32849: 1, 32848: 1, 3284
7: 1, 32846: 1, 32845: 1, 32844: 1, 32843: 1, 32842: 1, 32841: 1, 3284
0: 1, 32839: 1, 32838: 1, 32837: 1, 32836: 1, 32835: 1, 32834: 1, 3283
3: 1, 32832: 1, 32831: 1, 32830: 1, 32829: 1, 32828: 1, 32827: 1, 3282
6: 1, 32825: 1, 32824: 1, 32823: 1, 32822: 1, 32821: 1, 32820: 1, 3281
9: 1, 32818: 1, 32817: 1, 32816: 1, 32815: 1, 32814: 1, 32813: 1, 3281
2: 1, 32811: 1, 32810: 1, 32809: 1, 32808: 1, 32807: 1, 32806: 1, 3280
5: 1, 32804: 1, 32803: 1, 32802: 1, 32801: 1, 32800: 1, 32799: 1, 3279
8: 1, 32797: 1, 32796: 1, 32795: 1, 32794: 1, 32793: 1, 32792: 1, 3279
1: 1, 32790: 1, 32789: 1, 32788: 1, 32787: 1, 32786: 1, 32785: 1, 3278
4: 1, 32783: 1, 32782: 1, 32781: 1, 32780: 1, 32779: 1, 32778: 1, 3277
7: 1, 32776: 1, 32775: 1, 32774: 1, 32773: 1, 32772: 1, 32771: 1, 3277
0: 1, 32769: 1, 32768: 1, 32767: 1, 32766: 1, 32765: 1, 32764: 1, 3276
3: 1, 32762: 1, 32761: 1, 32760: 1, 32759: 1, 32758: 1, 32757: 1, 3275
6: 1, 32755: 1, 32754: 1, 32753: 1, 32752: 1, 32751: 1, 32750: 1, 3274
9: 1, 32748: 1, 32747: 1, 32746: 1, 32745: 1, 32744: 1, 32743: 1, 3274
2: 1, 32741: 1, 32740: 1, 32739: 1, 32738: 1, 32737: 1, 32736: 1, 3273
5: 1, 32734: 1, 32733: 1, 32732: 1, 32731: 1, 32730: 1, 32729: 1, 3272

8: 1, 32727: 1, 32726: 1, 32725: 1, 32724: 1, 32723: 1, 32722: 1, 3272
1: 1, 32720: 1, 32719: 1, 32718: 1, 32717: 1, 32716: 1, 32715: 1, 3271
4: 1, 32713: 1, 32712: 1, 32711: 1, 32710: 1, 32709: 1, 32708: 1, 3270
7: 1, 32706: 1, 32705: 1, 32704: 1, 32703: 1, 32702: 1, 32701: 1, 3270
0: 1, 32699: 1, 32698: 1, 32697: 1, 32696: 1, 32695: 1, 32694: 1, 3269
3: 1, 32692: 1, 32691: 1, 32690: 1, 32689: 1, 32688: 1, 32687: 1, 3268
6: 1, 32685: 1, 32684: 1, 32683: 1, 32682: 1, 32681: 1, 32680: 1, 3267
9: 1, 32678: 1, 32677: 1, 32676: 1, 32675: 1, 32674: 1, 32673: 1, 3267
2: 1, 32671: 1, 32670: 1, 32669: 1, 32668: 1, 32667: 1, 32666: 1, 3266
5: 1, 32664: 1, 32663: 1, 32662: 1, 32661: 1, 32660: 1, 32659: 1, 3265
8: 1, 32657: 1, 32656: 1, 32655: 1, 32654: 1, 32653: 1, 32652: 1, 3265
1: 1, 32650: 1, 32649: 1, 32648: 1, 32647: 1, 32646: 1, 32645: 1, 3264
4: 1, 32643: 1, 32642: 1, 32641: 1, 32640: 1, 32639: 1, 32638: 1, 3263
7: 1, 32636: 1, 32635: 1, 32634: 1, 32633: 1, 32632: 1, 32631: 1, 3263
0: 1, 32629: 1, 32628: 1, 32627: 1, 32626: 1, 32625: 1, 32624: 1, 3262
3: 1, 32622: 1, 32621: 1, 32620: 1, 32619: 1, 32618: 1, 32617: 1, 3261
6: 1, 32615: 1, 32614: 1, 32613: 1, 32612: 1, 32611: 1, 32610: 1, 3260
9: 1, 32608: 1, 32607: 1, 32606: 1, 32605: 1, 32604: 1, 32603: 1, 3260
2: 1, 32601: 1, 32600: 1, 32599: 1, 32598: 1, 32597: 1, 32596: 1, 3259
5: 1, 32594: 1, 32593: 1, 32592: 1, 32591: 1, 32590: 1, 32589: 1, 3258
8: 1, 32587: 1, 32586: 1, 32585: 1, 32584: 1, 32583: 1, 32582: 1, 3258
1: 1, 32580: 1, 32579: 1, 32578: 1, 32577: 1, 32576: 1, 32575: 1, 3257
4: 1, 32573: 1, 32572: 1, 32571: 1, 32570: 1, 32569: 1, 32568: 1, 3256
7: 1, 32566: 1, 32565: 1, 32564: 1, 32563: 1, 32562: 1, 32561: 1, 3256
0: 1, 32559: 1, 32558: 1, 32557: 1, 32556: 1, 32555: 1, 32554: 1, 3255
3: 1, 32552: 1, 32551: 1, 32550: 1, 32549: 1, 32548: 1, 32547: 1, 3254
6: 1, 32545: 1, 32544: 1, 32543: 1, 32542: 1, 32541: 1, 32540: 1, 3253
9: 1, 32538: 1, 32537: 1, 32536: 1, 32535: 1, 32534: 1, 32533: 1, 3253
2: 1, 32531: 1, 32530: 1, 32529: 1, 32528: 1, 32527: 1, 32526: 1, 3252
5: 1, 32524: 1, 32523: 1, 32522: 1, 32521: 1, 32520: 1, 32519: 1, 3251
8: 1, 32517: 1, 32516: 1, 32515: 1, 32514: 1, 32513: 1, 32512: 1, 3251
1: 1, 32510: 1, 32509: 1, 32508: 1, 32507: 1, 32506: 1, 32505: 1, 3250
4: 1, 32503: 1, 32502: 1, 32501: 1, 32500: 1, 32499: 1, 32498: 1, 3249
7: 1, 32496: 1, 32495: 1, 32494: 1, 32493: 1, 32492: 1, 32491: 1, 3249
0: 1, 32489: 1, 32488: 1, 32487: 1, 32486: 1, 32485: 1, 32484: 1, 3248
3: 1, 32482: 1, 32481: 1, 32480: 1, 32479: 1, 32478: 1, 32477: 1, 3247
6: 1, 32475: 1, 32474: 1, 32473: 1, 32472: 1, 32471: 1, 32470: 1, 3246
9: 1, 32468: 1, 32467: 1, 32466: 1, 32465: 1, 32464: 1, 32463: 1, 3246
2: 1, 32461: 1, 32460: 1, 32459: 1, 32458: 1, 32457: 1, 32456: 1, 3245

5: 1, 32454: 1, 32453: 1, 32452: 1, 32451: 1, 32450: 1, 32449: 1, 3244
8: 1, 32447: 1, 32446: 1, 32445: 1, 32444: 1, 32443: 1, 32442: 1, 3244
1: 1, 32440: 1, 32439: 1, 32438: 1, 32437: 1, 32436: 1, 32435: 1, 3243
4: 1, 32433: 1, 32432: 1, 32431: 1, 32430: 1, 32429: 1, 32428: 1, 3242
7: 1, 32426: 1, 32425: 1, 32424: 1, 32423: 1, 32422: 1, 32421: 1, 3242
0: 1, 32419: 1, 32418: 1, 32417: 1, 32416: 1, 32415: 1, 32414: 1, 3241
3: 1, 32412: 1, 32411: 1, 32410: 1, 32409: 1, 32408: 1, 32407: 1, 3240
6: 1, 32405: 1, 32404: 1, 32403: 1, 32402: 1, 32401: 1, 32400: 1, 3239
9: 1, 32398: 1, 32397: 1, 32396: 1, 32395: 1, 32394: 1, 32393: 1, 3239
2: 1, 32391: 1, 32390: 1, 32389: 1, 32388: 1, 32387: 1, 32386: 1, 3238
5: 1, 32384: 1, 32383: 1, 32382: 1, 32381: 1, 32380: 1, 32379: 1, 3237
8: 1, 32377: 1, 32376: 1, 32375: 1, 32374: 1, 32373: 1, 32372: 1, 3237
1: 1, 32370: 1, 32369: 1, 32368: 1, 32367: 1, 32366: 1, 32365: 1, 3236
4: 1, 32363: 1, 32362: 1, 32361: 1, 32360: 1, 32359: 1, 32358: 1, 3235
7: 1, 32356: 1, 32355: 1, 32354: 1, 32353: 1, 32352: 1, 32351: 1, 3235
0: 1, 32349: 1, 32348: 1, 32347: 1, 32346: 1, 32345: 1, 32344: 1, 3234
3: 1, 32342: 1, 32341: 1, 32340: 1, 32339: 1, 32338: 1, 32337: 1, 3233
6: 1, 32335: 1, 32334: 1, 32333: 1, 32332: 1, 32331: 1, 32330: 1, 3232
9: 1, 32328: 1, 32327: 1, 32326: 1, 32325: 1, 32324: 1, 32323: 1, 3232
2: 1, 32321: 1, 32320: 1, 32319: 1, 32318: 1, 32317: 1, 32316: 1, 3231
5: 1, 32314: 1, 32313: 1, 32312: 1, 32311: 1, 32310: 1, 32309: 1, 3230
8: 1, 32307: 1, 32306: 1, 32305: 1, 32304: 1, 32303: 1, 32302: 1, 3230
1: 1, 32300: 1, 32299: 1, 32298: 1, 32297: 1, 32296: 1, 32295: 1, 3229
4: 1, 32293: 1, 32292: 1, 32291: 1, 32290: 1, 32289: 1, 32288: 1, 3228
7: 1, 32286: 1, 32285: 1, 32284: 1, 32283: 1, 32282: 1, 32281: 1, 3228
0: 1, 32279: 1, 32278: 1, 32277: 1, 32276: 1, 32275: 1, 32274: 1, 3227
3: 1, 32272: 1, 32271: 1, 32270: 1, 32269: 1, 32268: 1, 32267: 1, 3226
6: 1, 32265: 1, 32264: 1, 32263: 1, 32262: 1, 32261: 1, 32260: 1, 3225
9: 1, 32258: 1, 32257: 1, 32256: 1, 32255: 1, 32254: 1, 32253: 1, 3225
2: 1, 32251: 1, 32250: 1, 32249: 1, 32248: 1, 32247: 1, 32246: 1, 3224
5: 1, 32244: 1, 32243: 1, 32242: 1, 32241: 1, 32240: 1, 32239: 1, 3223
8: 1, 32237: 1, 32236: 1, 32235: 1, 32234: 1, 32233: 1, 32232: 1, 3223
1: 1, 32230: 1, 32229: 1, 32228: 1, 32227: 1, 32226: 1, 32225: 1, 3222
4: 1, 32223: 1, 32222: 1, 32221: 1, 32220: 1, 32219: 1, 32218: 1, 3221
7: 1, 32216: 1, 32215: 1, 32214: 1, 32213: 1, 32212: 1, 32211: 1, 3221
0: 1, 32209: 1, 32208: 1, 32207: 1, 32206: 1, 32205: 1, 32204: 1, 3220
3: 1, 32202: 1, 32201: 1, 32200: 1, 32199: 1, 32198: 1, 32197: 1, 3219
6: 1, 32195: 1, 32194: 1, 32193: 1, 32192: 1, 32191: 1, 32190: 1, 3218
9: 1, 32188: 1, 32187: 1, 32186: 1, 32185: 1, 32184: 1, 32183: 1, 3218

2: 1, 32181: 1, 32180: 1, 32179: 1, 32178: 1, 32177: 1, 32176: 1, 3217
5: 1, 32174: 1, 32173: 1, 32172: 1, 32171: 1, 32170: 1, 32169: 1, 3216
8: 1, 32167: 1, 32166: 1, 32165: 1, 32164: 1, 32163: 1, 32162: 1, 3216
1: 1, 32160: 1, 32159: 1, 32158: 1, 32157: 1, 32156: 1, 32155: 1, 3215
4: 1, 32153: 1, 32152: 1, 32151: 1, 32150: 1, 32149: 1, 32148: 1, 3214
7: 1, 32146: 1, 32145: 1, 32144: 1, 32143: 1, 32142: 1, 32141: 1, 3214
0: 1, 32139: 1, 32138: 1, 32137: 1, 32136: 1, 32135: 1, 32134: 1, 3213
3: 1, 32132: 1, 32131: 1, 32130: 1, 32129: 1, 32128: 1, 32127: 1, 3212
6: 1, 32125: 1, 32124: 1, 32123: 1, 32122: 1, 32121: 1, 32120: 1, 3211
9: 1, 32118: 1, 32117: 1, 32116: 1, 32115: 1, 32114: 1, 32113: 1, 3211
2: 1, 32111: 1, 32110: 1, 32109: 1, 32108: 1, 32107: 1, 32106: 1, 3210
5: 1, 32104: 1, 32103: 1, 32102: 1, 32101: 1, 32100: 1, 32099: 1, 3209
8: 1, 32097: 1, 32096: 1, 32095: 1, 32094: 1, 32093: 1, 32092: 1, 3209
1: 1, 32090: 1, 32089: 1, 32088: 1, 32087: 1, 32086: 1, 32085: 1, 3208
4: 1, 32083: 1, 32082: 1, 32081: 1, 32080: 1, 32079: 1, 32078: 1, 3207
7: 1, 32076: 1, 32075: 1, 32074: 1, 32073: 1, 32072: 1, 32071: 1, 3207
0: 1, 32069: 1, 32068: 1, 32067: 1, 32066: 1, 32065: 1, 32064: 1, 3206
3: 1, 32062: 1, 32061: 1, 32060: 1, 32059: 1, 32058: 1, 32057: 1, 3205
6: 1, 32055: 1, 32054: 1, 32053: 1, 32052: 1, 32051: 1, 32050: 1, 3204
9: 1, 32048: 1, 32047: 1, 32046: 1, 32045: 1, 32044: 1, 32043: 1, 3204
2: 1, 32041: 1, 32040: 1, 32039: 1, 32038: 1, 32037: 1, 32036: 1, 3203
5: 1, 32034: 1, 32033: 1, 32032: 1, 32031: 1, 32030: 1, 32029: 1, 3202
8: 1, 32027: 1, 32026: 1, 32025: 1, 32024: 1, 32023: 1, 32022: 1, 3202
1: 1, 32020: 1, 32019: 1, 32018: 1, 32017: 1, 32016: 1, 32015: 1, 3201
4: 1, 32013: 1, 32012: 1, 32011: 1, 32010: 1, 32009: 1, 32008: 1, 3200
7: 1, 32006: 1, 32005: 1, 32004: 1, 32003: 1, 32002: 1, 32001: 1, 3200
0: 1, 31999: 1, 31998: 1, 31997: 1, 31996: 1, 31995: 1, 31994: 1, 3199
3: 1, 31992: 1, 31991: 1, 31990: 1, 31989: 1, 31988: 1, 31987: 1, 3198
6: 1, 31985: 1, 31984: 1, 31983: 1, 31982: 1, 31981: 1, 31980: 1, 3197
9: 1, 31978: 1, 31977: 1, 31976: 1, 31975: 1, 31974: 1, 31973: 1, 3197
2: 1, 31971: 1, 31970: 1, 31969: 1, 31968: 1, 31967: 1, 31966: 1, 3196
5: 1, 31964: 1, 31963: 1, 31962: 1, 31961: 1, 31960: 1, 31959: 1, 3195
8: 1, 31957: 1, 31956: 1, 31955: 1, 31954: 1, 31953: 1, 31952: 1, 3195
1: 1, 31950: 1, 31949: 1, 31948: 1, 31947: 1, 31946: 1, 31945: 1, 3194
4: 1, 31943: 1, 31942: 1, 31941: 1, 31940: 1, 31939: 1, 31938: 1, 3193
7: 1, 31936: 1, 31935: 1, 31934: 1, 31933: 1, 31932: 1, 31931: 1, 3193
0: 1, 31929: 1, 31928: 1, 31927: 1, 31926: 1, 31925: 1, 31924: 1, 3192
3: 1, 31922: 1, 31921: 1, 31920: 1, 31919: 1, 31918: 1, 31917: 1, 3191
6: 1, 31915: 1, 31914: 1, 31913: 1, 31912: 1, 31911: 1, 31910: 1, 3190

9: 1, 31908: 1, 31907: 1, 31906: 1, 31905: 1, 31904: 1, 31903: 1, 3190
2: 1, 31901: 1, 31900: 1, 31899: 1, 31898: 1, 31897: 1, 31896: 1, 3189
5: 1, 31894: 1, 31893: 1, 31892: 1, 31891: 1, 31890: 1, 31889: 1, 3188
8: 1, 31887: 1, 31886: 1, 31885: 1, 31884: 1, 31883: 1, 31882: 1, 3188
1: 1, 31880: 1, 31879: 1, 31878: 1, 31877: 1, 31876: 1, 31875: 1, 3187
4: 1, 31873: 1, 31872: 1, 31871: 1, 31870: 1, 31869: 1, 31868: 1, 3186
7: 1, 31866: 1, 31865: 1, 31864: 1, 31863: 1, 31862: 1, 31861: 1, 3186
0: 1, 31859: 1, 31858: 1, 31857: 1, 31856: 1, 31855: 1, 31854: 1, 3185
3: 1, 31852: 1, 31851: 1, 31850: 1, 31849: 1, 31848: 1, 31847: 1, 3184
6: 1, 31845: 1, 31844: 1, 31843: 1, 31842: 1, 31841: 1, 31840: 1, 3183
9: 1, 31838: 1, 31837: 1, 31836: 1, 31835: 1, 31834: 1, 31833: 1, 3183
2: 1, 31831: 1, 31830: 1, 31829: 1, 31828: 1, 31827: 1, 31826: 1, 3182
5: 1, 31824: 1, 31823: 1, 31822: 1, 31821: 1, 31820: 1, 31819: 1, 3181
8: 1, 31817: 1, 31816: 1, 31815: 1, 31814: 1, 31813: 1, 31812: 1, 3181
1: 1, 31810: 1, 31809: 1, 31808: 1, 31807: 1, 31806: 1, 31805: 1, 3180
4: 1, 31803: 1, 31802: 1, 31801: 1, 31800: 1, 31799: 1, 31798: 1, 3179
7: 1, 31796: 1, 31795: 1, 31794: 1, 31793: 1, 31792: 1, 31791: 1, 3179
0: 1, 31789: 1, 31788: 1, 31787: 1, 31786: 1, 31785: 1, 31784: 1, 3178
3: 1, 31782: 1, 31781: 1, 31780: 1, 31779: 1, 31778: 1, 31777: 1, 3177
6: 1, 31775: 1, 31774: 1, 31773: 1, 31772: 1, 31771: 1, 31770: 1, 3176
9: 1, 31768: 1, 31767: 1, 31766: 1, 31765: 1, 31764: 1, 31763: 1, 3176
2: 1, 31761: 1, 31760: 1, 31759: 1, 31758: 1, 31757: 1, 31756: 1, 3175
5: 1, 31754: 1, 31753: 1, 31752: 1, 31751: 1, 31750: 1, 31749: 1, 3174
8: 1, 31747: 1, 31746: 1, 31745: 1, 31744: 1, 31743: 1, 31742: 1, 3174
1: 1, 31740: 1, 31739: 1, 31738: 1, 31737: 1, 31736: 1, 31735: 1, 3173
4: 1, 31733: 1, 31732: 1, 31731: 1, 31730: 1, 31729: 1, 31728: 1, 3172
7: 1, 31726: 1, 31725: 1, 31724: 1, 31723: 1, 31722: 1, 31721: 1, 3172
0: 1, 31719: 1, 31718: 1, 31717: 1, 31716: 1, 31715: 1, 31714: 1, 3171
3: 1, 31712: 1, 31711: 1, 31710: 1, 31709: 1, 31708: 1, 31707: 1, 3170
6: 1, 31705: 1, 31704: 1, 31703: 1, 31702: 1, 31701: 1, 31700: 1, 3169
9: 1, 31698: 1, 31697: 1, 31696: 1, 31695: 1, 31694: 1, 31693: 1, 3169
2: 1, 31691: 1, 31690: 1, 31689: 1, 31688: 1, 31687: 1, 31686: 1, 3168
5: 1, 31684: 1, 31683: 1, 31682: 1, 31681: 1, 31680: 1, 31679: 1, 3167
8: 1, 31677: 1, 31676: 1, 31675: 1, 31674: 1, 31673: 1, 31672: 1, 3167
1: 1, 31670: 1, 31669: 1, 31668: 1, 31667: 1, 31666: 1, 31665: 1, 3166
4: 1, 31663: 1, 31662: 1, 31661: 1, 31660: 1, 31659: 1, 31658: 1, 3165
7: 1, 31656: 1, 31655: 1, 31654: 1, 31653: 1, 31652: 1, 31651: 1, 3165
0: 1, 31649: 1, 31648: 1, 31647: 1, 31646: 1, 31645: 1, 31644: 1, 3164
3: 1, 31642: 1, 31641: 1, 31640: 1, 31639: 1, 31638: 1, 31637: 1, 3163

6: 1, 31635: 1, 31634: 1, 31633: 1, 31632: 1, 31631: 1, 31630: 1, 3162
9: 1, 31628: 1, 31627: 1, 31626: 1, 31625: 1, 31624: 1, 31623: 1, 3162
2: 1, 31621: 1, 31620: 1, 31619: 1, 31618: 1, 31617: 1, 31616: 1, 3161
5: 1, 31614: 1, 31613: 1, 31612: 1, 31611: 1, 31610: 1, 31609: 1, 3160
8: 1, 31607: 1, 31606: 1, 31605: 1, 31604: 1, 31603: 1, 31602: 1, 3160
1: 1, 31600: 1, 31599: 1, 31598: 1, 31597: 1, 31596: 1, 31595: 1, 3159
4: 1, 31593: 1, 31592: 1, 31591: 1, 31590: 1, 31589: 1, 31588: 1, 3158
7: 1, 31586: 1, 31585: 1, 31584: 1, 31583: 1, 31582: 1, 31581: 1, 3158
0: 1, 31579: 1, 31578: 1, 31577: 1, 31576: 1, 31575: 1, 31574: 1, 3157
3: 1, 31572: 1, 31571: 1, 31570: 1, 31569: 1, 31568: 1, 31567: 1, 3156
6: 1, 31565: 1, 31564: 1, 31563: 1, 31562: 1, 31561: 1, 31560: 1, 3155
9: 1, 31558: 1, 31557: 1, 31556: 1, 31555: 1, 31554: 1, 31553: 1, 3155
2: 1, 31551: 1, 31550: 1, 31549: 1, 31548: 1, 31547: 1, 31546: 1, 3154
5: 1, 31544: 1, 31543: 1, 31542: 1, 31541: 1, 31540: 1, 31539: 1, 3153
8: 1, 31537: 1, 31536: 1, 31535: 1, 31534: 1, 31533: 1, 31532: 1, 3153
1: 1, 31530: 1, 31529: 1, 31528: 1, 31527: 1, 31526: 1, 31525: 1, 3152
4: 1, 31523: 1, 31522: 1, 31521: 1, 31520: 1, 31519: 1, 31518: 1, 3151
7: 1, 31516: 1, 31515: 1, 31514: 1, 31513: 1, 31512: 1, 31511: 1, 3151
0: 1, 31509: 1, 31508: 1, 31507: 1, 31506: 1, 31505: 1, 31504: 1, 3150
3: 1, 31502: 1, 31501: 1, 31500: 1, 31499: 1, 31498: 1, 31497: 1, 3149
6: 1, 31495: 1, 31494: 1, 31493: 1, 31492: 1, 31491: 1, 31490: 1, 3148
9: 1, 31488: 1, 31487: 1, 31486: 1, 31485: 1, 31484: 1, 31483: 1, 3148
2: 1, 31481: 1, 31480: 1, 31479: 1, 31478: 1, 31477: 1, 31476: 1, 3147
5: 1, 31474: 1, 31473: 1, 31472: 1, 31471: 1, 31470: 1, 31469: 1, 3146
8: 1, 31467: 1, 31466: 1, 31465: 1, 31464: 1, 31463: 1, 31462: 1, 3146
1: 1, 31460: 1, 31459: 1, 31458: 1, 31457: 1, 31456: 1, 31455: 1, 3145
4: 1, 31453: 1, 31452: 1, 31451: 1, 31450: 1, 31449: 1, 31448: 1, 3144
7: 1, 31446: 1, 31445: 1, 31444: 1, 31443: 1, 31442: 1, 31441: 1, 3144
0: 1, 31439: 1, 31438: 1, 31437: 1, 31436: 1, 31435: 1, 31434: 1, 3143
3: 1, 31432: 1, 31431: 1, 31430: 1, 31429: 1, 31428: 1, 31427: 1, 3142
6: 1, 31425: 1, 31424: 1, 31423: 1, 31422: 1, 31421: 1, 31420: 1, 3141
9: 1, 31418: 1, 31417: 1, 31416: 1, 31415: 1, 31414: 1, 31413: 1, 3141
2: 1, 31411: 1, 31410: 1, 31409: 1, 31408: 1, 31407: 1, 31406: 1, 3140
5: 1, 31404: 1, 31403: 1, 31402: 1, 31401: 1, 31400: 1, 31399: 1, 3139
8: 1, 31397: 1, 31396: 1, 31395: 1, 31394: 1, 31393: 1, 31392: 1, 3139
1: 1, 31390: 1, 31389: 1, 31388: 1, 31387: 1, 31386: 1, 31385: 1, 3138
4: 1, 31383: 1, 31382: 1, 31381: 1, 31380: 1, 31379: 1, 31378: 1, 3137
7: 1, 31376: 1, 31375: 1, 31374: 1, 31373: 1, 31372: 1, 31371: 1, 3137
0: 1, 31369: 1, 31368: 1, 31367: 1, 31366: 1, 31365: 1, 31364: 1, 3136

3: 1, 31362: 1, 31361: 1, 31360: 1, 31359: 1, 31358: 1, 31357: 1, 3135
6: 1, 31355: 1, 31354: 1, 31353: 1, 31352: 1, 31351: 1, 31350: 1, 3134
9: 1, 31348: 1, 31347: 1, 31346: 1, 31345: 1, 31344: 1, 31343: 1, 3134
2: 1, 31341: 1, 31340: 1, 31339: 1, 31338: 1, 31337: 1, 31336: 1, 3133
5: 1, 31334: 1, 31333: 1, 31332: 1, 31331: 1, 31330: 1, 31329: 1, 3132
8: 1, 31327: 1, 31326: 1, 31325: 1, 31324: 1, 31323: 1, 31322: 1, 3132
1: 1, 31320: 1, 31319: 1, 31318: 1, 31317: 1, 31316: 1, 31315: 1, 3131
4: 1, 31313: 1, 31312: 1, 31311: 1, 31310: 1, 31309: 1, 31308: 1, 3130
7: 1, 31306: 1, 31305: 1, 31304: 1, 31303: 1, 31302: 1, 31301: 1, 3130
0: 1, 31299: 1, 31298: 1, 31297: 1, 31296: 1, 31295: 1, 31294: 1, 3129
3: 1, 31292: 1, 31291: 1, 31290: 1, 31289: 1, 31288: 1, 31287: 1, 3128
6: 1, 31285: 1, 31284: 1, 31283: 1, 31282: 1, 31281: 1, 31280: 1, 3127
9: 1, 31278: 1, 31277: 1, 31276: 1, 31275: 1, 31274: 1, 31273: 1, 3127
2: 1, 31271: 1, 31270: 1, 31269: 1, 31268: 1, 31267: 1, 31266: 1, 3126
5: 1, 31264: 1, 31263: 1, 31262: 1, 31261: 1, 31260: 1, 31259: 1, 3125
8: 1, 31257: 1, 31256: 1, 31255: 1, 31254: 1, 31253: 1, 31252: 1, 3125
1: 1, 31250: 1, 31249: 1, 31248: 1, 31247: 1, 31246: 1, 31245: 1, 3124
4: 1, 31243: 1, 31242: 1, 31241: 1, 31240: 1, 31239: 1, 31238: 1, 3123
7: 1, 31236: 1, 31235: 1, 31234: 1, 31233: 1, 31232: 1, 31231: 1, 3123
0: 1, 31229: 1, 31228: 1, 31227: 1, 31226: 1, 31225: 1, 31224: 1, 3122
3: 1, 31222: 1, 31221: 1, 31220: 1, 31219: 1, 31218: 1, 31217: 1, 3121
6: 1, 31215: 1, 31214: 1, 31213: 1, 31212: 1, 31211: 1, 31210: 1, 3120
9: 1, 31208: 1, 31207: 1, 31206: 1, 31205: 1, 31204: 1, 31203: 1, 3120
2: 1, 31201: 1, 31200: 1, 31199: 1, 31198: 1, 31197: 1, 31196: 1, 3119
5: 1, 31194: 1, 31193: 1, 31192: 1, 31191: 1, 31190: 1, 31189: 1, 3118
8: 1, 31187: 1, 31186: 1, 31185: 1, 31184: 1, 31183: 1, 31182: 1, 3118
1: 1, 31180: 1, 31179: 1, 31178: 1, 31177: 1, 31176: 1, 31175: 1, 3117
4: 1, 31173: 1, 31172: 1, 31171: 1, 31170: 1, 31169: 1, 31168: 1, 3116
7: 1, 31166: 1, 31165: 1, 31164: 1, 31163: 1, 31162: 1, 31161: 1, 3116
0: 1, 31159: 1, 31158: 1, 31157: 1, 31156: 1, 31155: 1, 31154: 1, 3115
3: 1, 31152: 1, 31151: 1, 31150: 1, 31149: 1, 31148: 1, 31147: 1, 3114
6: 1, 31145: 1, 31144: 1, 31143: 1, 31142: 1, 31141: 1, 31140: 1, 3113
9: 1, 31138: 1, 31137: 1, 31136: 1, 31135: 1, 31134: 1, 31133: 1, 3113
2: 1, 31131: 1, 31130: 1, 31129: 1, 31128: 1, 31127: 1, 31126: 1, 3112
5: 1, 31124: 1, 31123: 1, 31122: 1, 31121: 1, 31120: 1, 31119: 1, 3111
8: 1, 31117: 1, 31116: 1, 31115: 1, 31114: 1, 31113: 1, 31112: 1, 3111
1: 1, 31110: 1, 31109: 1, 31108: 1, 31107: 1, 31106: 1, 31105: 1, 3110
4: 1, 31103: 1, 31102: 1, 31101: 1, 31100: 1, 31099: 1, 31098: 1, 3109
7: 1, 31096: 1, 31095: 1, 31094: 1, 31093: 1, 31092: 1, 31091: 1, 3109

0: 1, 31089: 1, 31088: 1, 31087: 1, 31086: 1, 31085: 1, 31084: 1, 3108
3: 1, 31082: 1, 31081: 1, 31080: 1, 31079: 1, 31078: 1, 31077: 1, 3107
6: 1, 31075: 1, 31074: 1, 31073: 1, 31072: 1, 31071: 1, 31070: 1, 3106
9: 1, 31068: 1, 31067: 1, 31066: 1, 31065: 1, 31064: 1, 31063: 1, 3106
2: 1, 31061: 1, 31060: 1, 31059: 1, 31058: 1, 31057: 1, 31056: 1, 3105
5: 1, 31054: 1, 31053: 1, 31052: 1, 31051: 1, 31050: 1, 31049: 1, 3104
8: 1, 31047: 1, 31046: 1, 31045: 1, 31044: 1, 31043: 1, 31042: 1, 3104
1: 1, 31040: 1, 31039: 1, 31038: 1, 31037: 1, 31036: 1, 31035: 1, 3103
4: 1, 31033: 1, 31032: 1, 31031: 1, 31030: 1, 31029: 1, 31028: 1, 3102
7: 1, 31026: 1, 31025: 1, 31024: 1, 31023: 1, 31022: 1, 31021: 1, 3102
0: 1, 31019: 1, 31018: 1, 31017: 1, 31016: 1, 31015: 1, 31014: 1, 3101
3: 1, 31012: 1, 31011: 1, 31010: 1, 31009: 1, 31008: 1, 31007: 1, 3100
6: 1, 31005: 1, 31004: 1, 31003: 1, 31002: 1, 31001: 1, 31000: 1, 3099
9: 1, 30998: 1, 30997: 1, 30996: 1, 30995: 1, 30994: 1, 30993: 1, 3099
2: 1, 30991: 1, 30990: 1, 30989: 1, 30988: 1, 30987: 1, 30986: 1, 3098
5: 1, 30984: 1, 30983: 1, 30982: 1, 30981: 1, 30980: 1, 30979: 1, 3097
8: 1, 30977: 1, 30976: 1, 30975: 1, 30974: 1, 30973: 1, 30972: 1, 3097
1: 1, 30970: 1, 30969: 1, 30968: 1, 30967: 1, 30966: 1, 30965: 1, 3096
4: 1, 30963: 1, 30962: 1, 30961: 1, 30960: 1, 30959: 1, 30958: 1, 3095
7: 1, 30956: 1, 30955: 1, 30954: 1, 30953: 1, 30952: 1, 30951: 1, 3095
0: 1, 30949: 1, 30948: 1, 30947: 1, 30946: 1, 30945: 1, 30944: 1, 3094
3: 1, 30942: 1, 30941: 1, 30940: 1, 30939: 1, 30938: 1, 30937: 1, 3093
6: 1, 30935: 1, 30934: 1, 30933: 1, 30932: 1, 30931: 1, 30930: 1, 3092
9: 1, 30928: 1, 30927: 1, 30926: 1, 30925: 1, 30924: 1, 30923: 1, 3092
2: 1, 30921: 1, 30920: 1, 30919: 1, 30918: 1, 30917: 1, 30916: 1, 3091
5: 1, 30914: 1, 30913: 1, 30912: 1, 30911: 1, 30910: 1, 30909: 1, 3090
8: 1, 30907: 1, 30906: 1, 30905: 1, 30904: 1, 30903: 1, 30902: 1, 3090
1: 1, 30900: 1, 30899: 1, 30898: 1, 30897: 1, 30896: 1, 30895: 1, 3089
4: 1, 30893: 1, 30892: 1, 30891: 1, 30890: 1, 30889: 1, 30888: 1, 3088
7: 1, 30886: 1, 30885: 1, 30884: 1, 30883: 1, 30882: 1, 30881: 1, 3088
0: 1, 30879: 1, 30878: 1, 30877: 1, 30876: 1, 30875: 1, 30874: 1, 3087
3: 1, 30872: 1, 30871: 1, 30870: 1, 30869: 1, 30868: 1, 30867: 1, 3086
6: 1, 30865: 1, 30864: 1, 30863: 1, 30862: 1, 30861: 1, 30860: 1, 3085
9: 1, 30858: 1, 30857: 1, 30856: 1, 30855: 1, 30854: 1, 30853: 1, 3085
2: 1, 30851: 1, 30850: 1, 30849: 1, 30848: 1, 30847: 1, 30846: 1, 3084
5: 1, 30844: 1, 30843: 1, 30842: 1, 30841: 1, 30840: 1, 30839: 1, 3083
8: 1, 30837: 1, 30836: 1, 30835: 1, 30834: 1, 30833: 1, 30832: 1, 3083
1: 1, 30830: 1, 30829: 1, 30828: 1, 30827: 1, 30826: 1, 30825: 1, 3082
4: 1, 30823: 1, 30822: 1, 30821: 1, 30820: 1, 30819: 1, 30818: 1, 3081

4: 1, 30543: 1, 30542: 1, 30541: 1, 30540: 1, 30539: 1, 30538: 1, 3053
7: 1, 30536: 1, 30535: 1, 30534: 1, 30533: 1, 30532: 1, 30531: 1, 3053
0: 1, 30529: 1, 30528: 1, 30527: 1, 30526: 1, 30525: 1, 30524: 1, 3052
3: 1, 30522: 1, 30521: 1, 30520: 1, 30519: 1, 30518: 1, 30517: 1, 3051
6: 1, 30515: 1, 30514: 1, 30513: 1, 30512: 1, 30511: 1, 30510: 1, 3050
9: 1, 30508: 1, 30507: 1, 30506: 1, 30505: 1, 30504: 1, 30503: 1, 3050
2: 1, 30501: 1, 30500: 1, 30499: 1, 30498: 1, 30497: 1, 30496: 1, 3049
5: 1, 30494: 1, 30493: 1, 30492: 1, 30491: 1, 30490: 1, 30489: 1, 3048
8: 1, 30487: 1, 30486: 1, 30485: 1, 30484: 1, 30483: 1, 30482: 1, 3048
1: 1, 30480: 1, 30479: 1, 30478: 1, 30477: 1, 30476: 1, 30475: 1, 3047
4: 1, 30473: 1, 30472: 1, 30471: 1, 30470: 1, 30469: 1, 30468: 1, 3046
7: 1, 30466: 1, 30465: 1, 30464: 1, 30463: 1, 30462: 1, 30461: 1, 3046
0: 1, 30459: 1, 30458: 1, 30457: 1, 30456: 1, 30455: 1, 30454: 1, 3045
3: 1, 30452: 1, 30451: 1, 30450: 1, 30449: 1, 30448: 1, 30447: 1, 3044
6: 1, 30445: 1, 30444: 1, 30443: 1, 30442: 1, 30441: 1, 30440: 1, 3043
9: 1, 30438: 1, 30437: 1, 30436: 1, 30435: 1, 30434: 1, 30433: 1, 3043
2: 1, 30431: 1, 30430: 1, 30429: 1, 30428: 1, 30427: 1, 30426: 1, 3042
5: 1, 30424: 1, 30423: 1, 30422: 1, 30421: 1, 30420: 1, 30419: 1, 3041
8: 1, 30417: 1, 30416: 1, 30415: 1, 30414: 1, 30413: 1, 30412: 1, 3041
1: 1, 30410: 1, 30409: 1, 30408: 1, 30407: 1, 30406: 1, 30405: 1, 3040
4: 1, 30403: 1, 30402: 1, 30401: 1, 30400: 1, 30399: 1, 30398: 1, 3039
7: 1, 30396: 1, 30395: 1, 30394: 1, 30393: 1, 30392: 1, 30391: 1, 3039
0: 1, 30389: 1, 30388: 1, 30387: 1, 30386: 1, 30385: 1, 30384: 1, 3038
3: 1, 30382: 1, 30381: 1, 30380: 1, 30379: 1, 30378: 1, 30377: 1, 3037
6: 1, 30375: 1, 30374: 1, 30373: 1, 30372: 1, 30371: 1, 30370: 1, 3036
9: 1, 30368: 1, 30367: 1, 30366: 1, 30365: 1, 30364: 1, 30363: 1, 3036
2: 1, 30361: 1, 30360: 1, 30359: 1, 30358: 1, 30357: 1, 30356: 1, 3035
5: 1, 30354: 1, 30353: 1, 30352: 1, 30351: 1, 30350: 1, 30349: 1, 3034
8: 1, 30347: 1, 30346: 1, 30345: 1, 30344: 1, 30343: 1, 30342: 1, 3034
1: 1, 30340: 1, 30339: 1, 30338: 1, 30337: 1, 30336: 1, 30335: 1, 3033
4: 1, 30333: 1, 30332: 1, 30331: 1, 30330: 1, 30329: 1, 30328: 1, 3032
7: 1, 30326: 1, 30325: 1, 30324: 1, 30323: 1, 30322: 1, 30321: 1, 3032
0: 1, 30319: 1, 30318: 1, 30317: 1, 30316: 1, 30315: 1, 30314: 1, 3031
3: 1, 30312: 1, 30311: 1, 30310: 1, 30309: 1, 30308: 1, 30307: 1, 3030
6: 1, 30305: 1, 30304: 1, 30303: 1, 30302: 1, 30301: 1, 30300: 1, 3029
9: 1, 30298: 1, 30297: 1, 30296: 1, 30295: 1, 30294: 1, 30293: 1, 3029
2: 1, 30291: 1, 30290: 1, 30289: 1, 30288: 1, 30287: 1, 30286: 1, 3028
5: 1, 30284: 1, 30283: 1, 30282: 1, 30281: 1, 30280: 1, 30279: 1, 3027
8: 1, 30277: 1, 30276: 1, 30275: 1, 30274: 1, 30273: 1, 30272: 1, 3027

1: 1, 30270: 1, 30269: 1, 30268: 1, 30267: 1, 30266: 1, 30265: 1, 3026
4: 1, 30263: 1, 30262: 1, 30261: 1, 30260: 1, 30259: 1, 30258: 1, 3025
7: 1, 30256: 1, 30255: 1, 30254: 1, 30253: 1, 30252: 1, 30251: 1, 3025
0: 1, 30249: 1, 30248: 1, 30247: 1, 30246: 1, 30245: 1, 30244: 1, 3024
3: 1, 30242: 1, 30241: 1, 30240: 1, 30239: 1, 30238: 1, 30237: 1, 3023
6: 1, 30235: 1, 30234: 1, 30233: 1, 30232: 1, 30231: 1, 30230: 1, 3022
9: 1, 30228: 1, 30227: 1, 30226: 1, 30225: 1, 30224: 1, 30223: 1, 3022
2: 1, 30221: 1, 30220: 1, 30219: 1, 30218: 1, 30217: 1, 30216: 1, 3021
5: 1, 30214: 1, 30213: 1, 30212: 1, 30211: 1, 30210: 1, 30209: 1, 3020
8: 1, 30207: 1, 30206: 1, 30205: 1, 30204: 1, 30203: 1, 30202: 1, 3020
1: 1, 30200: 1, 30199: 1, 30198: 1, 30197: 1, 30196: 1, 30195: 1, 3019
4: 1, 30193: 1, 30192: 1, 30191: 1, 30190: 1, 30189: 1, 30188: 1, 3018
7: 1, 30186: 1, 30185: 1, 30184: 1, 30183: 1, 30182: 1, 30181: 1, 3018
0: 1, 30179: 1, 30178: 1, 30177: 1, 30176: 1, 30175: 1, 30174: 1, 3017
3: 1, 30172: 1, 30171: 1, 30170: 1, 30169: 1, 30168: 1, 30167: 1, 3016
6: 1, 30165: 1, 30164: 1, 30163: 1, 30162: 1, 30161: 1, 30160: 1, 3015
9: 1, 30158: 1, 30157: 1, 30156: 1, 30155: 1, 30154: 1, 30153: 1, 3015
2: 1, 30151: 1, 30150: 1, 30149: 1, 30148: 1, 30147: 1, 30146: 1, 3014
5: 1, 30144: 1, 30143: 1, 30142: 1, 30141: 1, 30140: 1, 30139: 1, 3013
8: 1, 30137: 1, 30136: 1, 30135: 1, 30134: 1, 30133: 1, 30132: 1, 3013
1: 1, 30130: 1, 30129: 1, 30128: 1, 30127: 1, 30126: 1, 30125: 1, 3012
4: 1, 30123: 1, 30122: 1, 30121: 1, 30120: 1, 30119: 1, 30118: 1, 3011
7: 1, 30116: 1, 30115: 1, 30114: 1, 30113: 1, 30112: 1, 30111: 1, 3011
0: 1, 30109: 1, 30108: 1, 30107: 1, 30106: 1, 30105: 1, 30104: 1, 3010
3: 1, 30102: 1, 30101: 1, 30100: 1, 30099: 1, 30098: 1, 30097: 1, 3009
6: 1, 30095: 1, 30094: 1, 30093: 1, 30092: 1, 30091: 1, 30090: 1, 3008
9: 1, 30088: 1, 30087: 1, 30086: 1, 30085: 1, 30084: 1, 30083: 1, 3008
2: 1, 30081: 1, 30080: 1, 30079: 1, 30078: 1, 30077: 1, 30076: 1, 3007
5: 1, 30074: 1, 30073: 1, 30072: 1, 30071: 1, 30070: 1, 30069: 1, 3006
8: 1, 30067: 1, 30066: 1, 30065: 1, 30064: 1, 30063: 1, 30062: 1, 3006
1: 1, 30060: 1, 30059: 1, 30058: 1, 30057: 1, 30056: 1, 30055: 1, 3005
4: 1, 30053: 1, 30052: 1, 30051: 1, 30050: 1, 30049: 1, 30048: 1, 3004
7: 1, 30046: 1, 30045: 1, 30044: 1, 30043: 1, 30042: 1, 30041: 1, 3004
0: 1, 30039: 1, 30038: 1, 30037: 1, 30036: 1, 30035: 1, 30034: 1, 3003
3: 1, 30032: 1, 30031: 1, 30030: 1, 30029: 1, 30028: 1, 30027: 1, 3002
6: 1, 30025: 1, 30024: 1, 30023: 1, 30022: 1, 30021: 1, 30020: 1, 3001
9: 1, 30018: 1, 30017: 1, 30016: 1, 30015: 1, 30014: 1, 30013: 1, 3001
2: 1, 30011: 1, 30010: 1, 30009: 1, 30008: 1, 30007: 1, 30006: 1, 3000
5: 1, 30004: 1, 30003: 1, 30002: 1, 30001: 1, 30000: 1, 29999: 1, 2999

8: 1, 29997: 1, 29996: 1, 29995: 1, 29994: 1, 29993: 1, 29992: 1, 2999
1: 1, 29990: 1, 29989: 1, 29988: 1, 29987: 1, 29986: 1, 29985: 1, 2998
4: 1, 29983: 1, 29982: 1, 29981: 1, 29980: 1, 29979: 1, 29978: 1, 2997
7: 1, 29976: 1, 29975: 1, 29974: 1, 29973: 1, 29972: 1, 29971: 1, 2997
0: 1, 29969: 1, 29968: 1, 29967: 1, 29966: 1, 29965: 1, 29964: 1, 2996
3: 1, 29962: 1, 29961: 1, 29960: 1, 29959: 1, 29958: 1, 29957: 1, 2995
6: 1, 29955: 1, 29954: 1, 29953: 1, 29952: 1, 29951: 1, 29950: 1, 2994
9: 1, 29948: 1, 29947: 1, 29946: 1, 29945: 1, 29944: 1, 29943: 1, 2994
2: 1, 29941: 1, 29940: 1, 29939: 1, 29938: 1, 29937: 1, 29936: 1, 2993
5: 1, 29934: 1, 29933: 1, 29932: 1, 29931: 1, 29930: 1, 29929: 1, 2992
8: 1, 29927: 1, 29926: 1, 29925: 1, 29924: 1, 29923: 1, 29922: 1, 2992
1: 1, 29920: 1, 29919: 1, 29918: 1, 29917: 1, 29916: 1, 29915: 1, 2991
4: 1, 29913: 1, 29912: 1, 29911: 1, 29910: 1, 29909: 1, 29908: 1, 2990
7: 1, 29906: 1, 29905: 1, 29904: 1, 29903: 1, 29902: 1, 29901: 1, 2990
0: 1, 29899: 1, 29898: 1, 29897: 1, 29896: 1, 29895: 1, 29894: 1, 2989
3: 1, 29892: 1, 29891: 1, 29890: 1, 29889: 1, 29888: 1, 29887: 1, 2988
6: 1, 29885: 1, 29884: 1, 29883: 1, 29882: 1, 29881: 1, 29880: 1, 2987
9: 1, 29878: 1, 29877: 1, 29876: 1, 29875: 1, 29874: 1, 29873: 1, 2987
2: 1, 29871: 1, 29870: 1, 29869: 1, 29868: 1, 29867: 1, 29866: 1, 2986
5: 1, 29864: 1, 29863: 1, 29862: 1, 29861: 1, 29860: 1, 29859: 1, 2985
8: 1, 29857: 1, 29856: 1, 29855: 1, 29854: 1, 29853: 1, 29852: 1, 2985
1: 1, 29850: 1, 29849: 1, 29848: 1, 29847: 1, 29846: 1, 29845: 1, 2984
4: 1, 29843: 1, 29842: 1, 29841: 1, 29840: 1, 29839: 1, 29838: 1, 2983
7: 1, 29836: 1, 29835: 1, 29834: 1, 29833: 1, 29832: 1, 29831: 1, 2983
0: 1, 29829: 1, 29828: 1, 29827: 1, 29826: 1, 29825: 1, 29824: 1, 2982
3: 1, 29822: 1, 29821: 1, 29820: 1, 29819: 1, 29818: 1, 29817: 1, 2981
6: 1, 29815: 1, 29814: 1, 29813: 1, 29812: 1, 29811: 1, 29810: 1, 2980
9: 1, 29808: 1, 29807: 1, 29806: 1, 29805: 1, 29804: 1, 29803: 1, 2980
2: 1, 29801: 1, 29800: 1, 29799: 1, 29798: 1, 29797: 1, 29796: 1, 2979
5: 1, 29794: 1, 29793: 1, 29792: 1, 29791: 1, 29790: 1, 29789: 1, 2978
8: 1, 29787: 1, 29786: 1, 29785: 1, 29784: 1, 29783: 1, 29782: 1, 2978
1: 1, 29780: 1, 29779: 1, 29778: 1, 29777: 1, 29776: 1, 29775: 1, 2977
4: 1, 29773: 1, 29772: 1, 29771: 1, 29770: 1, 29769: 1, 29768: 1, 2976
7: 1, 29766: 1, 29765: 1, 29764: 1, 29763: 1, 29762: 1, 29761: 1, 2976
0: 1, 29759: 1, 29758: 1, 29757: 1, 29756: 1, 29755: 1, 29754: 1, 2975
3: 1, 29752: 1, 29751: 1, 29750: 1, 29749: 1, 29748: 1, 29747: 1, 2974
6: 1, 29745: 1, 29744: 1, 29743: 1, 29742: 1, 29741: 1, 29740: 1, 2973
9: 1, 29738: 1, 29737: 1, 29736: 1, 29735: 1, 29734: 1, 29733: 1, 2973
2: 1, 29731: 1, 29730: 1, 29729: 1, 29728: 1, 29727: 1, 29726: 1, 2972

5: 1, 29724: 1, 29723: 1, 29722: 1, 29721: 1, 29720: 1, 29719: 1, 2971
8: 1, 29717: 1, 29716: 1, 29715: 1, 29714: 1, 29713: 1, 29712: 1, 2971
1: 1, 29710: 1, 29709: 1, 29708: 1, 29707: 1, 29706: 1, 29705: 1, 2970
4: 1, 29703: 1, 29702: 1, 29701: 1, 29700: 1, 29699: 1, 29698: 1, 2969
7: 1, 29696: 1, 29695: 1, 29694: 1, 29693: 1, 29692: 1, 29691: 1, 2969
0: 1, 29689: 1, 29688: 1, 29687: 1, 29686: 1, 29685: 1, 29684: 1, 2968
3: 1, 29682: 1, 29681: 1, 29680: 1, 29679: 1, 29678: 1, 29677: 1, 2967
6: 1, 29675: 1, 29674: 1, 29673: 1, 29672: 1, 29671: 1, 29670: 1, 2966
9: 1, 29668: 1, 29667: 1, 29666: 1, 29665: 1, 29664: 1, 29663: 1, 2966
2: 1, 29661: 1, 29660: 1, 29659: 1, 29658: 1, 29657: 1, 29656: 1, 2965
5: 1, 29654: 1, 29653: 1, 29652: 1, 29651: 1, 29650: 1, 29649: 1, 2964
8: 1, 29647: 1, 29646: 1, 29645: 1, 29644: 1, 29643: 1, 29642: 1, 2964
1: 1, 29640: 1, 29639: 1, 29638: 1, 29637: 1, 29636: 1, 29635: 1, 2963
4: 1, 29633: 1, 29632: 1, 29631: 1, 29630: 1, 29629: 1, 29628: 1, 2962
7: 1, 29626: 1, 29625: 1, 29624: 1, 29623: 1, 29622: 1, 29621: 1, 2962
0: 1, 29619: 1, 29618: 1, 29617: 1, 29616: 1, 29615: 1, 29614: 1, 2961
3: 1, 29612: 1, 29611: 1, 29610: 1, 29609: 1, 29608: 1, 29607: 1, 2960
6: 1, 29605: 1, 29604: 1, 29603: 1, 29602: 1, 29601: 1, 29600: 1, 2959
9: 1, 29598: 1, 29597: 1, 29596: 1, 29595: 1, 29594: 1, 29593: 1, 2959
2: 1, 29591: 1, 29590: 1, 29589: 1, 29588: 1, 29587: 1, 29586: 1, 2958
5: 1, 29584: 1, 29583: 1, 29582: 1, 29581: 1, 29580: 1, 29579: 1, 2957
8: 1, 29577: 1, 29576: 1, 29575: 1, 29574: 1, 29573: 1, 29572: 1, 2957
1: 1, 29570: 1, 29569: 1, 29568: 1, 29567: 1, 29566: 1, 29565: 1, 2956
4: 1, 29563: 1, 29562: 1, 29561: 1, 29560: 1, 29559: 1, 29558: 1, 2955
7: 1, 29556: 1, 29555: 1, 29554: 1, 29553: 1, 29552: 1, 29551: 1, 2955
0: 1, 29549: 1, 29548: 1, 29547: 1, 29546: 1, 29545: 1, 29544: 1, 2954
3: 1, 29542: 1, 29541: 1, 29540: 1, 29539: 1, 29538: 1, 29537: 1, 2953
6: 1, 29535: 1, 29534: 1, 29533: 1, 29532: 1, 29531: 1, 29530: 1, 2952
9: 1, 29528: 1, 29527: 1, 29526: 1, 29525: 1, 29524: 1, 29523: 1, 2952
2: 1, 29521: 1, 29520: 1, 29519: 1, 29518: 1, 29517: 1, 29516: 1, 2951
5: 1, 29514: 1, 29513: 1, 29512: 1, 29511: 1, 29510: 1, 29509: 1, 2950
8: 1, 29507: 1, 29506: 1, 29505: 1, 29504: 1, 29503: 1, 29502: 1, 2950
1: 1, 29500: 1, 29499: 1, 29498: 1, 29497: 1, 29496: 1, 29495: 1, 2949
4: 1, 29493: 1, 29492: 1, 29491: 1, 29490: 1, 29489: 1, 29488: 1, 2948
7: 1, 29486: 1, 29485: 1, 29484: 1, 29483: 1, 29482: 1, 29481: 1, 2948
0: 1, 29479: 1, 29478: 1, 29477: 1, 29476: 1, 29475: 1, 29474: 1, 2947
3: 1, 29472: 1, 29471: 1, 29470: 1, 29469: 1, 29468: 1, 29467: 1, 2946
6: 1, 29465: 1, 29464: 1, 29463: 1, 29462: 1, 29461: 1, 29460: 1, 2945
9: 1, 29458: 1, 29457: 1, 29456: 1, 29455: 1, 29454: 1, 29453: 1, 2945

2: 1, 29451: 1, 29450: 1, 29449: 1, 29448: 1, 29447: 1, 29446: 1, 2944
5: 1, 29444: 1, 29443: 1, 29442: 1, 29441: 1, 29440: 1, 29439: 1, 2943
8: 1, 29437: 1, 29436: 1, 29435: 1, 29434: 1, 29433: 1, 29432: 1, 2943
1: 1, 29430: 1, 29429: 1, 29428: 1, 29427: 1, 29426: 1, 29425: 1, 2942
4: 1, 29423: 1, 29422: 1, 29421: 1, 29420: 1, 29419: 1, 29418: 1, 2941
7: 1, 29416: 1, 29415: 1, 29414: 1, 29413: 1, 29412: 1, 29411: 1, 2941
0: 1, 29409: 1, 29408: 1, 29407: 1, 29406: 1, 29405: 1, 29404: 1, 2940
3: 1, 29402: 1, 29401: 1, 29400: 1, 29399: 1, 29398: 1, 29397: 1, 2939
6: 1, 29395: 1, 29394: 1, 29393: 1, 29392: 1, 29391: 1, 29390: 1, 2938
9: 1, 29388: 1, 29387: 1, 29386: 1, 29385: 1, 29384: 1, 29383: 1, 2938
2: 1, 29381: 1, 29380: 1, 29379: 1, 29378: 1, 29377: 1, 29376: 1, 2937
5: 1, 29374: 1, 29373: 1, 29372: 1, 29371: 1, 29370: 1, 29369: 1, 2936
8: 1, 29367: 1, 29366: 1, 29365: 1, 29364: 1, 29363: 1, 29362: 1, 2936
1: 1, 29360: 1, 29359: 1, 29358: 1, 29357: 1, 29356: 1, 29355: 1, 2935
4: 1, 29353: 1, 29352: 1, 29351: 1, 29350: 1, 29349: 1, 29348: 1, 2934
7: 1, 29346: 1, 29345: 1, 29344: 1, 29343: 1, 29342: 1, 29341: 1, 2934
0: 1, 29339: 1, 29338: 1, 29337: 1, 29336: 1, 29335: 1, 29334: 1, 2933
3: 1, 29332: 1, 29331: 1, 29330: 1, 29329: 1, 29328: 1, 29327: 1, 2932
6: 1, 29325: 1, 29324: 1, 29323: 1, 29322: 1, 29321: 1, 29320: 1, 2931
9: 1, 29318: 1, 29317: 1, 29316: 1, 29315: 1, 29314: 1, 29313: 1, 2931
2: 1, 29311: 1, 29310: 1, 29309: 1, 29308: 1, 29307: 1, 29306: 1, 2930
5: 1, 29304: 1, 29303: 1, 29302: 1, 29301: 1, 29300: 1, 29299: 1, 2929
8: 1, 29297: 1, 29296: 1, 29295: 1, 29294: 1, 29293: 1, 29292: 1, 2929
1: 1, 29290: 1, 29289: 1, 29288: 1, 29287: 1, 29286: 1, 29285: 1, 2928
4: 1, 29283: 1, 29282: 1, 29281: 1, 29280: 1, 29279: 1, 29278: 1, 2927
7: 1, 29276: 1, 29275: 1, 29274: 1, 29273: 1, 29272: 1, 29271: 1, 2927
0: 1, 29269: 1, 29268: 1, 29267: 1, 29266: 1, 29265: 1, 29264: 1, 2926
3: 1, 29262: 1, 29261: 1, 29260: 1, 29259: 1, 29258: 1, 29257: 1, 2925
6: 1, 29255: 1, 29254: 1, 29253: 1, 29252: 1, 29251: 1, 29250: 1, 2924
9: 1, 29248: 1, 29247: 1, 29246: 1, 29245: 1, 29244: 1, 29243: 1, 2924
2: 1, 29241: 1, 29240: 1, 29239: 1, 29238: 1, 29237: 1, 29236: 1, 2923
5: 1, 29234: 1, 29233: 1, 29232: 1, 29231: 1, 29230: 1, 29229: 1, 2922
8: 1, 29227: 1, 29226: 1, 29225: 1, 29224: 1, 29223: 1, 29222: 1, 2922
1: 1, 29220: 1, 29219: 1, 29218: 1, 29217: 1, 29216: 1, 29215: 1, 2921
4: 1, 29213: 1, 29212: 1, 29211: 1, 29210: 1, 29209: 1, 29208: 1, 2920
7: 1, 29206: 1, 29205: 1, 29204: 1, 29203: 1, 29202: 1, 29201: 1, 2920
0: 1, 29199: 1, 29198: 1, 29197: 1, 29196: 1, 29195: 1, 29194: 1, 2919
3: 1, 29192: 1, 29191: 1, 29190: 1, 29189: 1, 29188: 1, 29187: 1, 2918
6: 1, 29185: 1, 29184: 1, 29183: 1, 29182: 1, 29181: 1, 29180: 1, 2917

9: 1, 29178: 1, 29177: 1, 29176: 1, 29175: 1, 29174: 1, 29173: 1, 2917
2: 1, 29171: 1, 29170: 1, 29169: 1, 29168: 1, 29167: 1, 29166: 1, 2916
5: 1, 29164: 1, 29163: 1, 29162: 1, 29161: 1, 29160: 1, 29159: 1, 2915
8: 1, 29157: 1, 29156: 1, 29155: 1, 29154: 1, 29153: 1, 29152: 1, 2915
1: 1, 29150: 1, 29149: 1, 29148: 1, 29147: 1, 29146: 1, 29145: 1, 2914
4: 1, 29143: 1, 29142: 1, 29141: 1, 29140: 1, 29139: 1, 29138: 1, 2913
7: 1, 29136: 1, 29135: 1, 29134: 1, 29133: 1, 29132: 1, 29131: 1, 2913
0: 1, 29129: 1, 29128: 1, 29127: 1, 29126: 1, 29125: 1, 29124: 1, 2912
3: 1, 29122: 1, 29121: 1, 29120: 1, 29119: 1, 29118: 1, 29117: 1, 2911
6: 1, 29115: 1, 29114: 1, 29113: 1, 29112: 1, 29111: 1, 29110: 1, 2910
9: 1, 29108: 1, 29107: 1, 29106: 1, 29105: 1, 29104: 1, 29103: 1, 2910
2: 1, 29101: 1, 29100: 1, 29099: 1, 29098: 1, 29097: 1, 29096: 1, 2909
5: 1, 29094: 1, 29093: 1, 29092: 1, 29091: 1, 29090: 1, 29089: 1, 2908
8: 1, 29087: 1, 29086: 1, 29085: 1, 29084: 1, 29083: 1, 29082: 1, 2908
1: 1, 29080: 1, 29079: 1, 29078: 1, 29077: 1, 29076: 1, 29075: 1, 2907
4: 1, 29073: 1, 29072: 1, 29071: 1, 29070: 1, 29069: 1, 29068: 1, 2906
7: 1, 29066: 1, 29065: 1, 29064: 1, 29063: 1, 29062: 1, 29061: 1, 2906
0: 1, 29059: 1, 29058: 1, 29057: 1, 29056: 1, 29055: 1, 29054: 1, 2905
3: 1, 29052: 1, 29051: 1, 29050: 1, 29049: 1, 29048: 1, 29047: 1, 2904
6: 1, 29045: 1, 29044: 1, 29043: 1, 29042: 1, 29041: 1, 29040: 1, 2903
9: 1, 29038: 1, 29037: 1, 29036: 1, 29035: 1, 29034: 1, 29033: 1, 2903
2: 1, 29031: 1, 29030: 1, 29029: 1, 29028: 1, 29027: 1, 29026: 1, 2902
5: 1, 29024: 1, 29023: 1, 29022: 1, 29021: 1, 29020: 1, 29019: 1, 2901
8: 1, 29017: 1, 29016: 1, 29015: 1, 29014: 1, 29013: 1, 29012: 1, 2901
1: 1, 29010: 1, 29009: 1, 29008: 1, 29007: 1, 29006: 1, 29005: 1, 2900
4: 1, 29003: 1, 29002: 1, 29001: 1, 29000: 1, 28999: 1, 28998: 1, 2899
7: 1, 28996: 1, 28995: 1, 28994: 1, 28993: 1, 28992: 1, 28991: 1, 2899
0: 1, 28989: 1, 28988: 1, 28987: 1, 28986: 1, 28985: 1, 28984: 1, 2898
3: 1, 28982: 1, 28981: 1, 28980: 1, 28979: 1, 28978: 1, 28977: 1, 2897
6: 1, 28975: 1, 28974: 1, 28973: 1, 28972: 1, 28971: 1, 28970: 1, 2896
9: 1, 28968: 1, 28967: 1, 28966: 1, 28965: 1, 28964: 1, 28963: 1, 2896
2: 1, 28961: 1, 28960: 1, 28959: 1, 28958: 1, 28957: 1, 28956: 1, 2895
5: 1, 28954: 1, 28953: 1, 28952: 1, 28951: 1, 28950: 1, 28949: 1, 2894
8: 1, 28947: 1, 28946: 1, 28945: 1, 28944: 1, 28943: 1, 28942: 1, 2894
1: 1, 28940: 1, 28939: 1, 28938: 1, 28937: 1, 28936: 1, 28935: 1, 2893
4: 1, 28933: 1, 28932: 1, 28931: 1, 28930: 1, 28929: 1, 28928: 1, 2892
7: 1, 28926: 1, 28925: 1, 28924: 1, 28923: 1, 28922: 1, 28921: 1, 2892
0: 1, 28919: 1, 28918: 1, 28917: 1, 28916: 1, 28915: 1, 28914: 1, 2891
3: 1, 28912: 1, 28911: 1, 28910: 1, 28909: 1, 28908: 1, 28907: 1, 2890

6: 1, 28905: 1, 28904: 1, 28903: 1, 28902: 1, 28901: 1, 28900: 1, 2889
9: 1, 28898: 1, 28897: 1, 28896: 1, 28895: 1, 28894: 1, 28893: 1, 2889
2: 1, 28891: 1, 28890: 1, 28889: 1, 28888: 1, 28887: 1, 28886: 1, 2888
5: 1, 28884: 1, 28883: 1, 28882: 1, 28881: 1, 28880: 1, 28879: 1, 2887
8: 1, 28877: 1, 28876: 1, 28875: 1, 28874: 1, 28873: 1, 28872: 1, 2887
1: 1, 28870: 1, 28869: 1, 28868: 1, 28867: 1, 28866: 1, 28865: 1, 2886
4: 1, 28863: 1, 28862: 1, 28861: 1, 28860: 1, 28859: 1, 28858: 1, 2885
7: 1, 28856: 1, 28855: 1, 28854: 1, 28853: 1, 28852: 1, 28851: 1, 2885
0: 1, 28849: 1, 28848: 1, 28847: 1, 28846: 1, 28845: 1, 28844: 1, 2884
3: 1, 28842: 1, 28841: 1, 28840: 1, 28839: 1, 28838: 1, 28837: 1, 2883
6: 1, 28835: 1, 28834: 1, 28833: 1, 28832: 1, 28831: 1, 28830: 1, 2882
9: 1, 28828: 1, 28827: 1, 28826: 1, 28825: 1, 28824: 1, 28823: 1, 2882
2: 1, 28821: 1, 28820: 1, 28819: 1, 28818: 1, 28817: 1, 28816: 1, 2881
5: 1, 28814: 1, 28813: 1, 28812: 1, 28811: 1, 28810: 1, 28809: 1, 2880
8: 1, 28807: 1, 28806: 1, 28805: 1, 28804: 1, 28803: 1, 28802: 1, 2880
1: 1, 28800: 1, 28799: 1, 28798: 1, 28797: 1, 28796: 1, 28795: 1, 2879
4: 1, 28793: 1, 28792: 1, 28791: 1, 28790: 1, 28789: 1, 28788: 1, 2878
7: 1, 28786: 1, 28785: 1, 28784: 1, 28783: 1, 28782: 1, 28781: 1, 2878
0: 1, 28779: 1, 28778: 1, 28777: 1, 28776: 1, 28775: 1, 28774: 1, 2877
3: 1, 28772: 1, 28771: 1, 28770: 1, 28769: 1, 28768: 1, 28767: 1, 2876
6: 1, 28765: 1, 28764: 1, 28763: 1, 28762: 1, 28761: 1, 28760: 1, 2875
9: 1, 28758: 1, 28757: 1, 28756: 1, 28755: 1, 28754: 1, 28753: 1, 2875
2: 1, 28751: 1, 28750: 1, 28749: 1, 28748: 1, 28747: 1, 28746: 1, 2874
5: 1, 28744: 1, 28743: 1, 28742: 1, 28741: 1, 28740: 1, 28739: 1, 2873
8: 1, 28737: 1, 28736: 1, 28735: 1, 28734: 1, 28733: 1, 28732: 1, 2873
1: 1, 28730: 1, 28729: 1, 28728: 1, 28727: 1, 28726: 1, 28725: 1, 2872
4: 1, 28723: 1, 28722: 1, 28721: 1, 28720: 1, 28719: 1, 28718: 1, 2871
7: 1, 28716: 1, 28715: 1, 28714: 1, 28713: 1, 28712: 1, 28711: 1, 2871
0: 1, 28709: 1, 28708: 1, 28707: 1, 28706: 1, 28705: 1, 28704: 1, 2870
3: 1, 28702: 1, 28701: 1, 28700: 1, 28699: 1, 28698: 1, 28697: 1, 2869
6: 1, 28695: 1, 28694: 1, 28693: 1, 28692: 1, 28691: 1, 28690: 1, 2868
9: 1, 28688: 1, 28687: 1, 28686: 1, 28685: 1, 28684: 1, 28683: 1, 2868
2: 1, 28681: 1, 28680: 1, 28679: 1, 28678: 1, 28677: 1, 28676: 1, 2867
5: 1, 28674: 1, 28673: 1, 28672: 1, 28671: 1, 28670: 1, 28669: 1, 2866
8: 1, 28667: 1, 28666: 1, 28665: 1, 28664: 1, 28663: 1, 28662: 1, 2866
1: 1, 28660: 1, 28659: 1, 28658: 1, 28657: 1, 28656: 1, 28655: 1, 2865
4: 1, 28653: 1, 28652: 1, 28651: 1, 28650: 1, 28649: 1, 28648: 1, 2864
7: 1, 28646: 1, 28645: 1, 28644: 1, 28643: 1, 28642: 1, 28641: 1, 2864
0: 1, 28639: 1, 28638: 1, 28637: 1, 28636: 1, 28635: 1, 28634: 1, 2863

3: 1, 28632: 1, 28631: 1, 28630: 1, 28629: 1, 28628: 1, 28627: 1, 2862
6: 1, 28625: 1, 28624: 1, 28623: 1, 28622: 1, 28621: 1, 28620: 1, 2861
9: 1, 28618: 1, 28617: 1, 28616: 1, 28615: 1, 28614: 1, 28613: 1, 2861
2: 1, 28611: 1, 28610: 1, 28609: 1, 28608: 1, 28607: 1, 28606: 1, 2860
5: 1, 28604: 1, 28603: 1, 28602: 1, 28601: 1, 28600: 1, 28599: 1, 2859
8: 1, 28597: 1, 28596: 1, 28595: 1, 28594: 1, 28593: 1, 28592: 1, 2859
1: 1, 28590: 1, 28589: 1, 28588: 1, 28587: 1, 28586: 1, 28585: 1, 2858
4: 1, 28583: 1, 28582: 1, 28581: 1, 28580: 1, 28579: 1, 28578: 1, 2857
7: 1, 28576: 1, 28575: 1, 28574: 1, 28573: 1, 28572: 1, 28571: 1, 2857
0: 1, 28569: 1, 28568: 1, 28567: 1, 28566: 1, 28565: 1, 28564: 1, 2856
3: 1, 28562: 1, 28561: 1, 28560: 1, 28559: 1, 28558: 1, 28557: 1, 2855
6: 1, 28555: 1, 28554: 1, 28553: 1, 28552: 1, 28551: 1, 28550: 1, 2854
9: 1, 28548: 1, 28547: 1, 28546: 1, 28545: 1, 28544: 1, 28543: 1, 2854
2: 1, 28541: 1, 28540: 1, 28539: 1, 28538: 1, 28537: 1, 28536: 1, 2853
5: 1, 28534: 1, 28533: 1, 28532: 1, 28531: 1, 28530: 1, 28529: 1, 2852
8: 1, 28527: 1, 28526: 1, 28525: 1, 28524: 1, 28523: 1, 28522: 1, 2852
1: 1, 28520: 1, 28519: 1, 28518: 1, 28517: 1, 28516: 1, 28515: 1, 2851
4: 1, 28513: 1, 28512: 1, 28511: 1, 28510: 1, 28509: 1, 28508: 1, 2850
7: 1, 28506: 1, 28505: 1, 28504: 1, 28503: 1, 28502: 1, 28501: 1, 2850
0: 1, 28499: 1, 28498: 1, 28497: 1, 28496: 1, 28495: 1, 28494: 1, 2849
3: 1, 28492: 1, 28491: 1, 28490: 1, 28489: 1, 28488: 1, 28487: 1, 2848
6: 1, 28485: 1, 28484: 1, 28483: 1, 28482: 1, 28481: 1, 28480: 1, 2847
9: 1, 28478: 1, 28477: 1, 28476: 1, 28475: 1, 28474: 1, 28473: 1, 2847
2: 1, 28471: 1, 28470: 1, 28469: 1, 28468: 1, 28467: 1, 28466: 1, 2846
5: 1, 28464: 1, 28463: 1, 28462: 1, 28461: 1, 28460: 1, 28459: 1, 2845
8: 1, 28457: 1, 28456: 1, 28455: 1, 28454: 1, 28453: 1, 28452: 1, 2845
1: 1, 28450: 1, 28449: 1, 28448: 1, 28447: 1, 28446: 1, 28445: 1, 2844
4: 1, 28443: 1, 28442: 1, 28441: 1, 28440: 1, 28439: 1, 28438: 1, 2843
7: 1, 28436: 1, 28435: 1, 28434: 1, 28433: 1, 28432: 1, 28431: 1, 2843
0: 1, 28429: 1, 28428: 1, 28427: 1, 28426: 1, 28425: 1, 28424: 1, 2842
3: 1, 28422: 1, 28421: 1, 28420: 1, 28419: 1, 28418: 1, 28417: 1, 2841
6: 1, 28415: 1, 28414: 1, 28413: 1, 28412: 1, 28411: 1, 28410: 1, 2840
9: 1, 28408: 1, 28407: 1, 28406: 1, 28405: 1, 28404: 1, 28403: 1, 2840
2: 1, 28401: 1, 28400: 1, 28399: 1, 28398: 1, 28397: 1, 28396: 1, 2839
5: 1, 28394: 1, 28393: 1, 28392: 1, 28391: 1, 28390: 1, 28389: 1, 2838
8: 1, 28387: 1, 28386: 1, 28385: 1, 28384: 1, 28383: 1, 28382: 1, 2838
1: 1, 28380: 1, 28379: 1, 28378: 1, 28377: 1, 28376: 1, 28375: 1, 2837
4: 1, 28373: 1, 28372: 1, 28371: 1, 28370: 1, 28369: 1, 28368: 1, 2836
7: 1, 28366: 1, 28365: 1, 28364: 1, 28363: 1, 28362: 1, 28361: 1, 2836

0: 1, 28359: 1, 28358: 1, 28357: 1, 28356: 1, 28355: 1, 28354: 1, 2835
3: 1, 28352: 1, 28351: 1, 28350: 1, 28349: 1, 28348: 1, 28347: 1, 2834
6: 1, 28345: 1, 28344: 1, 28343: 1, 28342: 1, 28341: 1, 28340: 1, 2833
9: 1, 28338: 1, 28337: 1, 28336: 1, 28335: 1, 28334: 1, 28333: 1, 2833
2: 1, 28331: 1, 28330: 1, 28329: 1, 28328: 1, 28327: 1, 28326: 1, 2832
5: 1, 28324: 1, 28323: 1, 28322: 1, 28321: 1, 28320: 1, 28319: 1, 2831
8: 1, 28317: 1, 28316: 1, 28315: 1, 28314: 1, 28313: 1, 28312: 1, 2831
1: 1, 28310: 1, 28309: 1, 28308: 1, 28307: 1, 28306: 1, 28305: 1, 2830
4: 1, 28303: 1, 28302: 1, 28301: 1, 28300: 1, 28299: 1, 28298: 1, 2829
7: 1, 28296: 1, 28295: 1, 28294: 1, 28293: 1, 28292: 1, 28291: 1, 2829
0: 1, 28289: 1, 28288: 1, 28287: 1, 28286: 1, 28285: 1, 28284: 1, 2828
3: 1, 28282: 1, 28281: 1, 28280: 1, 28279: 1, 28278: 1, 28277: 1, 2827
6: 1, 28275: 1, 28274: 1, 28273: 1, 28272: 1, 28271: 1, 28270: 1, 2826
9: 1, 28268: 1, 28267: 1, 28266: 1, 28265: 1, 28264: 1, 28263: 1, 2826
2: 1, 28261: 1, 28260: 1, 28259: 1, 28258: 1, 28257: 1, 28256: 1, 2825
5: 1, 28254: 1, 28253: 1, 28252: 1, 28251: 1, 28250: 1, 28249: 1, 2824
8: 1, 28247: 1, 28246: 1, 28245: 1, 28244: 1, 28243: 1, 28242: 1, 2824
1: 1, 28240: 1, 28239: 1, 28238: 1, 28237: 1, 28236: 1, 28235: 1, 2823
4: 1, 28233: 1, 28232: 1, 28231: 1, 28230: 1, 28229: 1, 28228: 1, 2822
7: 1, 28226: 1, 28225: 1, 28224: 1, 28223: 1, 28222: 1, 28221: 1, 2822
0: 1, 28219: 1, 28218: 1, 28217: 1, 28216: 1, 28215: 1, 28214: 1, 2821
3: 1, 28212: 1, 28211: 1, 28210: 1, 28209: 1, 28208: 1, 28207: 1, 2820
6: 1, 28205: 1, 28204: 1, 28203: 1, 28202: 1, 28201: 1, 28200: 1, 2819
9: 1, 28198: 1, 28197: 1, 28196: 1, 28195: 1, 28194: 1, 28193: 1, 2819
2: 1, 28191: 1, 28190: 1, 28189: 1, 28188: 1, 28187: 1, 28186: 1, 2818
5: 1, 28184: 1, 28183: 1, 28182: 1, 28181: 1, 28180: 1, 28179: 1, 2817
8: 1, 28177: 1, 28176: 1, 28175: 1, 28174: 1, 28173: 1, 28172: 1, 2817
1: 1, 28170: 1, 28169: 1, 28168: 1, 28167: 1, 28166: 1, 28165: 1, 2816
4: 1, 28163: 1, 28162: 1, 28161: 1, 28160: 1, 28159: 1, 28158: 1, 2815
7: 1, 28156: 1, 28155: 1, 28154: 1, 28153: 1, 28152: 1, 28151: 1, 2815
0: 1, 28149: 1, 28148: 1, 28147: 1, 28146: 1, 28145: 1, 28144: 1, 2814
3: 1, 28142: 1, 28141: 1, 28140: 1, 28139: 1, 28138: 1, 28137: 1, 2813
6: 1, 28135: 1, 28134: 1, 28133: 1, 28132: 1, 28131: 1, 28130: 1, 2812
9: 1, 28128: 1, 28127: 1, 28126: 1, 28125: 1, 28124: 1, 28123: 1, 2812
2: 1, 28121: 1, 28120: 1, 28119: 1, 28118: 1, 28117: 1, 28116: 1, 2811
5: 1, 28114: 1, 28113: 1, 28112: 1, 28111: 1, 28110: 1, 28109: 1, 2810
8: 1, 28107: 1, 28106: 1, 28105: 1, 28104: 1, 28103: 1, 28102: 1, 2810
1: 1, 28100: 1, 28099: 1, 28098: 1, 28097: 1, 28096: 1, 28095: 1, 2809
4: 1, 28093: 1, 28092: 1, 28091: 1, 28090: 1, 28089: 1, 28088: 1, 2808

7: 1, 28086: 1, 28085: 1, 28084: 1, 28083: 1, 28082: 1, 28081: 1, 2808
0: 1, 28079: 1, 28078: 1, 28077: 1, 28076: 1, 28075: 1, 28074: 1, 2807
3: 1, 28072: 1, 28071: 1, 28070: 1, 28069: 1, 28068: 1, 28067: 1, 2806
6: 1, 28065: 1, 28064: 1, 28063: 1, 28062: 1, 28061: 1, 28060: 1, 2805
9: 1, 28058: 1, 28057: 1, 28056: 1, 28055: 1, 28054: 1, 28053: 1, 2805
2: 1, 28051: 1, 28050: 1, 28049: 1, 28048: 1, 28047: 1, 28046: 1, 2804
5: 1, 28044: 1, 28043: 1, 28042: 1, 28041: 1, 28040: 1, 28039: 1, 2803
8: 1, 28037: 1, 28036: 1, 28035: 1, 28034: 1, 28033: 1, 28032: 1, 2803
1: 1, 28030: 1, 28029: 1, 28028: 1, 28027: 1, 28026: 1, 28025: 1, 2802
4: 1, 28023: 1, 28022: 1, 28021: 1, 28020: 1, 28019: 1, 28018: 1, 2801
7: 1, 28016: 1, 28015: 1, 28014: 1, 28013: 1, 28012: 1, 28011: 1, 2801
0: 1, 28009: 1, 28008: 1, 28007: 1, 28006: 1, 28005: 1, 28004: 1, 2800
3: 1, 28002: 1, 28001: 1, 28000: 1, 27999: 1, 27998: 1, 27997: 1, 2799
6: 1, 27995: 1, 27994: 1, 27993: 1, 27992: 1, 27991: 1, 27990: 1, 2798
9: 1, 27988: 1, 27987: 1, 27986: 1, 27985: 1, 27984: 1, 27983: 1, 2798
2: 1, 27981: 1, 27980: 1, 27979: 1, 27978: 1, 27977: 1, 27976: 1, 2797
5: 1, 27974: 1, 27973: 1, 27972: 1, 27971: 1, 27970: 1, 27969: 1, 2796
8: 1, 27967: 1, 27966: 1, 27965: 1, 27964: 1, 27963: 1, 27962: 1, 2796
1: 1, 27960: 1, 27959: 1, 27958: 1, 27957: 1, 27956: 1, 27955: 1, 2795
4: 1, 27953: 1, 27952: 1, 27951: 1, 27950: 1, 27949: 1, 27948: 1, 2794
7: 1, 27946: 1, 27945: 1, 27944: 1, 27943: 1, 27942: 1, 27941: 1, 2794
0: 1, 27939: 1, 27938: 1, 27937: 1, 27936: 1, 27935: 1, 27934: 1, 2793
3: 1, 27932: 1, 27931: 1, 27930: 1, 27929: 1, 27928: 1, 27927: 1, 2792
6: 1, 27925: 1, 27924: 1, 27923: 1, 27922: 1, 27921: 1, 27920: 1, 2791
9: 1, 27918: 1, 27917: 1, 27916: 1, 27915: 1, 27914: 1, 27913: 1, 2791
2: 1, 27911: 1, 27910: 1, 27909: 1, 27908: 1, 27907: 1, 27906: 1, 2790
5: 1, 27904: 1, 27903: 1, 27902: 1, 27901: 1, 27900: 1, 27899: 1, 2789
8: 1, 27897: 1, 27896: 1, 27895: 1, 27894: 1, 27893: 1, 27892: 1, 2789
1: 1, 27890: 1, 27889: 1, 27888: 1, 27887: 1, 27886: 1, 27885: 1, 2788
4: 1, 27883: 1, 27882: 1, 27881: 1, 27880: 1, 27879: 1, 27878: 1, 2787
7: 1, 27876: 1, 27875: 1, 27874: 1, 27873: 1, 27872: 1, 27871: 1, 2787
0: 1, 27869: 1, 27868: 1, 27867: 1, 27866: 1, 27865: 1, 27864: 1, 2786
3: 1, 27862: 1, 27861: 1, 27860: 1, 27859: 1, 27858: 1, 27857: 1, 2785
6: 1, 27855: 1, 27854: 1, 27853: 1, 27852: 1, 27851: 1, 27850: 1, 2784
9: 1, 27848: 1, 27847: 1, 27846: 1, 27845: 1, 27844: 1, 27843: 1, 2784
2: 1, 27841: 1, 27840: 1, 27839: 1, 27838: 1, 27837: 1, 27836: 1, 2783
5: 1, 27834: 1, 27833: 1, 27832: 1, 27831: 1, 27830: 1, 27829: 1, 2782
8: 1, 27827: 1, 27826: 1, 27825: 1, 27824: 1, 27823: 1, 27822: 1, 2782
1: 1, 27820: 1, 27819: 1, 27818: 1, 27817: 1, 27816: 1, 27815: 1, 2781

4: 1, 27813: 1, 27812: 1, 27811: 1, 27810: 1, 27809: 1, 27808: 1, 2780
7: 1, 27806: 1, 27805: 1, 27804: 1, 27803: 1, 27802: 1, 27801: 1, 2780
0: 1, 27799: 1, 27798: 1, 27797: 1, 27796: 1, 27795: 1, 27794: 1, 2779
3: 1, 27792: 1, 27791: 1, 27790: 1, 27789: 1, 27788: 1, 27787: 1, 2778
6: 1, 27785: 1, 27784: 1, 27783: 1, 27782: 1, 27781: 1, 27780: 1, 2777
9: 1, 27778: 1, 27777: 1, 27776: 1, 27775: 1, 27774: 1, 27773: 1, 2777
2: 1, 27771: 1, 27770: 1, 27769: 1, 27768: 1, 27767: 1, 27766: 1, 2776
5: 1, 27764: 1, 27763: 1, 27762: 1, 27761: 1, 27760: 1, 27759: 1, 2775
8: 1, 27757: 1, 27756: 1, 27755: 1, 27754: 1, 27753: 1, 27752: 1, 2775
1: 1, 27750: 1, 27749: 1, 27748: 1, 27747: 1, 27746: 1, 27745: 1, 2774
4: 1, 27743: 1, 27742: 1, 27741: 1, 27740: 1, 27739: 1, 27738: 1, 2773
7: 1, 27736: 1, 27735: 1, 27734: 1, 27733: 1, 27732: 1, 27731: 1, 2773
0: 1, 27729: 1, 27728: 1, 27727: 1, 27726: 1, 27725: 1, 27724: 1, 2772
3: 1, 27722: 1, 27721: 1, 27720: 1, 27719: 1, 27718: 1, 27717: 1, 2771
6: 1, 27715: 1, 27714: 1, 27713: 1, 27712: 1, 27711: 1, 27710: 1, 2770
9: 1, 27708: 1, 27707: 1, 27706: 1, 27705: 1, 27704: 1, 27703: 1, 2770
2: 1, 27701: 1, 27700: 1, 27699: 1, 27698: 1, 27697: 1, 27696: 1, 2769
5: 1, 27694: 1, 27693: 1, 27692: 1, 27691: 1, 27690: 1, 27689: 1, 2768
8: 1, 27687: 1, 27686: 1, 27685: 1, 27684: 1, 27683: 1, 27682: 1, 2768
1: 1, 27680: 1, 27679: 1, 27678: 1, 27677: 1, 27676: 1, 27675: 1, 2767
4: 1, 27673: 1, 27672: 1, 27671: 1, 27670: 1, 27669: 1, 27668: 1, 2766
7: 1, 27666: 1, 27665: 1, 27664: 1, 27663: 1, 27662: 1, 27661: 1, 2766
0: 1, 27659: 1, 27658: 1, 27657: 1, 27656: 1, 27655: 1, 27654: 1, 2765
3: 1, 27652: 1, 27651: 1, 27650: 1, 27649: 1, 27648: 1, 27647: 1, 2764
6: 1, 27645: 1, 27644: 1, 27643: 1, 27642: 1, 27641: 1, 27640: 1, 2763
9: 1, 27638: 1, 27637: 1, 27636: 1, 27635: 1, 27634: 1, 27633: 1, 2763
2: 1, 27631: 1, 27630: 1, 27629: 1, 27628: 1, 27627: 1, 27626: 1, 2762
5: 1, 27624: 1, 27623: 1, 27622: 1, 27621: 1, 27620: 1, 27619: 1, 2761
8: 1, 27617: 1, 27616: 1, 27615: 1, 27614: 1, 27613: 1, 27612: 1, 2761
1: 1, 27610: 1, 27609: 1, 27608: 1, 27607: 1, 27606: 1, 27605: 1, 2760
4: 1, 27603: 1, 27602: 1, 27601: 1, 27600: 1, 27599: 1, 27598: 1, 2759
7: 1, 27596: 1, 27595: 1, 27594: 1, 27593: 1, 27592: 1, 27591: 1, 2759
0: 1, 27589: 1, 27588: 1, 27587: 1, 27586: 1, 27585: 1, 27584: 1, 2758
3: 1, 27582: 1, 27581: 1, 27580: 1, 27579: 1, 27578: 1, 27577: 1, 2757
6: 1, 27575: 1, 27574: 1, 27573: 1, 27572: 1, 27571: 1, 27570: 1, 2756
9: 1, 27568: 1, 27567: 1, 27566: 1, 27565: 1, 27564: 1, 27563: 1, 2756
2: 1, 27561: 1, 27560: 1, 27559: 1, 27558: 1, 27557: 1, 27556: 1, 2755
5: 1, 27554: 1, 27553: 1, 27552: 1, 27551: 1, 27550: 1, 27549: 1, 2754
8: 1, 27547: 1, 27546: 1, 27545: 1, 27544: 1, 27543: 1, 27542: 1, 2754

1: 1, 27540: 1, 27539: 1, 27538: 1, 27537: 1, 27536: 1, 27535: 1, 2753
4: 1, 27533: 1, 27532: 1, 27531: 1, 27530: 1, 27529: 1, 27528: 1, 2752
7: 1, 27526: 1, 27525: 1, 27524: 1, 27523: 1, 27522: 1, 27521: 1, 2752
0: 1, 27519: 1, 27518: 1, 27517: 1, 27516: 1, 27515: 1, 27514: 1, 2751
3: 1, 27512: 1, 27511: 1, 27510: 1, 27509: 1, 27508: 1, 27507: 1, 2750
6: 1, 27505: 1, 27504: 1, 27503: 1, 27502: 1, 27501: 1, 27500: 1, 2749
9: 1, 27498: 1, 27497: 1, 27496: 1, 27495: 1, 27494: 1, 27493: 1, 2749
2: 1, 27491: 1, 27490: 1, 27489: 1, 27488: 1, 27487: 1, 27486: 1, 2748
5: 1, 27484: 1, 27483: 1, 27482: 1, 27481: 1, 27480: 1, 27479: 1, 2747
8: 1, 27477: 1, 27476: 1, 27475: 1, 27474: 1, 27473: 1, 27472: 1, 2747
1: 1, 27470: 1, 27469: 1, 27468: 1, 27467: 1, 27466: 1, 27465: 1, 2746
4: 1, 27463: 1, 27462: 1, 27461: 1, 27460: 1, 27459: 1, 27458: 1, 2745
7: 1, 27456: 1, 27455: 1, 27454: 1, 27453: 1, 27452: 1, 27451: 1, 2745
0: 1, 27449: 1, 27448: 1, 27447: 1, 27446: 1, 27445: 1, 27444: 1, 2744
3: 1, 27442: 1, 27441: 1, 27440: 1, 27439: 1, 27438: 1, 27437: 1, 2743
6: 1, 27435: 1, 27434: 1, 27433: 1, 27432: 1, 27431: 1, 27430: 1, 2742
9: 1, 27428: 1, 27427: 1, 27426: 1, 27425: 1, 27424: 1, 27423: 1, 2742
2: 1, 27421: 1, 27420: 1, 27419: 1, 27418: 1, 27417: 1, 27416: 1, 2741
5: 1, 27414: 1, 27413: 1, 27412: 1, 27411: 1, 27410: 1, 27409: 1, 2740
8: 1, 27407: 1, 27406: 1, 27405: 1, 27404: 1, 27403: 1, 27402: 1, 2740
1: 1, 27400: 1, 27399: 1, 27398: 1, 27397: 1, 27396: 1, 27395: 1, 2739
4: 1, 27393: 1, 27392: 1, 27391: 1, 27390: 1, 27389: 1, 27388: 1, 2738
7: 1, 27386: 1, 27385: 1, 27384: 1, 27383: 1, 27382: 1, 27381: 1, 2738
0: 1, 27379: 1, 27378: 1, 27377: 1, 27376: 1, 27375: 1, 27374: 1, 2737
3: 1, 27372: 1, 27371: 1, 27370: 1, 27369: 1, 27368: 1, 27367: 1, 2736
6: 1, 27365: 1, 27364: 1, 27363: 1, 27362: 1, 27361: 1, 27360: 1, 2735
9: 1, 27358: 1, 27357: 1, 27356: 1, 27355: 1, 27354: 1, 27353: 1, 2735
2: 1, 27351: 1, 27350: 1, 27349: 1, 27348: 1, 27347: 1, 27346: 1, 2734
5: 1, 27344: 1, 27343: 1, 27342: 1, 27341: 1, 27340: 1, 27339: 1, 2733
8: 1, 27337: 1, 27336: 1, 27335: 1, 27334: 1, 27333: 1, 27332: 1, 2733
1: 1, 27330: 1, 27329: 1, 27328: 1, 27327: 1, 27326: 1, 27325: 1, 2732
4: 1, 27323: 1, 27322: 1, 27321: 1, 27320: 1, 27319: 1, 27318: 1, 2731
7: 1, 27316: 1, 27315: 1, 27314: 1, 27313: 1, 27312: 1, 27311: 1, 2731
0: 1, 27309: 1, 27308: 1, 27307: 1, 27306: 1, 27305: 1, 27304: 1, 2730
3: 1, 27302: 1, 27301: 1, 27300: 1, 27299: 1, 27298: 1, 27297: 1, 2729
6: 1, 27295: 1, 27294: 1, 27293: 1, 27292: 1, 27291: 1, 27290: 1, 2728
9: 1, 27288: 1, 27287: 1, 27286: 1, 27285: 1, 27284: 1, 27283: 1, 2728
2: 1, 27281: 1, 27280: 1, 27279: 1, 27278: 1, 27277: 1, 27276: 1, 2727
5: 1, 27274: 1, 27273: 1, 27272: 1, 27271: 1, 27270: 1, 27269: 1, 2726

8: 1, 27267: 1, 27266: 1, 27265: 1, 27264: 1, 27263: 1, 27262: 1, 2726
1: 1, 27260: 1, 27259: 1, 27258: 1, 27257: 1, 27256: 1, 27255: 1, 2725
4: 1, 27253: 1, 27252: 1, 27251: 1, 27250: 1, 27249: 1, 27248: 1, 2724
7: 1, 27246: 1, 27245: 1, 27244: 1, 27243: 1, 27242: 1, 27241: 1, 2724
0: 1, 27239: 1, 27238: 1, 27237: 1, 27236: 1, 27235: 1, 27234: 1, 2723
3: 1, 27232: 1, 27231: 1, 27230: 1, 27229: 1, 27228: 1, 27227: 1, 2722
6: 1, 27225: 1, 27224: 1, 27223: 1, 27222: 1, 27221: 1, 27220: 1, 2721
9: 1, 27218: 1, 27217: 1, 27216: 1, 27215: 1, 27214: 1, 27213: 1, 2721
2: 1, 27211: 1, 27210: 1, 27209: 1, 27208: 1, 27207: 1, 27206: 1, 2720
5: 1, 27204: 1, 27203: 1, 27202: 1, 27201: 1, 27200: 1, 27199: 1, 2719
8: 1, 27197: 1, 27196: 1, 27195: 1, 27194: 1, 27193: 1, 27192: 1, 2719
1: 1, 27190: 1, 27189: 1, 27188: 1, 27187: 1, 27186: 1, 27185: 1, 2718
4: 1, 27183: 1, 27182: 1, 27181: 1, 27180: 1, 27179: 1, 27178: 1, 2717
7: 1, 27176: 1, 27175: 1, 27174: 1, 27173: 1, 27172: 1, 27171: 1, 2717
0: 1, 27169: 1, 27168: 1, 27167: 1, 27166: 1, 27165: 1, 27164: 1, 2716
3: 1, 27162: 1, 27161: 1, 27160: 1, 27159: 1, 27158: 1, 27157: 1, 2715
6: 1, 27155: 1, 27154: 1, 27153: 1, 27152: 1, 27151: 1, 27150: 1, 2714
9: 1, 27148: 1, 27147: 1, 27146: 1, 27145: 1, 27144: 1, 27143: 1, 2714
2: 1, 27141: 1, 27140: 1, 27139: 1, 27138: 1, 27137: 1, 27136: 1, 2713
5: 1, 27134: 1, 27133: 1, 27132: 1, 27131: 1, 27130: 1, 27129: 1, 2712
8: 1, 27127: 1, 27126: 1, 27125: 1, 27124: 1, 27123: 1, 27122: 1, 2712
1: 1, 27120: 1, 27119: 1, 27118: 1, 27117: 1, 27116: 1, 27115: 1, 2711
4: 1, 27113: 1, 27112: 1, 27111: 1, 27110: 1, 27109: 1, 27108: 1, 2710
7: 1, 27106: 1, 27105: 1, 27104: 1, 27103: 1, 27102: 1, 27101: 1, 2710
0: 1, 27099: 1, 27098: 1, 27097: 1, 27096: 1, 27095: 1, 27094: 1, 2709
3: 1, 27092: 1, 27091: 1, 27090: 1, 27089: 1, 27088: 1, 27087: 1, 2708
6: 1, 27085: 1, 27084: 1, 27083: 1, 27082: 1, 27081: 1, 27080: 1, 2707
9: 1, 27078: 1, 27077: 1, 27076: 1, 27075: 1, 27074: 1, 27073: 1, 2707
2: 1, 27071: 1, 27070: 1, 27069: 1, 27068: 1, 27067: 1, 27066: 1, 2706
5: 1, 27064: 1, 27063: 1, 27062: 1, 27061: 1, 27060: 1, 27059: 1, 2705
8: 1, 27057: 1, 27056: 1, 27055: 1, 27054: 1, 27053: 1, 27052: 1, 2705
1: 1, 27050: 1, 27049: 1, 27048: 1, 27047: 1, 27046: 1, 27045: 1, 2704
4: 1, 27043: 1, 27042: 1, 27041: 1, 27040: 1, 27039: 1, 27038: 1, 2703
7: 1, 27036: 1, 27035: 1, 27034: 1, 27033: 1, 27032: 1, 27031: 1, 2703
0: 1, 27029: 1, 27028: 1, 27027: 1, 27026: 1, 27025: 1, 27024: 1, 2702
3: 1, 27022: 1, 27021: 1, 27020: 1, 27019: 1, 27018: 1, 27017: 1, 2701
6: 1, 27015: 1, 27014: 1, 27013: 1, 27012: 1, 27011: 1, 27010: 1, 2700
9: 1, 27008: 1, 27007: 1, 27006: 1, 27005: 1, 27004: 1, 27003: 1, 2700
2: 1, 27001: 1, 27000: 1, 26999: 1, 26998: 1, 26997: 1, 26996: 1, 2699

5: 1, 26994: 1, 26993: 1, 26992: 1, 26991: 1, 26990: 1, 26989: 1, 2698
8: 1, 26987: 1, 26986: 1, 26985: 1, 26984: 1, 26983: 1, 26982: 1, 2698
1: 1, 26980: 1, 26979: 1, 26978: 1, 26977: 1, 26976: 1, 26975: 1, 2697
4: 1, 26973: 1, 26972: 1, 26971: 1, 26970: 1, 26969: 1, 26968: 1, 2696
7: 1, 26966: 1, 26965: 1, 26964: 1, 26963: 1, 26962: 1, 26961: 1, 2696
0: 1, 26959: 1, 26958: 1, 26957: 1, 26956: 1, 26955: 1, 26954: 1, 2695
3: 1, 26952: 1, 26951: 1, 26950: 1, 26949: 1, 26948: 1, 26947: 1, 2694
6: 1, 26945: 1, 26944: 1, 26943: 1, 26942: 1, 26941: 1, 26940: 1, 2693
9: 1, 26938: 1, 26937: 1, 26936: 1, 26935: 1, 26934: 1, 26933: 1, 2693
2: 1, 26931: 1, 26930: 1, 26929: 1, 26928: 1, 26927: 1, 26926: 1, 2692
5: 1, 26924: 1, 26923: 1, 26922: 1, 26921: 1, 26920: 1, 26919: 1, 2691
8: 1, 26917: 1, 26916: 1, 26915: 1, 26914: 1, 26913: 1, 26912: 1, 2691
1: 1, 26910: 1, 26909: 1, 26908: 1, 26907: 1, 26906: 1, 26905: 1, 2690
4: 1, 26903: 1, 26902: 1, 26901: 1, 26900: 1, 26899: 1, 26898: 1, 2689
7: 1, 26896: 1, 26895: 1, 26894: 1, 26893: 1, 26892: 1, 26891: 1, 2689
0: 1, 26889: 1, 26888: 1, 26887: 1, 26886: 1, 26885: 1, 26884: 1, 2688
3: 1, 26882: 1, 26881: 1, 26880: 1, 26879: 1, 26878: 1, 26877: 1, 2687
6: 1, 26875: 1, 26874: 1, 26873: 1, 26872: 1, 26871: 1, 26870: 1, 2686
9: 1, 26868: 1, 26867: 1, 26866: 1, 26865: 1, 26864: 1, 26863: 1, 2686
2: 1, 26861: 1, 26860: 1, 26859: 1, 26858: 1, 26857: 1, 26856: 1, 2685
5: 1, 26854: 1, 26853: 1, 26852: 1, 26851: 1, 26850: 1, 26849: 1, 2684
8: 1, 26847: 1, 26846: 1, 26845: 1, 26844: 1, 26843: 1, 26842: 1, 2684
1: 1, 26840: 1, 26839: 1, 26838: 1, 26837: 1, 26836: 1, 26835: 1, 2683
4: 1, 26833: 1, 26832: 1, 26831: 1, 26830: 1, 26829: 1, 26828: 1, 2682
7: 1, 26826: 1, 26825: 1, 26824: 1, 26823: 1, 26822: 1, 26821: 1, 2682
0: 1, 26819: 1, 26818: 1, 26817: 1, 26816: 1, 26815: 1, 26814: 1, 2681
3: 1, 26812: 1, 26811: 1, 26810: 1, 26809: 1, 26808: 1, 26807: 1, 2680
6: 1, 26805: 1, 26804: 1, 26803: 1, 26802: 1, 26801: 1, 26800: 1, 2679
9: 1, 26798: 1, 26797: 1, 26796: 1, 26795: 1, 26794: 1, 26793: 1, 2679
2: 1, 26791: 1, 26790: 1, 26789: 1, 26788: 1, 26787: 1, 26786: 1, 2678
5: 1, 26784: 1, 26783: 1, 26782: 1, 26781: 1, 26780: 1, 26779: 1, 2677
8: 1, 26777: 1, 26776: 1, 26775: 1, 26774: 1, 26773: 1, 26772: 1, 2677
1: 1, 26770: 1, 26769: 1, 26768: 1, 26767: 1, 26766: 1, 26765: 1, 2676
4: 1, 26763: 1, 26762: 1, 26761: 1, 26760: 1, 26759: 1, 26758: 1, 2675
7: 1, 26756: 1, 26755: 1, 26754: 1, 26753: 1, 26752: 1, 26751: 1, 2675
0: 1, 26749: 1, 26748: 1, 26747: 1, 26746: 1, 26745: 1, 26744: 1, 2674
3: 1, 26742: 1, 26741: 1, 26740: 1, 26739: 1, 26738: 1, 26737: 1, 2673
6: 1, 26735: 1, 26734: 1, 26733: 1, 26732: 1, 26731: 1, 26730: 1, 2672
9: 1, 26728: 1, 26727: 1, 26726: 1, 26725: 1, 26724: 1, 26723: 1, 2672

2: 1, 26721: 1, 26720: 1, 26719: 1, 26718: 1, 26717: 1, 26716: 1, 2671
5: 1, 26714: 1, 26713: 1, 26712: 1, 26711: 1, 26710: 1, 26709: 1, 2670
8: 1, 26707: 1, 26706: 1, 26705: 1, 26704: 1, 26703: 1, 26702: 1, 2670
1: 1, 26700: 1, 26699: 1, 26698: 1, 26697: 1, 26696: 1, 26695: 1, 2669
4: 1, 26693: 1, 26692: 1, 26691: 1, 26690: 1, 26689: 1, 26688: 1, 2668
7: 1, 26686: 1, 26685: 1, 26684: 1, 26683: 1, 26682: 1, 26681: 1, 2668
0: 1, 26679: 1, 26678: 1, 26677: 1, 26676: 1, 26675: 1, 26674: 1, 2667
3: 1, 26672: 1, 26671: 1, 26670: 1, 26669: 1, 26668: 1, 26667: 1, 2666
6: 1, 26665: 1, 26664: 1, 26663: 1, 26662: 1, 26661: 1, 26660: 1, 2665
9: 1, 26658: 1, 26657: 1, 26656: 1, 26655: 1, 26654: 1, 26653: 1, 2665
2: 1, 26651: 1, 26650: 1, 26649: 1, 26648: 1, 26647: 1, 26646: 1, 2664
5: 1, 26644: 1, 26643: 1, 26642: 1, 26641: 1, 26640: 1, 26639: 1, 2663
8: 1, 26637: 1, 26636: 1, 26635: 1, 26634: 1, 26633: 1, 26632: 1, 2663
1: 1, 26630: 1, 26629: 1, 26628: 1, 26627: 1, 26626: 1, 26625: 1, 2662
4: 1, 26623: 1, 26622: 1, 26621: 1, 26620: 1, 26619: 1, 26618: 1, 2661
7: 1, 26616: 1, 26615: 1, 26614: 1, 26613: 1, 26612: 1, 26611: 1, 2661
0: 1, 26609: 1, 26608: 1, 26607: 1, 26606: 1, 26605: 1, 26604: 1, 2660
3: 1, 26602: 1, 26601: 1, 26600: 1, 26599: 1, 26598: 1, 26597: 1, 2659
6: 1, 26595: 1, 26594: 1, 26593: 1, 26592: 1, 26591: 1, 26590: 1, 2658
9: 1, 26588: 1, 26587: 1, 26586: 1, 26585: 1, 26584: 1, 26583: 1, 2658
2: 1, 26581: 1, 26580: 1, 26579: 1, 26578: 1, 26577: 1, 26576: 1, 2657
5: 1, 26574: 1, 26573: 1, 26572: 1, 26571: 1, 26570: 1, 26569: 1, 2656
8: 1, 26567: 1, 26566: 1, 26565: 1, 26564: 1, 26563: 1, 26562: 1, 2656
1: 1, 26560: 1, 26559: 1, 26558: 1, 26557: 1, 26556: 1, 26555: 1, 2655
4: 1, 26553: 1, 26552: 1, 26551: 1, 26550: 1, 26549: 1, 26548: 1, 2654
7: 1, 26546: 1, 26545: 1, 26544: 1, 26543: 1, 26542: 1, 26541: 1, 2654
0: 1, 26539: 1, 26538: 1, 26537: 1, 26536: 1, 26535: 1, 26534: 1, 2653
3: 1, 26532: 1, 26531: 1, 26530: 1, 26529: 1, 26528: 1, 26527: 1, 2652
6: 1, 26525: 1, 26524: 1, 26523: 1, 26522: 1, 26521: 1, 26520: 1, 2651
9: 1, 26518: 1, 26517: 1, 26516: 1, 26515: 1, 26514: 1, 26513: 1, 2651
2: 1, 26511: 1, 26510: 1, 26509: 1, 26508: 1, 26507: 1, 26506: 1, 2650
5: 1, 26504: 1, 26503: 1, 26502: 1, 26501: 1, 26500: 1, 26499: 1, 2649
8: 1, 26497: 1, 26496: 1, 26495: 1, 26494: 1, 26493: 1, 26492: 1, 2649
1: 1, 26490: 1, 26489: 1, 26488: 1, 26487: 1, 26486: 1, 26485: 1, 2648
4: 1, 26483: 1, 26482: 1, 26481: 1, 26480: 1, 26479: 1, 26478: 1, 2647
7: 1, 26476: 1, 26475: 1, 26474: 1, 26473: 1, 26472: 1, 26471: 1, 2647
0: 1, 26469: 1, 26468: 1, 26467: 1, 26466: 1, 26465: 1, 26464: 1, 2646
3: 1, 26462: 1, 26461: 1, 26460: 1, 26459: 1, 26458: 1, 26457: 1, 2645
6: 1, 26455: 1, 26454: 1, 26453: 1, 26452: 1, 26451: 1, 26450: 1, 2644

9: 1, 26448: 1, 26447: 1, 26446: 1, 26445: 1, 26444: 1, 26443: 1, 2644
2: 1, 26441: 1, 26440: 1, 26439: 1, 26438: 1, 26437: 1, 26436: 1, 2643
5: 1, 26434: 1, 26433: 1, 26432: 1, 26431: 1, 26430: 1, 26429: 1, 2642
8: 1, 26427: 1, 26426: 1, 26425: 1, 26424: 1, 26423: 1, 26422: 1, 2642
1: 1, 26420: 1, 26419: 1, 26418: 1, 26417: 1, 26416: 1, 26415: 1, 2641
4: 1, 26413: 1, 26412: 1, 26411: 1, 26410: 1, 26409: 1, 26408: 1, 2640
7: 1, 26406: 1, 26405: 1, 26404: 1, 26403: 1, 26402: 1, 26401: 1, 2640
0: 1, 26399: 1, 26398: 1, 26397: 1, 26396: 1, 26395: 1, 26394: 1, 2639
3: 1, 26392: 1, 26391: 1, 26390: 1, 26389: 1, 26388: 1, 26387: 1, 2638
6: 1, 26385: 1, 26384: 1, 26383: 1, 26382: 1, 26381: 1, 26380: 1, 2637
9: 1, 26378: 1, 26377: 1, 26376: 1, 26375: 1, 26374: 1, 26373: 1, 2637
2: 1, 26371: 1, 26370: 1, 26369: 1, 26368: 1, 26367: 1, 26366: 1, 2636
5: 1, 26364: 1, 26363: 1, 26362: 1, 26361: 1, 26360: 1, 26359: 1, 2635
8: 1, 26357: 1, 26356: 1, 26355: 1, 26354: 1, 26353: 1, 26352: 1, 2635
1: 1, 26350: 1, 26349: 1, 26348: 1, 26347: 1, 26346: 1, 26345: 1, 2634
4: 1, 26343: 1, 26342: 1, 26341: 1, 26340: 1, 26339: 1, 26338: 1, 2633
7: 1, 26336: 1, 26335: 1, 26334: 1, 26333: 1, 26332: 1, 26331: 1, 2633
0: 1, 26329: 1, 26328: 1, 26327: 1, 26326: 1, 26325: 1, 26324: 1, 2632
3: 1, 26322: 1, 26321: 1, 26320: 1, 26319: 1, 26318: 1, 26317: 1, 2631
6: 1, 26315: 1, 26314: 1, 26313: 1, 26312: 1, 26311: 1, 26310: 1, 2630
9: 1, 26308: 1, 26307: 1, 26306: 1, 26305: 1, 26304: 1, 26303: 1, 2630
2: 1, 26301: 1, 26300: 1, 26299: 1, 26298: 1, 26297: 1, 26296: 1, 2629
5: 1, 26294: 1, 26293: 1, 26292: 1, 26291: 1, 26290: 1, 26289: 1, 2628
8: 1, 26287: 1, 26286: 1, 26285: 1, 26284: 1, 26283: 1, 26282: 1, 2628
1: 1, 26280: 1, 26279: 1, 26278: 1, 26277: 1, 26276: 1, 26275: 1, 2627
4: 1, 26273: 1, 26272: 1, 26271: 1, 26270: 1, 26269: 1, 26268: 1, 2626
7: 1, 26266: 1, 26265: 1, 26264: 1, 26263: 1, 26262: 1, 26261: 1, 2626
0: 1, 26259: 1, 26258: 1, 26257: 1, 26256: 1, 26255: 1, 26254: 1, 2625
3: 1, 26252: 1, 26251: 1, 26250: 1, 26249: 1, 26248: 1, 26247: 1, 2624
6: 1, 26245: 1, 26244: 1, 26243: 1, 26242: 1, 26241: 1, 26240: 1, 2623
9: 1, 26238: 1, 26237: 1, 26236: 1, 26235: 1, 26234: 1, 26233: 1, 2623
2: 1, 26231: 1, 26230: 1, 26229: 1, 26228: 1, 26227: 1, 26226: 1, 2622
5: 1, 26224: 1, 26223: 1, 26222: 1, 26221: 1, 26220: 1, 26219: 1, 2621
8: 1, 26217: 1, 26216: 1, 26215: 1, 26214: 1, 26213: 1, 26212: 1, 2621
1: 1, 26210: 1, 26209: 1, 26208: 1, 26207: 1, 26206: 1, 26205: 1, 2620
4: 1, 26203: 1, 26202: 1, 26201: 1, 26200: 1, 26199: 1, 26198: 1, 2619
7: 1, 26196: 1, 26195: 1, 26194: 1, 26193: 1, 26192: 1, 26191: 1, 2619
0: 1, 26189: 1, 26188: 1, 26187: 1, 26186: 1, 26185: 1, 26184: 1, 2618
3: 1, 26182: 1, 26181: 1, 26180: 1, 26179: 1, 26178: 1, 26177: 1, 2617

6: 1, 26175: 1, 26174: 1, 26173: 1, 26172: 1, 26171: 1, 26170: 1, 2616
9: 1, 26168: 1, 26167: 1, 26166: 1, 26165: 1, 26164: 1, 26163: 1, 2616
2: 1, 26161: 1, 26160: 1, 26159: 1, 26158: 1, 26157: 1, 26156: 1, 2615
5: 1, 26154: 1, 26153: 1, 26152: 1, 26151: 1, 26150: 1, 26149: 1, 2614
8: 1, 26147: 1, 26146: 1, 26145: 1, 26144: 1, 26143: 1, 26142: 1, 2614
1: 1, 26140: 1, 26139: 1, 26138: 1, 26137: 1, 26136: 1, 26135: 1, 2613
4: 1, 26133: 1, 26132: 1, 26131: 1, 26130: 1, 26129: 1, 26128: 1, 2612
7: 1, 26126: 1, 26125: 1, 26124: 1, 26123: 1, 26122: 1, 26121: 1, 2612
0: 1, 26119: 1, 26118: 1, 26117: 1, 26116: 1, 26115: 1, 26114: 1, 2611
3: 1, 26112: 1, 26111: 1, 26110: 1, 26109: 1, 26108: 1, 26107: 1, 2610
6: 1, 26105: 1, 26104: 1, 26103: 1, 26102: 1, 26101: 1, 26100: 1, 2609
9: 1, 26098: 1, 26097: 1, 26096: 1, 26095: 1, 26094: 1, 26093: 1, 2609
2: 1, 26091: 1, 26090: 1, 26089: 1, 26088: 1, 26087: 1, 26086: 1, 2608
5: 1, 26084: 1, 26083: 1, 26082: 1, 26081: 1, 26080: 1, 26079: 1, 2607
8: 1, 26077: 1, 26076: 1, 26075: 1, 26074: 1, 26073: 1, 26072: 1, 2607
1: 1, 26070: 1, 26069: 1, 26068: 1, 26067: 1, 26066: 1, 26065: 1, 2606
4: 1, 26063: 1, 26062: 1, 26061: 1, 26060: 1, 26059: 1, 26058: 1, 2605
7: 1, 26056: 1, 26055: 1, 26054: 1, 26053: 1, 26052: 1, 26051: 1, 2605
0: 1, 26049: 1, 26048: 1, 26047: 1, 26046: 1, 26045: 1, 26044: 1, 2604
3: 1, 26042: 1, 26041: 1, 26040: 1, 26039: 1, 26038: 1, 26037: 1, 2603
6: 1, 26035: 1, 26034: 1, 26033: 1, 26032: 1, 26031: 1, 26030: 1, 2602
9: 1, 26028: 1, 26027: 1, 26026: 1, 26025: 1, 26024: 1, 26023: 1, 2602
2: 1, 26021: 1, 26020: 1, 26019: 1, 26018: 1, 26017: 1, 26016: 1, 2601
5: 1, 26014: 1, 26013: 1, 26012: 1, 26011: 1, 26010: 1, 26009: 1, 2600
8: 1, 26007: 1, 26006: 1, 26005: 1, 26004: 1, 26003: 1, 26002: 1, 2600
1: 1, 26000: 1, 25999: 1, 25998: 1, 25997: 1, 25996: 1, 25995: 1, 2599
4: 1, 25993: 1, 25992: 1, 25991: 1, 25990: 1, 25989: 1, 25988: 1, 2598
7: 1, 25986: 1, 25985: 1, 25984: 1, 25983: 1, 25982: 1, 25981: 1, 2598
0: 1, 25979: 1, 25978: 1, 25977: 1, 25976: 1, 25975: 1, 25974: 1, 2597
3: 1, 25972: 1, 25971: 1, 25970: 1, 25969: 1, 25968: 1, 25967: 1, 2596
6: 1, 25965: 1, 25964: 1, 25963: 1, 25962: 1, 25961: 1, 25960: 1, 2595
9: 1, 25958: 1, 25957: 1, 25956: 1, 25955: 1, 25954: 1, 25953: 1, 2595
2: 1, 25951: 1, 25950: 1, 25949: 1, 25948: 1, 25947: 1, 25946: 1, 2594
5: 1, 25944: 1, 25943: 1, 25942: 1, 25941: 1, 25940: 1, 25939: 1, 2593
8: 1, 25937: 1, 25936: 1, 25935: 1, 25934: 1, 25933: 1, 25932: 1, 2593
1: 1, 25930: 1, 25929: 1, 25928: 1, 25927: 1, 25926: 1, 25925: 1, 2592
4: 1, 25923: 1, 25922: 1, 25921: 1, 25920: 1, 25919: 1, 25918: 1, 2591
7: 1, 25916: 1, 25915: 1, 25914: 1, 25913: 1, 25912: 1, 25911: 1, 2591
0: 1, 25909: 1, 25908: 1, 25907: 1, 25906: 1, 25905: 1, 25904: 1, 2590

3: 1, 25902: 1, 25901: 1, 25900: 1, 25899: 1, 25898: 1, 25897: 1, 2589
6: 1, 25895: 1, 25894: 1, 25893: 1, 25892: 1, 25891: 1, 25890: 1, 2588
9: 1, 25888: 1, 25887: 1, 25886: 1, 25885: 1, 25884: 1, 25883: 1, 2588
2: 1, 25881: 1, 25880: 1, 25879: 1, 25878: 1, 25877: 1, 25876: 1, 2587
5: 1, 25874: 1, 25873: 1, 25872: 1, 25871: 1, 25870: 1, 25869: 1, 2586
8: 1, 25867: 1, 25866: 1, 25865: 1, 25864: 1, 25863: 1, 25862: 1, 2586
1: 1, 25860: 1, 25859: 1, 25858: 1, 25857: 1, 25856: 1, 25855: 1, 2585
4: 1, 25853: 1, 25852: 1, 25851: 1, 25850: 1, 25849: 1, 25848: 1, 2584
7: 1, 25846: 1, 25845: 1, 25844: 1, 25843: 1, 25842: 1, 25841: 1, 2584
0: 1, 25839: 1, 25838: 1, 25837: 1, 25836: 1, 25835: 1, 25834: 1, 2583
3: 1, 25832: 1, 25831: 1, 25830: 1, 25829: 1, 25828: 1, 25827: 1, 2582
6: 1, 25825: 1, 25824: 1, 25823: 1, 25822: 1, 25821: 1, 25820: 1, 2581
9: 1, 25818: 1, 25817: 1, 25816: 1, 25815: 1, 25814: 1, 25813: 1, 2581
2: 1, 25811: 1, 25810: 1, 25809: 1, 25808: 1, 25807: 1, 25806: 1, 2580
5: 1, 25804: 1, 25803: 1, 25802: 1, 25801: 1, 25800: 1, 25799: 1, 2579
8: 1, 25797: 1, 25796: 1, 25795: 1, 25794: 1, 25793: 1, 25792: 1, 2579
1: 1, 25790: 1, 25789: 1, 25788: 1, 25787: 1, 25786: 1, 25785: 1, 2578
4: 1, 25783: 1, 25782: 1, 25781: 1, 25780: 1, 25779: 1, 25778: 1, 2577
7: 1, 25776: 1, 25775: 1, 25774: 1, 25773: 1, 25772: 1, 25771: 1, 2577
0: 1, 25769: 1, 25768: 1, 25767: 1, 25766: 1, 25765: 1, 25764: 1, 2576
3: 1, 25762: 1, 25761: 1, 25760: 1, 25759: 1, 25758: 1, 25757: 1, 2575
6: 1, 25755: 1, 25754: 1, 25753: 1, 25752: 1, 25751: 1, 25750: 1, 2574
9: 1, 25748: 1, 25747: 1, 25746: 1, 25745: 1, 25744: 1, 25743: 1, 2574
2: 1, 25741: 1, 25740: 1, 25739: 1, 25738: 1, 25737: 1, 25736: 1, 2573
5: 1, 25734: 1, 25733: 1, 25732: 1, 25731: 1, 25730: 1, 25729: 1, 2572
8: 1, 25727: 1, 25726: 1, 25725: 1, 25724: 1, 25723: 1, 25722: 1, 2572
1: 1, 25720: 1, 25719: 1, 25718: 1, 25717: 1, 25716: 1, 25715: 1, 2571
4: 1, 25713: 1, 25712: 1, 25711: 1, 25710: 1, 25709: 1, 25708: 1, 2570
7: 1, 25706: 1, 25705: 1, 25704: 1, 25703: 1, 25702: 1, 25701: 1, 2570
0: 1, 25699: 1, 25698: 1, 25697: 1, 25696: 1, 25695: 1, 25694: 1, 2569
3: 1, 25692: 1, 25691: 1, 25690: 1, 25689: 1, 25688: 1, 25687: 1, 2568
6: 1, 25685: 1, 25684: 1, 25683: 1, 25682: 1, 25681: 1, 25680: 1, 2567
9: 1, 25678: 1, 25677: 1, 25676: 1, 25675: 1, 25674: 1, 25673: 1, 2567
2: 1, 25671: 1, 25670: 1, 25669: 1, 25668: 1, 25667: 1, 25666: 1, 2566
5: 1, 25664: 1, 25663: 1, 25662: 1, 25661: 1, 25660: 1, 25659: 1, 2565
8: 1, 25657: 1, 25656: 1, 25655: 1, 25654: 1, 25653: 1, 25652: 1, 2565
1: 1, 25650: 1, 25649: 1, 25648: 1, 25647: 1, 25646: 1, 25645: 1, 2564
4: 1, 25643: 1, 25642: 1, 25641: 1, 25640: 1, 25639: 1, 25638: 1, 2563
7: 1, 25636: 1, 25635: 1, 25634: 1, 25633: 1, 25632: 1, 25631: 1, 2563

0: 1, 25629: 1, 25628: 1, 25627: 1, 25626: 1, 25625: 1, 25624: 1, 2562
3: 1, 25622: 1, 25621: 1, 25620: 1, 25619: 1, 25618: 1, 25617: 1, 2561
6: 1, 25615: 1, 25614: 1, 25613: 1, 25612: 1, 25611: 1, 25610: 1, 2560
9: 1, 25608: 1, 25607: 1, 25606: 1, 25605: 1, 25604: 1, 25603: 1, 2560
2: 1, 25601: 1, 25600: 1, 25599: 1, 25598: 1, 25597: 1, 25596: 1, 2559
5: 1, 25594: 1, 25593: 1, 25592: 1, 25591: 1, 25590: 1, 25589: 1, 2558
8: 1, 25587: 1, 25586: 1, 25585: 1, 25584: 1, 25583: 1, 25582: 1, 2558
1: 1, 25580: 1, 25579: 1, 25578: 1, 25577: 1, 25576: 1, 25575: 1, 2557
4: 1, 25573: 1, 25572: 1, 25571: 1, 25570: 1, 25569: 1, 25568: 1, 2556
7: 1, 25566: 1, 25565: 1, 25564: 1, 25563: 1, 25562: 1, 25561: 1, 2556
0: 1, 25559: 1, 25558: 1, 25557: 1, 25556: 1, 25555: 1, 25554: 1, 2555
3: 1, 25552: 1, 25551: 1, 25550: 1, 25549: 1, 25548: 1, 25547: 1, 2554
6: 1, 25545: 1, 25544: 1, 25543: 1, 25542: 1, 25541: 1, 25540: 1, 2553
9: 1, 25538: 1, 25537: 1, 25536: 1, 25535: 1, 25534: 1, 25533: 1, 2553
2: 1, 25531: 1, 25530: 1, 25529: 1, 25528: 1, 25527: 1, 25526: 1, 2552
5: 1, 25524: 1, 25523: 1, 25522: 1, 25521: 1, 25520: 1, 25519: 1, 2551
8: 1, 25517: 1, 25516: 1, 25515: 1, 25514: 1, 25513: 1, 25512: 1, 2551
1: 1, 25510: 1, 25509: 1, 25508: 1, 25507: 1, 25506: 1, 25505: 1, 2550
4: 1, 25503: 1, 25502: 1, 25501: 1, 25500: 1, 25499: 1, 25498: 1, 2549
7: 1, 25496: 1, 25495: 1, 25494: 1, 25493: 1, 25492: 1, 25491: 1, 2549
0: 1, 25489: 1, 25488: 1, 25487: 1, 25486: 1, 25485: 1, 25484: 1, 2548
3: 1, 25482: 1, 25481: 1, 25480: 1, 25479: 1, 25478: 1, 25477: 1, 2547
6: 1, 25475: 1, 25474: 1, 25473: 1, 25472: 1, 25471: 1, 25470: 1, 2546
9: 1, 25468: 1, 25467: 1, 25466: 1, 25465: 1, 25464: 1, 25463: 1, 2546
2: 1, 25461: 1, 25460: 1, 25459: 1, 25458: 1, 25457: 1, 25456: 1, 2545
5: 1, 25454: 1, 25453: 1, 25452: 1, 25451: 1, 25450: 1, 25449: 1, 2544
8: 1, 25447: 1, 25446: 1, 25445: 1, 25444: 1, 25443: 1, 25442: 1, 2544
1: 1, 25440: 1, 25439: 1, 25438: 1, 25437: 1, 25436: 1, 25435: 1, 2543
4: 1, 25433: 1, 25432: 1, 25431: 1, 25430: 1, 25429: 1, 25428: 1, 2542
7: 1, 25426: 1, 25425: 1, 25424: 1, 25423: 1, 25422: 1, 25421: 1, 2542
0: 1, 25419: 1, 25418: 1, 25417: 1, 25416: 1, 25415: 1, 25414: 1, 2541
3: 1, 25412: 1, 25411: 1, 25410: 1, 25409: 1, 25408: 1, 25407: 1, 2540
6: 1, 25405: 1, 25404: 1, 25403: 1, 25402: 1, 25401: 1, 25400: 1, 2539
9: 1, 25398: 1, 25397: 1, 25396: 1, 25395: 1, 25394: 1, 25393: 1, 2539
2: 1, 25391: 1, 25390: 1, 25389: 1, 25388: 1, 25387: 1, 25386: 1, 2538
5: 1, 25384: 1, 25383: 1, 25382: 1, 25381: 1, 25380: 1, 25379: 1, 2537
8: 1, 25377: 1, 25376: 1, 25375: 1, 25374: 1, 25373: 1, 25372: 1, 2537
1: 1, 25370: 1, 25369: 1, 25368: 1, 25367: 1, 25366: 1, 25365: 1, 2536
4: 1, 25363: 1, 25362: 1, 25361: 1, 25360: 1, 25359: 1, 25358: 1, 2535

7: 1, 25356: 1, 25355: 1, 25354: 1, 25353: 1, 25352: 1, 25351: 1, 2535
0: 1, 25349: 1, 25348: 1, 25347: 1, 25346: 1, 25345: 1, 25344: 1, 2534
3: 1, 25342: 1, 25341: 1, 25340: 1, 25339: 1, 25338: 1, 25337: 1, 2533
6: 1, 25335: 1, 25334: 1, 25333: 1, 25332: 1, 25331: 1, 25330: 1, 2532
9: 1, 25328: 1, 25327: 1, 25326: 1, 25325: 1, 25324: 1, 25323: 1, 2532
2: 1, 25321: 1, 25320: 1, 25319: 1, 25318: 1, 25317: 1, 25316: 1, 2531
5: 1, 25314: 1, 25313: 1, 25312: 1, 25311: 1, 25310: 1, 25309: 1, 2530
8: 1, 25307: 1, 25306: 1, 25305: 1, 25304: 1, 25303: 1, 25302: 1, 2530
1: 1, 25300: 1, 25299: 1, 25298: 1, 25297: 1, 25296: 1, 25295: 1, 2529
4: 1, 25293: 1, 25292: 1, 25291: 1, 25290: 1, 25289: 1, 25288: 1, 2528
7: 1, 25286: 1, 25285: 1, 25284: 1, 25283: 1, 25282: 1, 25281: 1, 2528
0: 1, 25279: 1, 25278: 1, 25277: 1, 25276: 1, 25275: 1, 25274: 1, 2527
3: 1, 25272: 1, 25271: 1, 25270: 1, 25269: 1, 25268: 1, 25267: 1, 2526
6: 1, 25265: 1, 25264: 1, 25263: 1, 25262: 1, 25261: 1, 25260: 1, 2525
9: 1, 25258: 1, 25257: 1, 25256: 1, 25255: 1, 25254: 1, 25253: 1, 2525
2: 1, 25251: 1, 25250: 1, 25249: 1, 25248: 1, 25247: 1, 25246: 1, 2524
5: 1, 25244: 1, 25243: 1, 25242: 1, 25241: 1, 25240: 1, 25239: 1, 2523
8: 1, 25237: 1, 25236: 1, 25235: 1, 25234: 1, 25233: 1, 25232: 1, 2523
1: 1, 25230: 1, 25229: 1, 25228: 1, 25227: 1, 25226: 1, 25225: 1, 2522
4: 1, 25223: 1, 25222: 1, 25221: 1, 25220: 1, 25219: 1, 25218: 1, 2521
7: 1, 25216: 1, 25215: 1, 25214: 1, 25213: 1, 25212: 1, 25211: 1, 2521
0: 1, 25209: 1, 25208: 1, 25207: 1, 25206: 1, 25205: 1, 25204: 1, 2520
3: 1, 25202: 1, 25201: 1, 25200: 1, 25199: 1, 25198: 1, 25197: 1, 2519
6: 1, 25195: 1, 25194: 1, 25193: 1, 25192: 1, 25191: 1, 25190: 1, 2518
9: 1, 25188: 1, 25187: 1, 25186: 1, 25185: 1, 25184: 1, 25183: 1, 2518
2: 1, 25181: 1, 25180: 1, 25179: 1, 25178: 1, 25177: 1, 25176: 1, 2517
5: 1, 25174: 1, 25173: 1, 25172: 1, 25171: 1, 25170: 1, 25169: 1, 2516
8: 1, 25167: 1, 25166: 1, 25165: 1, 25164: 1, 25163: 1, 25162: 1, 2516
1: 1, 25160: 1, 25159: 1, 25158: 1, 25157: 1, 25156: 1, 25155: 1, 2515
4: 1, 25153: 1, 25152: 1, 25151: 1, 25150: 1, 25149: 1, 25148: 1, 2514
7: 1, 25146: 1, 25145: 1, 25144: 1, 25143: 1, 25142: 1, 25141: 1, 2514
0: 1, 25139: 1, 25138: 1, 25137: 1, 25136: 1, 25135: 1, 25134: 1, 2513
3: 1, 25132: 1, 25131: 1, 25130: 1, 25129: 1, 25128: 1, 25127: 1, 2512
6: 1, 25125: 1, 25124: 1, 25123: 1, 25122: 1, 25121: 1, 25120: 1, 2511
9: 1, 25118: 1, 25117: 1, 25116: 1, 25115: 1, 25114: 1, 25113: 1, 2511
2: 1, 25111: 1, 25110: 1, 25109: 1, 25108: 1, 25107: 1, 25106: 1, 2510
5: 1, 25104: 1, 25103: 1, 25102: 1, 25101: 1, 25100: 1, 25099: 1, 2509
8: 1, 25097: 1, 25096: 1, 25095: 1, 25094: 1, 25093: 1, 25092: 1, 2509
1: 1, 25090: 1, 25089: 1, 25088: 1, 25087: 1, 25086: 1, 25085: 1, 2508

4: 1, 25083: 1, 25082: 1, 25081: 1, 25080: 1, 25079: 1, 25078: 1, 2507
7: 1, 25076: 1, 25075: 1, 25074: 1, 25073: 1, 25072: 1, 25071: 1, 2507
0: 1, 25069: 1, 25068: 1, 25067: 1, 25066: 1, 25065: 1, 25064: 1, 2506
3: 1, 25062: 1, 25061: 1, 25060: 1, 25059: 1, 25058: 1, 25057: 1, 2505
6: 1, 25055: 1, 25054: 1, 25053: 1, 25052: 1, 25051: 1, 25050: 1, 2504
9: 1, 25048: 1, 25047: 1, 25046: 1, 25045: 1, 25044: 1, 25043: 1, 2504
2: 1, 25041: 1, 25040: 1, 25039: 1, 25038: 1, 25037: 1, 25036: 1, 2503
5: 1, 25034: 1, 25033: 1, 25032: 1, 25031: 1, 25030: 1, 25029: 1, 2502
8: 1, 25027: 1, 25026: 1, 25025: 1, 25024: 1, 25023: 1, 25022: 1, 2502
1: 1, 25020: 1, 25019: 1, 25018: 1, 25017: 1, 25016: 1, 25015: 1, 2501
4: 1, 25013: 1, 25012: 1, 25011: 1, 25010: 1, 25009: 1, 25008: 1, 2500
7: 1, 25006: 1, 25005: 1, 25004: 1, 25003: 1, 25002: 1, 25001: 1, 2500
0: 1, 24999: 1, 24998: 1, 24997: 1, 24996: 1, 24995: 1, 24994: 1, 2499
3: 1, 24992: 1, 24991: 1, 24990: 1, 24989: 1, 24988: 1, 24987: 1, 2498
6: 1, 24985: 1, 24984: 1, 24983: 1, 24982: 1, 24981: 1, 24980: 1, 2497
9: 1, 24978: 1, 24977: 1, 24976: 1, 24975: 1, 24974: 1, 24973: 1, 2497
2: 1, 24971: 1, 24970: 1, 24969: 1, 24968: 1, 24967: 1, 24966: 1, 2496
5: 1, 24964: 1, 24963: 1, 24962: 1, 24961: 1, 24960: 1, 24959: 1, 2495
8: 1, 24957: 1, 24956: 1, 24955: 1, 24954: 1, 24953: 1, 24952: 1, 2495
1: 1, 24950: 1, 24949: 1, 24948: 1, 24947: 1, 24946: 1, 24945: 1, 2494
4: 1, 24943: 1, 24942: 1, 24941: 1, 24940: 1, 24939: 1, 24938: 1, 2493
7: 1, 24936: 1, 24935: 1, 24934: 1, 24933: 1, 24932: 1, 24931: 1, 2493
0: 1, 24929: 1, 24928: 1, 24927: 1, 24926: 1, 24925: 1, 24924: 1, 2492
3: 1, 24922: 1, 24921: 1, 24920: 1, 24919: 1, 24918: 1, 24917: 1, 2491
6: 1, 24915: 1, 24914: 1, 24913: 1, 24912: 1, 24911: 1, 24910: 1, 2490
9: 1, 24908: 1, 24907: 1, 24906: 1, 24905: 1, 24904: 1, 24903: 1, 2490
2: 1, 24901: 1, 24900: 1, 24899: 1, 24898: 1, 24897: 1, 24896: 1, 2489
5: 1, 24894: 1, 24893: 1, 24892: 1, 24891: 1, 24890: 1, 24889: 1, 2488
8: 1, 24887: 1, 24886: 1, 24885: 1, 24884: 1, 24883: 1, 24882: 1, 2488
1: 1, 24880: 1, 24879: 1, 24878: 1, 24877: 1, 24876: 1, 24875: 1, 2487
4: 1, 24873: 1, 24872: 1, 24871: 1, 24870: 1, 24869: 1, 24868: 1, 2486
7: 1, 24866: 1, 24865: 1, 24864: 1, 24863: 1, 24862: 1, 24861: 1, 2486
0: 1, 24859: 1, 24858: 1, 24857: 1, 24856: 1, 24855: 1, 24854: 1, 2485
3: 1, 24852: 1, 24851: 1, 24850: 1, 24849: 1, 24848: 1, 24847: 1, 2484
6: 1, 24845: 1, 24844: 1, 24843: 1, 24842: 1, 24841: 1, 24840: 1, 2483
9: 1, 24838: 1, 24837: 1, 24836: 1, 24835: 1, 24834: 1, 24833: 1, 2483
2: 1, 24831: 1, 24830: 1, 24829: 1, 24828: 1, 24827: 1, 24826: 1, 2482
5: 1, 24824: 1, 24823: 1, 24822: 1, 24821: 1, 24820: 1, 24819: 1, 2481
8: 1, 24817: 1, 24816: 1, 24815: 1, 24814: 1, 24813: 1, 24812: 1, 2481

1: 1, 24810: 1, 24809: 1, 24808: 1, 24807: 1, 24806: 1, 24805: 1, 2480
4: 1, 24803: 1, 24802: 1, 24801: 1, 24800: 1, 24799: 1, 24798: 1, 2479
7: 1, 24796: 1, 24795: 1, 24794: 1, 24793: 1, 24792: 1, 24791: 1, 2479
0: 1, 24789: 1, 24788: 1, 24787: 1, 24786: 1, 24785: 1, 24784: 1, 2478
3: 1, 24782: 1, 24781: 1, 24780: 1, 24779: 1, 24778: 1, 24777: 1, 2477
6: 1, 24775: 1, 24774: 1, 24773: 1, 24772: 1, 24771: 1, 24770: 1, 2476
9: 1, 24768: 1, 24767: 1, 24766: 1, 24765: 1, 24764: 1, 24763: 1, 2476
2: 1, 24761: 1, 24760: 1, 24759: 1, 24758: 1, 24757: 1, 24756: 1, 2475
5: 1, 24754: 1, 24753: 1, 24752: 1, 24751: 1, 24750: 1, 24749: 1, 2474
8: 1, 24747: 1, 24746: 1, 24745: 1, 24744: 1, 24743: 1, 24742: 1, 2474
1: 1, 24740: 1, 24739: 1, 24738: 1, 24737: 1, 24736: 1, 24735: 1, 2473
4: 1, 24733: 1, 24732: 1, 24731: 1, 24730: 1, 24729: 1, 24728: 1, 2472
7: 1, 24726: 1, 24725: 1, 24724: 1, 24723: 1, 24722: 1, 24721: 1, 2472
0: 1, 24719: 1, 24718: 1, 24717: 1, 24716: 1, 24715: 1, 24714: 1, 2471
3: 1, 24712: 1, 24711: 1, 24710: 1, 24709: 1, 24708: 1, 24707: 1, 2470
6: 1, 24705: 1, 24704: 1, 24703: 1, 24702: 1, 24701: 1, 24700: 1, 2469
9: 1, 24698: 1, 24697: 1, 24696: 1, 24695: 1, 24694: 1, 24693: 1, 2469
2: 1, 24691: 1, 24690: 1, 24689: 1, 24688: 1, 24687: 1, 24686: 1, 2468
5: 1, 24684: 1, 24683: 1, 24682: 1, 24681: 1, 24680: 1, 24679: 1, 2467
8: 1, 24677: 1, 24676: 1, 24675: 1, 24674: 1, 24673: 1, 24672: 1, 2467
1: 1, 24670: 1, 24669: 1, 24668: 1, 24667: 1, 24666: 1, 24665: 1, 2466
4: 1, 24663: 1, 24662: 1, 24661: 1, 24660: 1, 24659: 1, 24658: 1, 2465
7: 1, 24656: 1, 24655: 1, 24654: 1, 24653: 1, 24652: 1, 24651: 1, 2465
0: 1, 24649: 1, 24648: 1, 24647: 1, 24646: 1, 24645: 1, 24644: 1, 2464
3: 1, 24642: 1, 24641: 1, 24640: 1, 24639: 1, 24638: 1, 24637: 1, 2463
6: 1, 24635: 1, 24634: 1, 24633: 1, 24632: 1, 24631: 1, 24630: 1, 2462
9: 1, 24628: 1, 24627: 1, 24626: 1, 24625: 1, 24624: 1, 24623: 1, 2462
2: 1, 24621: 1, 24620: 1, 24619: 1, 24618: 1, 24617: 1, 24616: 1, 2461
5: 1, 24614: 1, 24613: 1, 24612: 1, 24611: 1, 24610: 1, 24609: 1, 2460
8: 1, 24607: 1, 24606: 1, 24605: 1, 24604: 1, 24603: 1, 24602: 1, 2460
1: 1, 24600: 1, 24599: 1, 24598: 1, 24597: 1, 24596: 1, 24595: 1, 2459
4: 1, 24593: 1, 24592: 1, 24591: 1, 24590: 1, 24589: 1, 24588: 1, 2458
7: 1, 24586: 1, 24585: 1, 24584: 1, 24583: 1, 24582: 1, 24581: 1, 2458
0: 1, 24579: 1, 24578: 1, 24577: 1, 24576: 1, 24575: 1, 24574: 1, 2457
3: 1, 24572: 1, 24571: 1, 24570: 1, 24569: 1, 24568: 1, 24567: 1, 2456
6: 1, 24565: 1, 24564: 1, 24563: 1, 24562: 1, 24561: 1, 24560: 1, 2455
9: 1, 24558: 1, 24557: 1, 24556: 1, 24555: 1, 24554: 1, 24553: 1, 2455
2: 1, 24551: 1, 24550: 1, 24549: 1, 24548: 1, 24547: 1, 24546: 1, 2454
5: 1, 24544: 1, 24543: 1, 24542: 1, 24541: 1, 24540: 1, 24539: 1, 2453

8: 1, 24537: 1, 24536: 1, 24535: 1, 24534: 1, 24533: 1, 24532: 1, 2453
1: 1, 24530: 1, 24529: 1, 24528: 1, 24527: 1, 24526: 1, 24525: 1, 2452
4: 1, 24523: 1, 24522: 1, 24521: 1, 24520: 1, 24519: 1, 24518: 1, 2451
7: 1, 24516: 1, 24515: 1, 24514: 1, 24513: 1, 24512: 1, 24511: 1, 2451
0: 1, 24509: 1, 24508: 1, 24507: 1, 24506: 1, 24505: 1, 24504: 1, 2450
3: 1, 24502: 1, 24501: 1, 24500: 1, 24499: 1, 24498: 1, 24497: 1, 2449
6: 1, 24495: 1, 24494: 1, 24493: 1, 24492: 1, 24491: 1, 24490: 1, 2448
9: 1, 24488: 1, 24487: 1, 24486: 1, 24485: 1, 24484: 1, 24483: 1, 2448
2: 1, 24481: 1, 24480: 1, 24479: 1, 24478: 1, 24477: 1, 24476: 1, 2447
5: 1, 24474: 1, 24473: 1, 24472: 1, 24471: 1, 24470: 1, 24469: 1, 2446
8: 1, 24467: 1, 24466: 1, 24465: 1, 24464: 1, 24463: 1, 24462: 1, 2446
1: 1, 24460: 1, 24459: 1, 24458: 1, 24457: 1, 24456: 1, 24455: 1, 2445
4: 1, 24453: 1, 24452: 1, 24451: 1, 24450: 1, 24449: 1, 24448: 1, 2444
7: 1, 24446: 1, 24445: 1, 24444: 1, 24443: 1, 24442: 1, 24441: 1, 2444
0: 1, 24439: 1, 24438: 1, 24437: 1, 24436: 1, 24435: 1, 24434: 1, 2443
3: 1, 24432: 1, 24431: 1, 24430: 1, 24429: 1, 24428: 1, 24427: 1, 2442
6: 1, 24425: 1, 24424: 1, 24423: 1, 24422: 1, 24421: 1, 24420: 1, 2441
9: 1, 24418: 1, 24417: 1, 24416: 1, 24415: 1, 24414: 1, 24413: 1, 2441
2: 1, 24411: 1, 24410: 1, 24409: 1, 24408: 1, 24407: 1, 24406: 1, 2440
5: 1, 24404: 1, 24403: 1, 24402: 1, 24401: 1, 24400: 1, 24399: 1, 2439
8: 1, 24397: 1, 24396: 1, 24395: 1, 24394: 1, 24393: 1, 24392: 1, 2439
1: 1, 24390: 1, 24389: 1, 24388: 1, 24387: 1, 24386: 1, 24385: 1, 2438
4: 1, 24383: 1, 24382: 1, 24381: 1, 24380: 1, 24379: 1, 24378: 1, 2437
7: 1, 24376: 1, 24375: 1, 24374: 1, 24373: 1, 24372: 1, 24371: 1, 2437
0: 1, 24369: 1, 24368: 1, 24367: 1, 24366: 1, 24365: 1, 24364: 1, 2436
3: 1, 24362: 1, 24361: 1, 24360: 1, 24359: 1, 24358: 1, 24357: 1, 2435
6: 1, 24355: 1, 24354: 1, 24353: 1, 24352: 1, 24351: 1, 24350: 1, 2434
9: 1, 24348: 1, 24347: 1, 24346: 1, 24345: 1, 24344: 1, 24343: 1, 2434
2: 1, 24341: 1, 24340: 1, 24339: 1, 24338: 1, 24337: 1, 24336: 1, 2433
5: 1, 24334: 1, 24333: 1, 24332: 1, 24331: 1, 24330: 1, 24329: 1, 2432
8: 1, 24327: 1, 24326: 1, 24325: 1, 24324: 1, 24323: 1, 24322: 1, 2432
1: 1, 24320: 1, 24319: 1, 24318: 1, 24317: 1, 24316: 1, 24315: 1, 2431
4: 1, 24313: 1, 24312: 1, 24311: 1, 24310: 1, 24309: 1, 24308: 1, 2430
7: 1, 24306: 1, 24305: 1, 24304: 1, 24303: 1, 24302: 1, 24301: 1, 2430
0: 1, 24299: 1, 24298: 1, 24297: 1, 24296: 1, 24295: 1, 24294: 1, 2429
3: 1, 24292: 1, 24291: 1, 24290: 1, 24289: 1, 24288: 1, 24287: 1, 2428
6: 1, 24285: 1, 24284: 1, 24283: 1, 24282: 1, 24281: 1, 24280: 1, 2427
9: 1, 24278: 1, 24277: 1, 24276: 1, 24275: 1, 24274: 1, 24273: 1, 2427
2: 1, 24271: 1, 24270: 1, 24269: 1, 24268: 1, 24267: 1, 24266: 1, 2426

5: 1, 24264: 1, 24263: 1, 24262: 1, 24261: 1, 24260: 1, 24259: 1, 2425
8: 1, 24257: 1, 24256: 1, 24255: 1, 24254: 1, 24253: 1, 24252: 1, 2425
1: 1, 24250: 1, 24249: 1, 24248: 1, 24247: 1, 24246: 1, 24245: 1, 2424
4: 1, 24243: 1, 24242: 1, 24241: 1, 24240: 1, 24239: 1, 24238: 1, 2423
7: 1, 24236: 1, 24235: 1, 24234: 1, 24233: 1, 24232: 1, 24231: 1, 2423
0: 1, 24229: 1, 24228: 1, 24227: 1, 24226: 1, 24225: 1, 24224: 1, 2422
3: 1, 24222: 1, 24221: 1, 24220: 1, 24219: 1, 24218: 1, 24217: 1, 2421
6: 1, 24215: 1, 24214: 1, 24213: 1, 24212: 1, 24211: 1, 24210: 1, 2420
9: 1, 24208: 1, 24207: 1, 24206: 1, 24205: 1, 24204: 1, 24203: 1, 2420
2: 1, 24201: 1, 24200: 1, 24199: 1, 24198: 1, 24197: 1, 24196: 1, 2419
5: 1, 24194: 1, 24193: 1, 24192: 1, 24191: 1, 24190: 1, 24189: 1, 2418
8: 1, 24187: 1, 24186: 1, 24185: 1, 24184: 1, 24183: 1, 24182: 1, 2418
1: 1, 24180: 1, 24179: 1, 24178: 1, 24177: 1, 24176: 1, 24175: 1, 2417
4: 1, 24173: 1, 24172: 1, 24171: 1, 24170: 1, 24169: 1, 24168: 1, 2416
7: 1, 24166: 1, 24165: 1, 24164: 1, 24163: 1, 24162: 1, 24161: 1, 2416
0: 1, 24159: 1, 24158: 1, 24157: 1, 24156: 1, 24155: 1, 24154: 1, 2415
3: 1, 24152: 1, 24151: 1, 24150: 1, 24149: 1, 24148: 1, 24147: 1, 2414
6: 1, 24145: 1, 24144: 1, 24143: 1, 24142: 1, 24141: 1, 24140: 1, 2413
9: 1, 24138: 1, 24137: 1, 24136: 1, 24135: 1, 24134: 1, 24133: 1, 2413
2: 1, 24131: 1, 24130: 1, 24129: 1, 24128: 1, 24127: 1, 24126: 1, 2412
5: 1, 24124: 1, 24123: 1, 24122: 1, 24121: 1, 24120: 1, 24119: 1, 2411
8: 1, 24117: 1, 24116: 1, 24115: 1, 24114: 1, 24113: 1, 24112: 1, 2411
1: 1, 24110: 1, 24109: 1, 24108: 1, 24107: 1, 24106: 1, 24105: 1, 2410
4: 1, 24103: 1, 24102: 1, 24101: 1, 24100: 1, 24099: 1, 24098: 1, 2409
7: 1, 24096: 1, 24095: 1, 24094: 1, 24093: 1, 24092: 1, 24091: 1, 2409
0: 1, 24089: 1, 24088: 1, 24087: 1, 24086: 1, 24085: 1, 24084: 1, 2408
3: 1, 24082: 1, 24081: 1, 24080: 1, 24079: 1, 24078: 1, 24077: 1, 2407
6: 1, 24075: 1, 24074: 1, 24073: 1, 24072: 1, 24071: 1, 24070: 1, 2406
9: 1, 24068: 1, 24067: 1, 24066: 1, 24065: 1, 24064: 1, 24063: 1, 2406
2: 1, 24061: 1, 24060: 1, 24059: 1, 24058: 1, 24057: 1, 24056: 1, 2405
5: 1, 24054: 1, 24053: 1, 24052: 1, 24051: 1, 24050: 1, 24049: 1, 2404
8: 1, 24047: 1, 24046: 1, 24045: 1, 24044: 1, 24043: 1, 24042: 1, 2404
1: 1, 24040: 1, 24039: 1, 24038: 1, 24037: 1, 24036: 1, 24035: 1, 2403
4: 1, 24033: 1, 24032: 1, 24031: 1, 24030: 1, 24029: 1, 24028: 1, 2402
7: 1, 24026: 1, 24025: 1, 24024: 1, 24023: 1, 24022: 1, 24021: 1, 2402
0: 1, 24019: 1, 24018: 1, 24017: 1, 24016: 1, 24015: 1, 24014: 1, 2401
3: 1, 24012: 1, 24011: 1, 24010: 1, 24009: 1, 24008: 1, 24007: 1, 2400
6: 1, 24005: 1, 24004: 1, 24003: 1, 24002: 1, 24001: 1, 24000: 1, 2399
9: 1, 23998: 1, 23997: 1, 23996: 1, 23995: 1, 23994: 1, 23993: 1, 2399

2: 1, 23991: 1, 23990: 1, 23989: 1, 23988: 1, 23987: 1, 23986: 1, 2398
5: 1, 23984: 1, 23983: 1, 23982: 1, 23981: 1, 23980: 1, 23979: 1, 2397
8: 1, 23977: 1, 23976: 1, 23975: 1, 23974: 1, 23973: 1, 23972: 1, 2397
1: 1, 23970: 1, 23969: 1, 23968: 1, 23967: 1, 23966: 1, 23965: 1, 2396
4: 1, 23963: 1, 23962: 1, 23961: 1, 23960: 1, 23959: 1, 23958: 1, 2395
7: 1, 23956: 1, 23955: 1, 23954: 1, 23953: 1, 23952: 1, 23951: 1, 2395
0: 1, 23949: 1, 23948: 1, 23947: 1, 23946: 1, 23945: 1, 23944: 1, 2394
3: 1, 23942: 1, 23941: 1, 23940: 1, 23939: 1, 23938: 1, 23937: 1, 2393
6: 1, 23935: 1, 23934: 1, 23933: 1, 23932: 1, 23931: 1, 23930: 1, 2392
9: 1, 23928: 1, 23927: 1, 23926: 1, 23925: 1, 23924: 1, 23923: 1, 2392
2: 1, 23921: 1, 23920: 1, 23919: 1, 23918: 1, 23917: 1, 23916: 1, 2391
5: 1, 23914: 1, 23913: 1, 23912: 1, 23911: 1, 23910: 1, 23909: 1, 2390
8: 1, 23907: 1, 23906: 1, 23905: 1, 23904: 1, 23903: 1, 23902: 1, 2390
1: 1, 23900: 1, 23899: 1, 23898: 1, 23897: 1, 23896: 1, 23895: 1, 2389
4: 1, 23893: 1, 23892: 1, 23891: 1, 23890: 1, 23889: 1, 23888: 1, 2388
7: 1, 23886: 1, 23885: 1, 23884: 1, 23883: 1, 23882: 1, 23881: 1, 2388
0: 1, 23879: 1, 23878: 1, 23877: 1, 23876: 1, 23875: 1, 23874: 1, 2387
3: 1, 23872: 1, 23871: 1, 23870: 1, 23869: 1, 23868: 1, 23867: 1, 2386
6: 1, 23865: 1, 23864: 1, 23863: 1, 23862: 1, 23861: 1, 23860: 1, 2385
9: 1, 23858: 1, 23857: 1, 23856: 1, 23855: 1, 23854: 1, 23853: 1, 2385
2: 1, 23851: 1, 23850: 1, 23849: 1, 23848: 1, 23847: 1, 23846: 1, 2384
5: 1, 23844: 1, 23843: 1, 23842: 1, 23841: 1, 23840: 1, 23839: 1, 2383
8: 1, 23837: 1, 23836: 1, 23835: 1, 23834: 1, 23833: 1, 23832: 1, 2383
1: 1, 23830: 1, 23829: 1, 23828: 1, 23827: 1, 23826: 1, 23825: 1, 2382
4: 1, 23823: 1, 23822: 1, 23821: 1, 23820: 1, 23819: 1, 23818: 1, 2381
7: 1, 23816: 1, 23815: 1, 23814: 1, 23813: 1, 23812: 1, 23811: 1, 2381
0: 1, 23809: 1, 23808: 1, 23807: 1, 23806: 1, 23805: 1, 23804: 1, 2380
3: 1, 23802: 1, 23801: 1, 23800: 1, 23799: 1, 23798: 1, 23797: 1, 2379
6: 1, 23795: 1, 23794: 1, 23793: 1, 23792: 1, 23791: 1, 23790: 1, 2378
9: 1, 23788: 1, 23787: 1, 23786: 1, 23785: 1, 23784: 1, 23783: 1, 2378
2: 1, 23781: 1, 23780: 1, 23779: 1, 23778: 1, 23777: 1, 23776: 1, 2377
5: 1, 23774: 1, 23773: 1, 23772: 1, 23771: 1, 23770: 1, 23769: 1, 2376
8: 1, 23767: 1, 23766: 1, 23765: 1, 23764: 1, 23763: 1, 23762: 1, 2376
1: 1, 23760: 1, 23759: 1, 23758: 1, 23757: 1, 23756: 1, 23755: 1, 2375
4: 1, 23753: 1, 23752: 1, 23751: 1, 23750: 1, 23749: 1, 23748: 1, 2374
7: 1, 23746: 1, 23745: 1, 23744: 1, 23743: 1, 23742: 1, 23741: 1, 2374
0: 1, 23739: 1, 23738: 1, 23737: 1, 23736: 1, 23735: 1, 23734: 1, 2373
3: 1, 23732: 1, 23731: 1, 23730: 1, 23729: 1, 23728: 1, 23727: 1, 2372
6: 1, 23725: 1, 23724: 1, 23723: 1, 23722: 1, 23721: 1, 23720: 1, 2371

9: 1, 23718: 1, 23717: 1, 23716: 1, 23715: 1, 23714: 1, 23713: 1, 2371
2: 1, 23711: 1, 23710: 1, 23709: 1, 23708: 1, 23707: 1, 23706: 1, 2370
5: 1, 23704: 1, 23703: 1, 23702: 1, 23701: 1, 23700: 1, 23699: 1, 2369
8: 1, 23697: 1, 23696: 1, 23695: 1, 23694: 1, 23693: 1, 23692: 1, 2369
1: 1, 23690: 1, 23689: 1, 23688: 1, 23687: 1, 23686: 1, 23685: 1, 2368
4: 1, 23683: 1, 23682: 1, 23681: 1, 23680: 1, 23679: 1, 23678: 1, 2367
7: 1, 23676: 1, 23675: 1, 23674: 1, 23673: 1, 23672: 1, 23671: 1, 2367
0: 1, 23669: 1, 23668: 1, 23667: 1, 23666: 1, 23665: 1, 23664: 1, 2366
3: 1, 23662: 1, 23661: 1, 23660: 1, 23659: 1, 23658: 1, 23657: 1, 2365
6: 1, 23655: 1, 23654: 1, 23653: 1, 23652: 1, 23651: 1, 23650: 1, 2364
9: 1, 23648: 1, 23647: 1, 23646: 1, 23645: 1, 23644: 1, 23643: 1, 2364
2: 1, 23641: 1, 23640: 1, 23639: 1, 23638: 1, 23637: 1, 23636: 1, 2363
5: 1, 23634: 1, 23633: 1, 23632: 1, 23631: 1, 23630: 1, 23629: 1, 2362
8: 1, 23627: 1, 23626: 1, 23625: 1, 23624: 1, 23623: 1, 23622: 1, 2362
1: 1, 23620: 1, 23619: 1, 23618: 1, 23617: 1, 23616: 1, 23615: 1, 2361
4: 1, 23613: 1, 23612: 1, 23611: 1, 23610: 1, 23609: 1, 23608: 1, 2360
7: 1, 23606: 1, 23605: 1, 23604: 1, 23603: 1, 23602: 1, 23601: 1, 2360
0: 1, 23599: 1, 23598: 1, 23597: 1, 23596: 1, 23595: 1, 23594: 1, 2359
3: 1, 23592: 1, 23591: 1, 23590: 1, 23589: 1, 23588: 1, 23587: 1, 2358
6: 1, 23585: 1, 23584: 1, 23583: 1, 23582: 1, 23581: 1, 23580: 1, 2357
9: 1, 23578: 1, 23577: 1, 23576: 1, 23575: 1, 23574: 1, 23573: 1, 2357
2: 1, 23571: 1, 23570: 1, 23569: 1, 23568: 1, 23567: 1, 23566: 1, 2356
5: 1, 23564: 1, 23563: 1, 23562: 1, 23561: 1, 23560: 1, 23559: 1, 2355
8: 1, 23557: 1, 23556: 1, 23555: 1, 23554: 1, 23553: 1, 23552: 1, 2355
1: 1, 23550: 1, 23549: 1, 23548: 1, 23547: 1, 23546: 1, 23545: 1, 2354
4: 1, 23543: 1, 23542: 1, 23541: 1, 23540: 1, 23539: 1, 23538: 1, 2353
7: 1, 23536: 1, 23535: 1, 23534: 1, 23533: 1, 23532: 1, 23531: 1, 2353
0: 1, 23529: 1, 23528: 1, 23527: 1, 23526: 1, 23525: 1, 23524: 1, 2352
3: 1, 23522: 1, 23521: 1, 23520: 1, 23519: 1, 23518: 1, 23517: 1, 2351
6: 1, 23515: 1, 23514: 1, 23513: 1, 23512: 1, 23511: 1, 23510: 1, 2350
9: 1, 23508: 1, 23507: 1, 23506: 1, 23505: 1, 23504: 1, 23503: 1, 2350
2: 1, 23501: 1, 23500: 1, 23499: 1, 23498: 1, 23497: 1, 23496: 1, 2349
5: 1, 23494: 1, 23493: 1, 23492: 1, 23491: 1, 23490: 1, 23489: 1, 2348
8: 1, 23487: 1, 23486: 1, 23485: 1, 23484: 1, 23483: 1, 23482: 1, 2348
1: 1, 23480: 1, 23479: 1, 23478: 1, 23477: 1, 23476: 1, 23475: 1, 2347
4: 1, 23473: 1, 23472: 1, 23471: 1, 23470: 1, 23469: 1, 23468: 1, 2346
7: 1, 23466: 1, 23465: 1, 23464: 1, 23463: 1, 23462: 1, 23461: 1, 2346
0: 1, 23459: 1, 23458: 1, 23457: 1, 23456: 1, 23455: 1, 23454: 1, 2345
3: 1, 23452: 1, 23451: 1, 23450: 1, 23449: 1, 23448: 1, 23447: 1, 2344

6: 1, 23445: 1, 23444: 1, 23443: 1, 23442: 1, 23441: 1, 23440: 1, 2343
9: 1, 23438: 1, 23437: 1, 23436: 1, 23435: 1, 23434: 1, 23433: 1, 2343
2: 1, 23431: 1, 23430: 1, 23429: 1, 23428: 1, 23427: 1, 23426: 1, 2342
5: 1, 23424: 1, 23423: 1, 23422: 1, 23421: 1, 23420: 1, 23419: 1, 2341
8: 1, 23417: 1, 23416: 1, 23415: 1, 23414: 1, 23413: 1, 23412: 1, 2341
1: 1, 23410: 1, 23409: 1, 23408: 1, 23407: 1, 23406: 1, 23405: 1, 2340
4: 1, 23403: 1, 23402: 1, 23401: 1, 23400: 1, 23399: 1, 23398: 1, 2339
7: 1, 23396: 1, 23395: 1, 23394: 1, 23393: 1, 23392: 1, 23391: 1, 2339
0: 1, 23389: 1, 23388: 1, 23387: 1, 23386: 1, 23385: 1, 23384: 1, 2338
3: 1, 23382: 1, 23381: 1, 23380: 1, 23379: 1, 23378: 1, 23377: 1, 2337
6: 1, 23375: 1, 23374: 1, 23373: 1, 23372: 1, 23371: 1, 23370: 1, 2336
9: 1, 23368: 1, 23367: 1, 23366: 1, 23365: 1, 23364: 1, 23363: 1, 2336
2: 1, 23361: 1, 23360: 1, 23359: 1, 23358: 1, 23357: 1, 23356: 1, 2335
5: 1, 23354: 1, 23353: 1, 23352: 1, 23351: 1, 23350: 1, 23349: 1, 2334
8: 1, 23347: 1, 23346: 1, 23345: 1, 23344: 1, 23343: 1, 23342: 1, 2334
1: 1, 23340: 1, 23339: 1, 23338: 1, 23337: 1, 23336: 1, 23335: 1, 2333
4: 1, 23333: 1, 23332: 1, 23331: 1, 23330: 1, 23329: 1, 23328: 1, 2332
7: 1, 23326: 1, 23325: 1, 23324: 1, 23323: 1, 23322: 1, 23321: 1, 2332
0: 1, 23319: 1, 23318: 1, 23317: 1, 23316: 1, 23315: 1, 23314: 1, 2331
3: 1, 23312: 1, 23311: 1, 23310: 1, 23309: 1, 23308: 1, 23307: 1, 2330
6: 1, 23305: 1, 23304: 1, 23303: 1, 23302: 1, 23301: 1, 23300: 1, 2329
9: 1, 23298: 1, 23297: 1, 23296: 1, 23295: 1, 23294: 1, 23293: 1, 2329
2: 1, 23291: 1, 23290: 1, 23289: 1, 23288: 1, 23287: 1, 23286: 1, 2328
5: 1, 23284: 1, 23283: 1, 23282: 1, 23281: 1, 23280: 1, 23279: 1, 2327
8: 1, 23277: 1, 23276: 1, 23275: 1, 23274: 1, 23273: 1, 23272: 1, 2327
1: 1, 23270: 1, 23269: 1, 23268: 1, 23267: 1, 23266: 1, 23265: 1, 2326
4: 1, 23263: 1, 23262: 1, 23261: 1, 23260: 1, 23259: 1, 23258: 1, 2325
7: 1, 23256: 1, 23255: 1, 23254: 1, 23253: 1, 23252: 1, 23251: 1, 2325
0: 1, 23249: 1, 23248: 1, 23247: 1, 23246: 1, 23245: 1, 23244: 1, 2324
3: 1, 23242: 1, 23241: 1, 23240: 1, 23239: 1, 23238: 1, 23237: 1, 2323
6: 1, 23235: 1, 23234: 1, 23233: 1, 23232: 1, 23231: 1, 23230: 1, 2322
9: 1, 23228: 1, 23227: 1, 23226: 1, 23225: 1, 23224: 1, 23223: 1, 2322
2: 1, 23221: 1, 23220: 1, 23219: 1, 23218: 1, 23217: 1, 23216: 1, 2321
5: 1, 23214: 1, 23213: 1, 23212: 1, 23211: 1, 23210: 1, 23209: 1, 2320
8: 1, 23207: 1, 23206: 1, 23205: 1, 23204: 1, 23203: 1, 23202: 1, 2320
1: 1, 23200: 1, 23199: 1, 23198: 1, 23197: 1, 23196: 1, 23195: 1, 2319
4: 1, 23193: 1, 23192: 1, 23191: 1, 23190: 1, 23189: 1, 23188: 1, 2318
7: 1, 23186: 1, 23185: 1, 23184: 1, 23183: 1, 23182: 1, 23181: 1, 2318
0: 1, 23179: 1, 23178: 1, 23177: 1, 23176: 1, 23175: 1, 23174: 1, 2317

3: 1, 23172: 1, 23171: 1, 23170: 1, 23169: 1, 23168: 1, 23167: 1, 2316
6: 1, 23165: 1, 23164: 1, 23163: 1, 23162: 1, 23161: 1, 23160: 1, 2315
9: 1, 23158: 1, 23157: 1, 23156: 1, 23155: 1, 23154: 1, 23153: 1, 2315
2: 1, 23151: 1, 23150: 1, 23149: 1, 23148: 1, 23147: 1, 23146: 1, 2314
5: 1, 23144: 1, 23143: 1, 23142: 1, 23141: 1, 23140: 1, 23139: 1, 2313
8: 1, 23137: 1, 23136: 1, 23135: 1, 23134: 1, 23133: 1, 23132: 1, 2313
1: 1, 23130: 1, 23129: 1, 23128: 1, 23127: 1, 23126: 1, 23125: 1, 2312
4: 1, 23123: 1, 23122: 1, 23121: 1, 23120: 1, 23119: 1, 23118: 1, 2311
7: 1, 23116: 1, 23115: 1, 23114: 1, 23113: 1, 23112: 1, 23111: 1, 2311
0: 1, 23109: 1, 23108: 1, 23107: 1, 23106: 1, 23105: 1, 23104: 1, 2310
3: 1, 23102: 1, 23101: 1, 23100: 1, 23099: 1, 23098: 1, 23097: 1, 2309
6: 1, 23095: 1, 23094: 1, 23093: 1, 23092: 1, 23091: 1, 23090: 1, 2308
9: 1, 23088: 1, 23087: 1, 23086: 1, 23085: 1, 23084: 1, 23083: 1, 2308
2: 1, 23081: 1, 23080: 1, 23079: 1, 23078: 1, 23077: 1, 23076: 1, 2307
5: 1, 23074: 1, 23073: 1, 23072: 1, 23071: 1, 23070: 1, 23069: 1, 2306
8: 1, 23067: 1, 23066: 1, 23065: 1, 23064: 1, 23063: 1, 23062: 1, 2306
1: 1, 23060: 1, 23059: 1, 23058: 1, 23057: 1, 23056: 1, 23055: 1, 2305
4: 1, 23053: 1, 23052: 1, 23051: 1, 23050: 1, 23049: 1, 23048: 1, 2304
7: 1, 23046: 1, 23045: 1, 23044: 1, 23043: 1, 23042: 1, 23041: 1, 2304
0: 1, 23039: 1, 23038: 1, 23037: 1, 23036: 1, 23035: 1, 23034: 1, 2303
3: 1, 23032: 1, 23031: 1, 23030: 1, 23029: 1, 23028: 1, 23027: 1, 2302
6: 1, 23025: 1, 23024: 1, 23023: 1, 23022: 1, 23021: 1, 23020: 1, 2301
9: 1, 23018: 1, 23017: 1, 23016: 1, 23015: 1, 23014: 1, 23013: 1, 2301
2: 1, 23011: 1, 23010: 1, 23009: 1, 23008: 1, 23007: 1, 23006: 1, 2300
5: 1, 23004: 1, 23003: 1, 23002: 1, 23001: 1, 23000: 1, 22999: 1, 2299
8: 1, 22997: 1, 22996: 1, 22995: 1, 22994: 1, 22993: 1, 22992: 1, 2299
1: 1, 22990: 1, 22989: 1, 22988: 1, 22987: 1, 22986: 1, 22985: 1, 2298
4: 1, 22983: 1, 22982: 1, 22981: 1, 22980: 1, 22979: 1, 22978: 1, 2297
7: 1, 22976: 1, 22975: 1, 22974: 1, 22973: 1, 22972: 1, 22971: 1, 2297
0: 1, 22969: 1, 22968: 1, 22967: 1, 22966: 1, 22965: 1, 22964: 1, 2296
3: 1, 22962: 1, 22961: 1, 22960: 1, 22959: 1, 22958: 1, 22957: 1, 2295
6: 1, 22955: 1, 22954: 1, 22953: 1, 22952: 1, 22951: 1, 22950: 1, 2294
9: 1, 22948: 1, 22947: 1, 22946: 1, 22945: 1, 22944: 1, 22943: 1, 2294
2: 1, 22941: 1, 22940: 1, 22939: 1, 22938: 1, 22937: 1, 22936: 1, 2293
5: 1, 22934: 1, 22933: 1, 22932: 1, 22931: 1, 22930: 1, 22929: 1, 2292
8: 1, 22927: 1, 22926: 1, 22925: 1, 22924: 1, 22923: 1, 22922: 1, 2292
1: 1, 22920: 1, 22919: 1, 22918: 1, 22917: 1, 22916: 1, 22915: 1, 2291
4: 1, 22913: 1, 22912: 1, 22911: 1, 22910: 1, 22909: 1, 22908: 1, 2290
7: 1, 22906: 1, 22905: 1, 22904: 1, 22903: 1, 22902: 1, 22901: 1, 2290

0: 1, 22899: 1, 22898: 1, 22897: 1, 22896: 1, 22895: 1, 22894: 1, 2289
3: 1, 22892: 1, 22891: 1, 22890: 1, 22889: 1, 22888: 1, 22887: 1, 2288
6: 1, 22885: 1, 22884: 1, 22883: 1, 22882: 1, 22881: 1, 22880: 1, 2287
9: 1, 22878: 1, 22877: 1, 22876: 1, 22875: 1, 22874: 1, 22873: 1, 2287
2: 1, 22871: 1, 22870: 1, 22869: 1, 22868: 1, 22867: 1, 22866: 1, 2286
5: 1, 22864: 1, 22863: 1, 22862: 1, 22861: 1, 22860: 1, 22859: 1, 2285
8: 1, 22857: 1, 22856: 1, 22855: 1, 22854: 1, 22853: 1, 22852: 1, 2285
1: 1, 22850: 1, 22849: 1, 22848: 1, 22847: 1, 22846: 1, 22845: 1, 2284
4: 1, 22843: 1, 22842: 1, 22841: 1, 22840: 1, 22839: 1, 22838: 1, 2283
7: 1, 22836: 1, 22835: 1, 22834: 1, 22833: 1, 22832: 1, 22831: 1, 2283
0: 1, 22829: 1, 22828: 1, 22827: 1, 22826: 1, 22825: 1, 22824: 1, 2282
3: 1, 22822: 1, 22821: 1, 22820: 1, 22819: 1, 22818: 1, 22817: 1, 2281
6: 1, 22815: 1, 22814: 1, 22813: 1, 22812: 1, 22811: 1, 22810: 1, 2280
9: 1, 22808: 1, 22807: 1, 22806: 1, 22805: 1, 22804: 1, 22803: 1, 2280
2: 1, 22801: 1, 22800: 1, 22799: 1, 22798: 1, 22797: 1, 22796: 1, 2279
5: 1, 22794: 1, 22793: 1, 22792: 1, 22791: 1, 22790: 1, 22789: 1, 2278
8: 1, 22787: 1, 22786: 1, 22785: 1, 22784: 1, 22783: 1, 22782: 1, 2278
1: 1, 22780: 1, 22779: 1, 22778: 1, 22777: 1, 22776: 1, 22775: 1, 2277
4: 1, 22773: 1, 22772: 1, 22771: 1, 22770: 1, 22769: 1, 22768: 1, 2276
7: 1, 22766: 1, 22765: 1, 22764: 1, 22763: 1, 22762: 1, 22761: 1, 2276
0: 1, 22759: 1, 22758: 1, 22757: 1, 22756: 1, 22755: 1, 22754: 1, 2275
3: 1, 22752: 1, 22751: 1, 22750: 1, 22749: 1, 22748: 1, 22747: 1, 2274
6: 1, 22745: 1, 22744: 1, 22743: 1, 22742: 1, 22741: 1, 22740: 1, 2273
9: 1, 22738: 1, 22737: 1, 22736: 1, 22735: 1, 22734: 1, 22733: 1, 2273
2: 1, 22731: 1, 22730: 1, 22729: 1, 22728: 1, 22727: 1, 22726: 1, 2272
5: 1, 22724: 1, 22723: 1, 22722: 1, 22721: 1, 22720: 1, 22719: 1, 2271
8: 1, 22717: 1, 22716: 1, 22715: 1, 22714: 1, 22713: 1, 22712: 1, 2271
1: 1, 22710: 1, 22709: 1, 22708: 1, 22707: 1, 22706: 1, 22705: 1, 2270
4: 1, 22703: 1, 22702: 1, 22701: 1, 22700: 1, 22699: 1, 22698: 1, 2269
7: 1, 22696: 1, 22695: 1, 22694: 1, 22693: 1, 22692: 1, 22691: 1, 2269
0: 1, 22689: 1, 22688: 1, 22687: 1, 22686: 1, 22685: 1, 22684: 1, 2268
3: 1, 22682: 1, 22681: 1, 22680: 1, 22679: 1, 22678: 1, 22677: 1, 2267
6: 1, 22675: 1, 22674: 1, 22673: 1, 22672: 1, 22671: 1, 22670: 1, 2266
9: 1, 22668: 1, 22667: 1, 22666: 1, 22665: 1, 22664: 1, 22663: 1, 2266
2: 1, 22661: 1, 22660: 1, 22659: 1, 22658: 1, 22657: 1, 22656: 1, 2265
5: 1, 22654: 1, 22653: 1, 22652: 1, 22651: 1, 22650: 1, 22649: 1, 2264
8: 1, 22647: 1, 22646: 1, 22645: 1, 22644: 1, 22643: 1, 22642: 1, 2264
1: 1, 22640: 1, 22639: 1, 22638: 1, 22637: 1, 22636: 1, 22635: 1, 2263
4: 1, 22633: 1, 22632: 1, 22631: 1, 22630: 1, 22629: 1, 22628: 1, 2262

7: 1, 22626: 1, 22625: 1, 22624: 1, 22623: 1, 22622: 1, 22621: 1, 2262
0: 1, 22619: 1, 22618: 1, 22617: 1, 22616: 1, 22615: 1, 22614: 1, 2261
3: 1, 22612: 1, 22611: 1, 22610: 1, 22609: 1, 22608: 1, 22607: 1, 2260
6: 1, 22605: 1, 22604: 1, 22603: 1, 22602: 1, 22601: 1, 22600: 1, 2259
9: 1, 22598: 1, 22597: 1, 22596: 1, 22595: 1, 22594: 1, 22593: 1, 2259
2: 1, 22591: 1, 22590: 1, 22589: 1, 22588: 1, 22587: 1, 22586: 1, 2258
5: 1, 22584: 1, 22583: 1, 22582: 1, 22581: 1, 22580: 1, 22579: 1, 2257
8: 1, 22577: 1, 22576: 1, 22575: 1, 22574: 1, 22573: 1, 22572: 1, 2257
1: 1, 22570: 1, 22569: 1, 22568: 1, 22567: 1, 22566: 1, 22565: 1, 2256
4: 1, 22563: 1, 22562: 1, 22561: 1, 22560: 1, 22559: 1, 22558: 1, 2255
7: 1, 22556: 1, 22555: 1, 22554: 1, 22553: 1, 22552: 1, 22551: 1, 2255
0: 1, 22549: 1, 22548: 1, 22547: 1, 22546: 1, 22545: 1, 22544: 1, 2254
3: 1, 22542: 1, 22541: 1, 22540: 1, 22539: 1, 22538: 1, 22537: 1, 2253
6: 1, 22535: 1, 22534: 1, 22533: 1, 22532: 1, 22531: 1, 22530: 1, 2252
9: 1, 22528: 1, 22527: 1, 22526: 1, 22525: 1, 22524: 1, 22523: 1, 2252
2: 1, 22521: 1, 22520: 1, 22519: 1, 22518: 1, 22517: 1, 22516: 1, 2251
5: 1, 22514: 1, 22513: 1, 22512: 1, 22511: 1, 22510: 1, 22509: 1, 2250
8: 1, 22507: 1, 22506: 1, 22505: 1, 22504: 1, 22503: 1, 22502: 1, 2250
1: 1, 22500: 1, 22499: 1, 22498: 1, 22497: 1, 22496: 1, 22495: 1, 2249
4: 1, 22493: 1, 22492: 1, 22491: 1, 22490: 1, 22489: 1, 22488: 1, 2248
7: 1, 22486: 1, 22485: 1, 22484: 1, 22483: 1, 22482: 1, 22481: 1, 2248
0: 1, 22479: 1, 22478: 1, 22477: 1, 22476: 1, 22475: 1, 22474: 1, 2247
3: 1, 22472: 1, 22471: 1, 22470: 1, 22469: 1, 22468: 1, 22467: 1, 2246
6: 1, 22465: 1, 22464: 1, 22463: 1, 22462: 1, 22461: 1, 22460: 1, 2245
9: 1, 22458: 1, 22457: 1, 22456: 1, 22455: 1, 22454: 1, 22453: 1, 2245
2: 1, 22451: 1, 22450: 1, 22449: 1, 22448: 1, 22447: 1, 22446: 1, 2244
5: 1, 22444: 1, 22443: 1, 22442: 1, 22441: 1, 22440: 1, 22439: 1, 2243
8: 1, 22437: 1, 22436: 1, 22435: 1, 22434: 1, 22433: 1, 22432: 1, 2243
1: 1, 22430: 1, 22429: 1, 22428: 1, 22427: 1, 22426: 1, 22425: 1, 2242
4: 1, 22423: 1, 22422: 1, 22421: 1, 22420: 1, 22419: 1, 22418: 1, 2241
7: 1, 22416: 1, 22415: 1, 22414: 1, 22413: 1, 22412: 1, 22411: 1, 2241
0: 1, 22409: 1, 22408: 1, 22407: 1, 22406: 1, 22405: 1, 22404: 1, 2240
3: 1, 22402: 1, 22401: 1, 22400: 1, 22399: 1, 22398: 1, 22397: 1, 2239
6: 1, 22395: 1, 22394: 1, 22393: 1, 22392: 1, 22391: 1, 22390: 1, 2238
9: 1, 22388: 1, 22387: 1, 22386: 1, 22385: 1, 22384: 1, 22383: 1, 2238
2: 1, 22381: 1, 22380: 1, 22379: 1, 22378: 1, 22377: 1, 22376: 1, 2237
5: 1, 22374: 1, 22373: 1, 22372: 1, 22371: 1, 22370: 1, 22369: 1, 2236
8: 1, 22367: 1, 22366: 1, 22365: 1, 22364: 1, 22363: 1, 22362: 1, 2236
1: 1, 22360: 1, 22359: 1, 22358: 1, 22357: 1, 22356: 1, 22355: 1, 2235

4: 1, 22353: 1, 22352: 1, 22351: 1, 22350: 1, 22349: 1, 22348: 1, 2234
7: 1, 22346: 1, 22345: 1, 22344: 1, 22343: 1, 22342: 1, 22341: 1, 2234
0: 1, 22339: 1, 22338: 1, 22337: 1, 22336: 1, 22335: 1, 22334: 1, 2233
3: 1, 22332: 1, 22331: 1, 22330: 1, 22329: 1, 22328: 1, 22327: 1, 2232
6: 1, 22325: 1, 22324: 1, 22323: 1, 22322: 1, 22321: 1, 22320: 1, 2231
9: 1, 22318: 1, 22317: 1, 22316: 1, 22315: 1, 22314: 1, 22313: 1, 2231
2: 1, 22311: 1, 22310: 1, 22309: 1, 22308: 1, 22307: 1, 22306: 1, 2230
5: 1, 22304: 1, 22303: 1, 22302: 1, 22301: 1, 22300: 1, 22299: 1, 2229
8: 1, 22297: 1, 22296: 1, 22295: 1, 22294: 1, 22293: 1, 22292: 1, 2229
1: 1, 22290: 1, 22289: 1, 22288: 1, 22287: 1, 22286: 1, 22285: 1, 2228
4: 1, 22283: 1, 22282: 1, 22281: 1, 22280: 1, 22279: 1, 22278: 1, 2227
7: 1, 22276: 1, 22275: 1, 22274: 1, 22273: 1, 22272: 1, 22271: 1, 2227
0: 1, 22269: 1, 22268: 1, 22267: 1, 22266: 1, 22265: 1, 22264: 1, 2226
3: 1, 22262: 1, 22261: 1, 22260: 1, 22259: 1, 22258: 1, 22257: 1, 2225
6: 1, 22255: 1, 22254: 1, 22253: 1, 22252: 1, 22251: 1, 22250: 1, 2224
9: 1, 22248: 1, 22247: 1, 22246: 1, 22245: 1, 22244: 1, 22243: 1, 2224
2: 1, 22241: 1, 22240: 1, 22239: 1, 22238: 1, 22237: 1, 22236: 1, 2223
5: 1, 22234: 1, 22233: 1, 22232: 1, 22231: 1, 22230: 1, 22229: 1, 2222
8: 1, 22227: 1, 22226: 1, 22225: 1, 22224: 1, 22223: 1, 22222: 1, 2222
1: 1, 22220: 1, 22219: 1, 22218: 1, 22217: 1, 22216: 1, 22215: 1, 2221
4: 1, 22213: 1, 22212: 1, 22211: 1, 22210: 1, 22209: 1, 22208: 1, 2220
7: 1, 22206: 1, 22205: 1, 22204: 1, 22203: 1, 22202: 1, 22201: 1, 2220
0: 1, 22199: 1, 22198: 1, 22197: 1, 22196: 1, 22195: 1, 22194: 1, 2219
3: 1, 22192: 1, 22191: 1, 22190: 1, 22189: 1, 22188: 1, 22187: 1, 2218
6: 1, 22185: 1, 22184: 1, 22183: 1, 22182: 1, 22181: 1, 22180: 1, 2217
9: 1, 22178: 1, 22177: 1, 22176: 1, 22175: 1, 22174: 1, 22173: 1, 2217
2: 1, 22171: 1, 22170: 1, 22169: 1, 22168: 1, 22167: 1, 22166: 1, 2216
5: 1, 22164: 1, 22163: 1, 22162: 1, 22161: 1, 22160: 1, 22159: 1, 2215
8: 1, 22157: 1, 22156: 1, 22155: 1, 22154: 1, 22153: 1, 22152: 1, 2215
1: 1, 22150: 1, 22149: 1, 22148: 1, 22147: 1, 22146: 1, 22145: 1, 2214
4: 1, 22143: 1, 22142: 1, 22141: 1, 22140: 1, 22139: 1, 22138: 1, 2213
7: 1, 22136: 1, 22135: 1, 22134: 1, 22133: 1, 22132: 1, 22131: 1, 2213
0: 1, 22129: 1, 22128: 1, 22127: 1, 22126: 1, 22125: 1, 22124: 1, 2212
3: 1, 22122: 1, 22121: 1, 22120: 1, 22119: 1, 22118: 1, 22117: 1, 2211
6: 1, 22115: 1, 22114: 1, 22113: 1, 22112: 1, 22111: 1, 22110: 1, 2210
9: 1, 22108: 1, 22107: 1, 22106: 1, 22105: 1, 22104: 1, 22103: 1, 2210
2: 1, 22101: 1, 22100: 1, 22099: 1, 22098: 1, 22097: 1, 22096: 1, 2209
5: 1, 22094: 1, 22093: 1, 22092: 1, 22091: 1, 22090: 1, 22089: 1, 2208
8: 1, 22087: 1, 22086: 1, 22085: 1, 22084: 1, 22083: 1, 22082: 1, 2208

1: 1, 22080: 1, 22079: 1, 22078: 1, 22077: 1, 22076: 1, 22075: 1, 2207
4: 1, 22073: 1, 22072: 1, 22071: 1, 22070: 1, 22069: 1, 22068: 1, 2206
7: 1, 22066: 1, 22065: 1, 22064: 1, 22063: 1, 22062: 1, 22061: 1, 2206
0: 1, 22059: 1, 22058: 1, 22057: 1, 22056: 1, 22055: 1, 22054: 1, 2205
3: 1, 22052: 1, 22051: 1, 22050: 1, 22049: 1, 22048: 1, 22047: 1, 2204
6: 1, 22045: 1, 22044: 1, 22043: 1, 22042: 1, 22041: 1, 22040: 1, 2203
9: 1, 22038: 1, 22037: 1, 22036: 1, 22035: 1, 22034: 1, 22033: 1, 2203
2: 1, 22031: 1, 22030: 1, 22029: 1, 22028: 1, 22027: 1, 22026: 1, 2202
5: 1, 22024: 1, 22023: 1, 22022: 1, 22021: 1, 22020: 1, 22019: 1, 2201
8: 1, 22017: 1, 22016: 1, 22015: 1, 22014: 1, 22013: 1, 22012: 1, 2201
1: 1, 22010: 1, 22009: 1, 22008: 1, 22007: 1, 22006: 1, 22005: 1, 2200
4: 1, 22003: 1, 22002: 1, 22001: 1, 22000: 1, 21999: 1, 21998: 1, 2199
7: 1, 21996: 1, 21995: 1, 21994: 1, 21993: 1, 21992: 1, 21991: 1, 2199
0: 1, 21989: 1, 21988: 1, 21987: 1, 21986: 1, 21985: 1, 21984: 1, 2198
3: 1, 21982: 1, 21981: 1, 21980: 1, 21979: 1, 21978: 1, 21977: 1, 2197
6: 1, 21975: 1, 21974: 1, 21973: 1, 21972: 1, 21971: 1, 21970: 1, 2196
9: 1, 21968: 1, 21967: 1, 21966: 1, 21965: 1, 21964: 1, 21963: 1, 2196
2: 1, 21961: 1, 21960: 1, 21959: 1, 21958: 1, 21957: 1, 21956: 1, 2195
5: 1, 21954: 1, 21953: 1, 21952: 1, 21951: 1, 21950: 1, 21949: 1, 2194
8: 1, 21947: 1, 21946: 1, 21945: 1, 21944: 1, 21943: 1, 21942: 1, 2194
1: 1, 21940: 1, 21939: 1, 21938: 1, 21937: 1, 21936: 1, 21935: 1, 2193
4: 1, 21933: 1, 21932: 1, 21931: 1, 21930: 1, 21929: 1, 21928: 1, 2192
7: 1, 21926: 1, 21925: 1, 21924: 1, 21923: 1, 21922: 1, 21921: 1, 2192
0: 1, 21919: 1, 21918: 1, 21917: 1, 21916: 1, 21915: 1, 21914: 1, 2191
3: 1, 21912: 1, 21911: 1, 21910: 1, 21909: 1, 21908: 1, 21907: 1, 2190
6: 1, 21905: 1, 21904: 1, 21903: 1, 21902: 1, 21901: 1, 21900: 1, 2189
9: 1, 21898: 1, 21897: 1, 21896: 1, 21895: 1, 21894: 1, 21893: 1, 2189
2: 1, 21891: 1, 21890: 1, 21889: 1, 21888: 1, 21887: 1, 21886: 1, 2188
5: 1, 21884: 1, 21883: 1, 21882: 1, 21881: 1, 21880: 1, 21879: 1, 2187
8: 1, 21877: 1, 21876: 1, 21875: 1, 21874: 1, 21873: 1, 21872: 1, 2187
1: 1, 21870: 1, 21869: 1, 21868: 1, 21867: 1, 21866: 1, 21865: 1, 2186
4: 1, 21863: 1, 21862: 1, 21861: 1, 21860: 1, 21859: 1, 21858: 1, 2185
7: 1, 21856: 1, 21855: 1, 21854: 1, 21853: 1, 21852: 1, 21851: 1, 2185
0: 1, 21849: 1, 21848: 1, 21847: 1, 21846: 1, 21845: 1, 21844: 1, 2184
3: 1, 21842: 1, 21841: 1, 21840: 1, 21839: 1, 21838: 1, 21837: 1, 2183
6: 1, 21835: 1, 21834: 1, 21833: 1, 21832: 1, 21831: 1, 21830: 1, 2182
9: 1, 21828: 1, 21827: 1, 21826: 1, 21825: 1, 21824: 1, 21823: 1, 2182
2: 1, 21821: 1, 21820: 1, 21819: 1, 21818: 1, 21817: 1, 21816: 1, 2181
5: 1, 21814: 1, 21813: 1, 21812: 1, 21811: 1, 21810: 1, 21809: 1, 2180

8: 1, 21807: 1, 21806: 1, 21805: 1, 21804: 1, 21803: 1, 21802: 1, 2180
1: 1, 21800: 1, 21799: 1, 21798: 1, 21797: 1, 21796: 1, 21795: 1, 2179
4: 1, 21793: 1, 21792: 1, 21791: 1, 21790: 1, 21789: 1, 21788: 1, 2178
7: 1, 21786: 1, 21785: 1, 21784: 1, 21783: 1, 21782: 1, 21781: 1, 2178
0: 1, 21779: 1, 21778: 1, 21777: 1, 21776: 1, 21775: 1, 21774: 1, 2177
3: 1, 21772: 1, 21771: 1, 21770: 1, 21769: 1, 21768: 1, 21767: 1, 2176
6: 1, 21765: 1, 21764: 1, 21763: 1, 21762: 1, 21761: 1, 21760: 1, 2175
9: 1, 21758: 1, 21757: 1, 21756: 1, 21755: 1, 21754: 1, 21753: 1, 2175
2: 1, 21751: 1, 21750: 1, 21749: 1, 21748: 1, 21747: 1, 21746: 1, 2174
5: 1, 21744: 1, 21743: 1, 21742: 1, 21741: 1, 21740: 1, 21739: 1, 2173
8: 1, 21737: 1, 21736: 1, 21735: 1, 21734: 1, 21733: 1, 21732: 1, 2173
1: 1, 21730: 1, 21729: 1, 21728: 1, 21727: 1, 21726: 1, 21725: 1, 2172
4: 1, 21723: 1, 21722: 1, 21721: 1, 21720: 1, 21719: 1, 21718: 1, 2171
7: 1, 21716: 1, 21715: 1, 21714: 1, 21713: 1, 21712: 1, 21711: 1, 2171
0: 1, 21709: 1, 21708: 1, 21707: 1, 21706: 1, 21705: 1, 21704: 1, 2170
3: 1, 21702: 1, 21701: 1, 21700: 1, 21699: 1, 21698: 1, 21697: 1, 2169
6: 1, 21695: 1, 21694: 1, 21693: 1, 21692: 1, 21691: 1, 21690: 1, 2168
9: 1, 21688: 1, 21687: 1, 21686: 1, 21685: 1, 21684: 1, 21683: 1, 2168
2: 1, 21681: 1, 21680: 1, 21679: 1, 21678: 1, 21677: 1, 21676: 1, 2167
5: 1, 21674: 1, 21673: 1, 21672: 1, 21671: 1, 21670: 1, 21669: 1, 2166
8: 1, 21667: 1, 21666: 1, 21665: 1, 21664: 1, 21663: 1, 21662: 1, 2166
1: 1, 21660: 1, 21659: 1, 21658: 1, 21657: 1, 21656: 1, 21655: 1, 2165
4: 1, 21653: 1, 21652: 1, 21651: 1, 21650: 1, 21649: 1, 21648: 1, 2164
7: 1, 21646: 1, 21645: 1, 21644: 1, 21643: 1, 21642: 1, 21641: 1, 2164
0: 1, 21639: 1, 21638: 1, 21637: 1, 21636: 1, 21635: 1, 21634: 1, 2163
3: 1, 21632: 1, 21631: 1, 21630: 1, 21629: 1, 21628: 1, 21627: 1, 2162
6: 1, 21625: 1, 21624: 1, 21623: 1, 21622: 1, 21621: 1, 21620: 1, 2161
9: 1, 21618: 1, 21617: 1, 21616: 1, 21615: 1, 21614: 1, 21613: 1, 2161
2: 1, 21611: 1, 21610: 1, 21609: 1, 21608: 1, 21607: 1, 21606: 1, 2160
5: 1, 21604: 1, 21603: 1, 21602: 1, 21601: 1, 21600: 1, 21599: 1, 2159
8: 1, 21597: 1, 21596: 1, 21595: 1, 21594: 1, 21593: 1, 21592: 1, 2159
1: 1, 21590: 1, 21589: 1, 21588: 1, 21587: 1, 21586: 1, 21585: 1, 2158
4: 1, 21583: 1, 21582: 1, 21581: 1, 21580: 1, 21579: 1, 21578: 1, 2157
7: 1, 21576: 1, 21575: 1, 21574: 1, 21573: 1, 21572: 1, 21571: 1, 2157
0: 1, 21569: 1, 21568: 1, 21567: 1, 21566: 1, 21565: 1, 21564: 1, 2156
3: 1, 21562: 1, 21561: 1, 21560: 1, 21559: 1, 21558: 1, 21557: 1, 2155
6: 1, 21555: 1, 21554: 1, 21553: 1, 21552: 1, 21551: 1, 21550: 1, 2154
9: 1, 21548: 1, 21547: 1, 21546: 1, 21545: 1, 21544: 1, 21543: 1, 2154
2: 1, 21541: 1, 21540: 1, 21539: 1, 21538: 1, 21537: 1, 21536: 1, 2153

5: 1, 21534: 1, 21533: 1, 21532: 1, 21531: 1, 21530: 1, 21529: 1, 2152
8: 1, 21527: 1, 21526: 1, 21525: 1, 21524: 1, 21523: 1, 21522: 1, 2152
1: 1, 21520: 1, 21519: 1, 21518: 1, 21517: 1, 21516: 1, 21515: 1, 2151
4: 1, 21513: 1, 21512: 1, 21511: 1, 21510: 1, 21509: 1, 21508: 1, 2150
7: 1, 21506: 1, 21505: 1, 21504: 1, 21503: 1, 21502: 1, 21501: 1, 2150
0: 1, 21499: 1, 21498: 1, 21497: 1, 21496: 1, 21495: 1, 21494: 1, 2149
3: 1, 21492: 1, 21491: 1, 21490: 1, 21489: 1, 21488: 1, 21487: 1, 2148
6: 1, 21485: 1, 21484: 1, 21483: 1, 21482: 1, 21481: 1, 21480: 1, 2147
9: 1, 21478: 1, 21477: 1, 21476: 1, 21475: 1, 21474: 1, 21473: 1, 2147
2: 1, 21471: 1, 21470: 1, 21469: 1, 21468: 1, 21467: 1, 21466: 1, 2146
5: 1, 21464: 1, 21463: 1, 21462: 1, 21461: 1, 21460: 1, 21459: 1, 2145
8: 1, 21457: 1, 21456: 1, 21455: 1, 21454: 1, 21453: 1, 21452: 1, 2145
1: 1, 21450: 1, 21449: 1, 21448: 1, 21447: 1, 21446: 1, 21445: 1, 2144
4: 1, 21443: 1, 21442: 1, 21441: 1, 21440: 1, 21439: 1, 21438: 1, 2143
7: 1, 21436: 1, 21435: 1, 21434: 1, 21433: 1, 21432: 1, 21431: 1, 2143
0: 1, 21429: 1, 21428: 1, 21427: 1, 21426: 1, 21425: 1, 21424: 1, 2142
3: 1, 21422: 1, 21421: 1, 21420: 1, 21419: 1, 21418: 1, 21417: 1, 2141
6: 1, 21415: 1, 21414: 1, 21413: 1, 21412: 1, 21411: 1, 21410: 1, 2140
9: 1, 21408: 1, 21407: 1, 21406: 1, 21405: 1, 21404: 1, 21403: 1, 2140
2: 1, 21401: 1, 21400: 1, 21399: 1, 21398: 1, 21397: 1, 21396: 1, 2139
5: 1, 21394: 1, 21393: 1, 21392: 1, 21391: 1, 21390: 1, 21389: 1, 2138
8: 1, 21387: 1, 21386: 1, 21385: 1, 21384: 1, 21383: 1, 21382: 1, 2138
1: 1, 21380: 1, 21379: 1, 21378: 1, 21377: 1, 21376: 1, 21375: 1, 2137
4: 1, 21373: 1, 21372: 1, 21371: 1, 21370: 1, 21369: 1, 21368: 1, 2136
7: 1, 21366: 1, 21365: 1, 21364: 1, 21363: 1, 21362: 1, 21361: 1, 2136
0: 1, 21359: 1, 21358: 1, 21357: 1, 21356: 1, 21355: 1, 21354: 1, 2135
3: 1, 21352: 1, 21351: 1, 21350: 1, 21349: 1, 21348: 1, 21347: 1, 2134
6: 1, 21345: 1, 21344: 1, 21343: 1, 21342: 1, 21341: 1, 21340: 1, 2133
9: 1, 21338: 1, 21337: 1, 21336: 1, 21335: 1, 21334: 1, 21333: 1, 2133
2: 1, 21331: 1, 21330: 1, 21329: 1, 21328: 1, 21327: 1, 21326: 1, 2132
5: 1, 21324: 1, 21323: 1, 21322: 1, 21321: 1, 21320: 1, 21319: 1, 2131
8: 1, 21317: 1, 21316: 1, 21315: 1, 21314: 1, 21313: 1, 21312: 1, 2131
1: 1, 21310: 1, 21309: 1, 21308: 1, 21307: 1, 21306: 1, 21305: 1, 2130
4: 1, 21303: 1, 21302: 1, 21301: 1, 21300: 1, 21299: 1, 21298: 1, 2129
7: 1, 21296: 1, 21295: 1, 21294: 1, 21293: 1, 21292: 1, 21291: 1, 2129
0: 1, 21289: 1, 21288: 1, 21287: 1, 21286: 1, 21285: 1, 21284: 1, 2128
3: 1, 21282: 1, 21281: 1, 21280: 1, 21279: 1, 21278: 1, 21277: 1, 2127
6: 1, 21275: 1, 21274: 1, 21273: 1, 21272: 1, 21271: 1, 21270: 1, 2126
9: 1, 21268: 1, 21267: 1, 21266: 1, 21265: 1, 21264: 1, 21263: 1, 2126

2: 1, 21261: 1, 21260: 1, 21259: 1, 21258: 1, 21257: 1, 21256: 1, 2125
5: 1, 21254: 1, 21253: 1, 21252: 1, 21251: 1, 21250: 1, 21249: 1, 2124
8: 1, 21247: 1, 21246: 1, 21245: 1, 21244: 1, 21243: 1, 21242: 1, 2124
1: 1, 21240: 1, 21239: 1, 21238: 1, 21237: 1, 21236: 1, 21235: 1, 2123
4: 1, 21233: 1, 21232: 1, 21231: 1, 21230: 1, 21229: 1, 21228: 1, 2122
7: 1, 21226: 1, 21225: 1, 21224: 1, 21223: 1, 21222: 1, 21221: 1, 2122
0: 1, 21219: 1, 21218: 1, 21217: 1, 21216: 1, 21215: 1, 21214: 1, 2121
3: 1, 21212: 1, 21211: 1, 21210: 1, 21209: 1, 21208: 1, 21207: 1, 2120
6: 1, 21205: 1, 21204: 1, 21203: 1, 21202: 1, 21201: 1, 21200: 1, 2119
9: 1, 21198: 1, 21197: 1, 21196: 1, 21195: 1, 21194: 1, 21193: 1, 2119
2: 1, 21191: 1, 21190: 1, 21189: 1, 21188: 1, 21187: 1, 21186: 1, 2118
5: 1, 21184: 1, 21183: 1, 21182: 1, 21181: 1, 21180: 1, 21179: 1, 2117
8: 1, 21177: 1, 21176: 1, 21175: 1, 21174: 1, 21173: 1, 21172: 1, 2117
1: 1, 21170: 1, 21169: 1, 21168: 1, 21167: 1, 21166: 1, 21165: 1, 2116
4: 1, 21163: 1, 21162: 1, 21161: 1, 21160: 1, 21159: 1, 21158: 1, 2115
7: 1, 21156: 1, 21155: 1, 21154: 1, 21153: 1, 21152: 1, 21151: 1, 2115
0: 1, 21149: 1, 21148: 1, 21147: 1, 21146: 1, 21145: 1, 21144: 1, 2114
3: 1, 21142: 1, 21141: 1, 21140: 1, 21139: 1, 21138: 1, 21137: 1, 2113
6: 1, 21135: 1, 21134: 1, 21133: 1, 21132: 1, 21131: 1, 21130: 1, 2112
9: 1, 21128: 1, 21127: 1, 21126: 1, 21125: 1, 21124: 1, 21123: 1, 2112
2: 1, 21121: 1, 21120: 1, 21119: 1, 21118: 1, 21117: 1, 21116: 1, 2111
5: 1, 21114: 1, 21113: 1, 21112: 1, 21111: 1, 21110: 1, 21109: 1, 2110
8: 1, 21107: 1, 21106: 1, 21105: 1, 21104: 1, 21103: 1, 21102: 1, 2110
1: 1, 21100: 1, 21099: 1, 21098: 1, 21097: 1, 21096: 1, 21095: 1, 2109
4: 1, 21093: 1, 21092: 1, 21091: 1, 21090: 1, 21089: 1, 21088: 1, 2108
7: 1, 21086: 1, 21085: 1, 21084: 1, 21083: 1, 21082: 1, 21081: 1, 2108
0: 1, 21079: 1, 21078: 1, 21077: 1, 21076: 1, 21075: 1, 21074: 1, 2107
3: 1, 21072: 1, 21071: 1, 21070: 1, 21069: 1, 21068: 1, 21067: 1, 2106
6: 1, 21065: 1, 21064: 1, 21063: 1, 21062: 1, 21061: 1, 21060: 1, 2105
9: 1, 21058: 1, 21057: 1, 21056: 1, 21055: 1, 21054: 1, 21053: 1, 2105
2: 1, 21051: 1, 21050: 1, 21049: 1, 21048: 1, 21047: 1, 21046: 1, 2104
5: 1, 21044: 1, 21043: 1, 21042: 1, 21041: 1, 21040: 1, 21039: 1, 2103
8: 1, 21037: 1, 21036: 1, 21035: 1, 21034: 1, 21033: 1, 21032: 1, 2103
1: 1, 21030: 1, 21029: 1, 21028: 1, 21027: 1, 21026: 1, 21025: 1, 2102
4: 1, 21023: 1, 21022: 1, 21021: 1, 21020: 1, 21019: 1, 21018: 1, 2101
7: 1, 21016: 1, 21015: 1, 21014: 1, 21013: 1, 21012: 1, 21011: 1, 2101
0: 1, 21009: 1, 21008: 1, 21007: 1, 21006: 1, 21005: 1, 21004: 1, 2100
3: 1, 21002: 1, 21001: 1, 21000: 1, 20999: 1, 20998: 1, 20997: 1, 2099
6: 1, 20995: 1, 20994: 1, 20993: 1, 20992: 1, 20991: 1, 20990: 1, 2098

9: 1, 20988: 1, 20987: 1, 20986: 1, 20985: 1, 20984: 1, 20983: 1, 2098
2: 1, 20981: 1, 20980: 1, 20979: 1, 20978: 1, 20977: 1, 20976: 1, 2097
5: 1, 20974: 1, 20973: 1, 20972: 1, 20971: 1, 20970: 1, 20969: 1, 2096
8: 1, 20967: 1, 20966: 1, 20965: 1, 20964: 1, 20963: 1, 20962: 1, 2096
1: 1, 20960: 1, 20959: 1, 20958: 1, 20957: 1, 20956: 1, 20955: 1, 2095
4: 1, 20953: 1, 20952: 1, 20951: 1, 20950: 1, 20949: 1, 20948: 1, 2094
7: 1, 20946: 1, 20945: 1, 20944: 1, 20943: 1, 20942: 1, 20941: 1, 2094
0: 1, 20939: 1, 20938: 1, 20937: 1, 20936: 1, 20935: 1, 20934: 1, 2093
3: 1, 20932: 1, 20931: 1, 20930: 1, 20929: 1, 20928: 1, 20927: 1, 2092
6: 1, 20925: 1, 20924: 1, 20923: 1, 20922: 1, 20921: 1, 20920: 1, 2091
9: 1, 20918: 1, 20917: 1, 20916: 1, 20915: 1, 20914: 1, 20913: 1, 2091
2: 1, 20911: 1, 20910: 1, 20909: 1, 20908: 1, 20907: 1, 20906: 1, 2090
5: 1, 20904: 1, 20903: 1, 20902: 1, 20901: 1, 20900: 1, 20899: 1, 2089
8: 1, 20897: 1, 20896: 1, 20895: 1, 20894: 1, 20893: 1, 20892: 1, 2089
1: 1, 20890: 1, 20889: 1, 20888: 1, 20887: 1, 20886: 1, 20885: 1, 2088
4: 1, 20883: 1, 20882: 1, 20881: 1, 20880: 1, 20879: 1, 20878: 1, 2087
7: 1, 20876: 1, 20875: 1, 20874: 1, 20873: 1, 20872: 1, 20871: 1, 2087
0: 1, 20869: 1, 20868: 1, 20867: 1, 20866: 1, 20865: 1, 20864: 1, 2086
3: 1, 20862: 1, 20861: 1, 20860: 1, 20859: 1, 20858: 1, 20857: 1, 2085
6: 1, 20855: 1, 20854: 1, 20853: 1, 20852: 1, 20851: 1, 20850: 1, 2084
9: 1, 20848: 1, 20847: 1, 20846: 1, 20845: 1, 20844: 1, 20843: 1, 2084
2: 1, 20841: 1, 20840: 1, 20839: 1, 20838: 1, 20837: 1, 20836: 1, 2083
5: 1, 20834: 1, 20833: 1, 20832: 1, 20831: 1, 20830: 1, 20829: 1, 2082
8: 1, 20827: 1, 20826: 1, 20825: 1, 20824: 1, 20823: 1, 20822: 1, 2082
1: 1, 20820: 1, 20819: 1, 20818: 1, 20817: 1, 20816: 1, 20815: 1, 2081
4: 1, 20813: 1, 20812: 1, 20811: 1, 20810: 1, 20809: 1, 20808: 1, 2080
7: 1, 20806: 1, 20805: 1, 20804: 1, 20803: 1, 20802: 1, 20801: 1, 2080
0: 1, 20799: 1, 20798: 1, 20797: 1, 20796: 1, 20795: 1, 20794: 1, 2079
3: 1, 20792: 1, 20791: 1, 20790: 1, 20789: 1, 20788: 1, 20787: 1, 2078
6: 1, 20785: 1, 20784: 1, 20783: 1, 20782: 1, 20781: 1, 20780: 1, 2077
9: 1, 20778: 1, 20777: 1, 20776: 1, 20775: 1, 20774: 1, 20773: 1, 2077
2: 1, 20771: 1, 20770: 1, 20769: 1, 20768: 1, 20767: 1, 20766: 1, 2076
5: 1, 20764: 1, 20763: 1, 20762: 1, 20761: 1, 20760: 1, 20759: 1, 2075
8: 1, 20757: 1, 20756: 1, 20755: 1, 20754: 1, 20753: 1, 20752: 1, 2075
1: 1, 20750: 1, 20749: 1, 20748: 1, 20747: 1, 20746: 1, 20745: 1, 2074
4: 1, 20743: 1, 20742: 1, 20741: 1, 20740: 1, 20739: 1, 20738: 1, 2073
7: 1, 20736: 1, 20735: 1, 20734: 1, 20733: 1, 20732: 1, 20731: 1, 2073
0: 1, 20729: 1, 20728: 1, 20727: 1, 20726: 1, 20725: 1, 20724: 1, 2072
3: 1, 20722: 1, 20721: 1, 20720: 1, 20719: 1, 20718: 1, 20717: 1, 2071

6: 1, 20715: 1, 20714: 1, 20713: 1, 20712: 1, 20711: 1, 20710: 1, 2070
9: 1, 20708: 1, 20707: 1, 20706: 1, 20705: 1, 20704: 1, 20703: 1, 2070
2: 1, 20701: 1, 20700: 1, 20699: 1, 20698: 1, 20697: 1, 20696: 1, 2069
5: 1, 20694: 1, 20693: 1, 20692: 1, 20691: 1, 20690: 1, 20689: 1, 2068
8: 1, 20687: 1, 20686: 1, 20685: 1, 20684: 1, 20683: 1, 20682: 1, 2068
1: 1, 20680: 1, 20679: 1, 20678: 1, 20677: 1, 20676: 1, 20675: 1, 2067
4: 1, 20673: 1, 20672: 1, 20671: 1, 20670: 1, 20669: 1, 20668: 1, 2066
7: 1, 20666: 1, 20665: 1, 20664: 1, 20663: 1, 20662: 1, 20661: 1, 2066
0: 1, 20659: 1, 20658: 1, 20657: 1, 20656: 1, 20655: 1, 20654: 1, 2065
3: 1, 20652: 1, 20651: 1, 20650: 1, 20649: 1, 20648: 1, 20647: 1, 2064
6: 1, 20645: 1, 20644: 1, 20643: 1, 20642: 1, 20641: 1, 20640: 1, 2063
9: 1, 20638: 1, 20637: 1, 20636: 1, 20635: 1, 20634: 1, 20633: 1, 2063
2: 1, 20631: 1, 20630: 1, 20629: 1, 20628: 1, 20627: 1, 20626: 1, 2062
5: 1, 20624: 1, 20623: 1, 20622: 1, 20621: 1, 20620: 1, 20619: 1, 2061
8: 1, 20617: 1, 20616: 1, 20615: 1, 20614: 1, 20613: 1, 20612: 1, 2061
1: 1, 20610: 1, 20609: 1, 20608: 1, 20607: 1, 20606: 1, 20605: 1, 2060
4: 1, 20603: 1, 20602: 1, 20601: 1, 20600: 1, 20599: 1, 20598: 1, 2059
7: 1, 20596: 1, 20595: 1, 20594: 1, 20593: 1, 20592: 1, 20591: 1, 2059
0: 1, 20589: 1, 20588: 1, 20587: 1, 20586: 1, 20585: 1, 20584: 1, 2058
3: 1, 20582: 1, 20581: 1, 20580: 1, 20579: 1, 20578: 1, 20577: 1, 2057
6: 1, 20575: 1, 20574: 1, 20573: 1, 20572: 1, 20571: 1, 20570: 1, 2056
9: 1, 20568: 1, 20567: 1, 20566: 1, 20565: 1, 20564: 1, 20563: 1, 2056
2: 1, 20561: 1, 20560: 1, 20559: 1, 20558: 1, 20557: 1, 20556: 1, 2055
5: 1, 20554: 1, 20553: 1, 20552: 1, 20551: 1, 20550: 1, 20549: 1, 2054
8: 1, 20547: 1, 20546: 1, 20545: 1, 20544: 1, 20543: 1, 20542: 1, 2054
1: 1, 20540: 1, 20539: 1, 20538: 1, 20537: 1, 20536: 1, 20535: 1, 2053
4: 1, 20533: 1, 20532: 1, 20531: 1, 20530: 1, 20529: 1, 20528: 1, 2052
7: 1, 20526: 1, 20525: 1, 20524: 1, 20523: 1, 20522: 1, 20521: 1, 2052
0: 1, 20519: 1, 20518: 1, 20517: 1, 20516: 1, 20515: 1, 20514: 1, 2051
3: 1, 20512: 1, 20511: 1, 20510: 1, 20509: 1, 20508: 1, 20507: 1, 2050
6: 1, 20505: 1, 20504: 1, 20503: 1, 20502: 1, 20501: 1, 20500: 1, 2049
9: 1, 20498: 1, 20497: 1, 20496: 1, 20495: 1, 20494: 1, 20493: 1, 2049
2: 1, 20491: 1, 20490: 1, 20489: 1, 20488: 1, 20487: 1, 20486: 1, 2048
5: 1, 20484: 1, 20483: 1, 20482: 1, 20481: 1, 20480: 1, 20479: 1, 2047
8: 1, 20477: 1, 20476: 1, 20475: 1, 20474: 1, 20473: 1, 20472: 1, 2047
1: 1, 20470: 1, 20469: 1, 20468: 1, 20467: 1, 20466: 1, 20465: 1, 2046
4: 1, 20463: 1, 20462: 1, 20461: 1, 20460: 1, 20459: 1, 20458: 1, 2045
7: 1, 20456: 1, 20455: 1, 20454: 1, 20453: 1, 20452: 1, 20451: 1, 2045
0: 1, 20449: 1, 20448: 1, 20447: 1, 20446: 1, 20445: 1, 20444: 1, 2044

3: 1, 20442: 1, 20441: 1, 20440: 1, 20439: 1, 20438: 1, 20437: 1, 2043
6: 1, 20435: 1, 20434: 1, 20433: 1, 20432: 1, 20431: 1, 20430: 1, 2042
9: 1, 20428: 1, 20427: 1, 20426: 1, 20425: 1, 20424: 1, 20423: 1, 2042
2: 1, 20421: 1, 20420: 1, 20419: 1, 20418: 1, 20417: 1, 20416: 1, 2041
5: 1, 20414: 1, 20413: 1, 20412: 1, 20411: 1, 20410: 1, 20409: 1, 2040
8: 1, 20407: 1, 20406: 1, 20405: 1, 20404: 1, 20403: 1, 20402: 1, 2040
1: 1, 20400: 1, 20399: 1, 20398: 1, 20397: 1, 20396: 1, 20395: 1, 2039
4: 1, 20393: 1, 20392: 1, 20391: 1, 20390: 1, 20389: 1, 20388: 1, 2038
7: 1, 20386: 1, 20385: 1, 20384: 1, 20383: 1, 20382: 1, 20381: 1, 2038
0: 1, 20379: 1, 20378: 1, 20377: 1, 20376: 1, 20375: 1, 20374: 1, 2037
3: 1, 20372: 1, 20371: 1, 20370: 1, 20369: 1, 20368: 1, 20367: 1, 2036
6: 1, 20365: 1, 20364: 1, 20363: 1, 20362: 1, 20361: 1, 20360: 1, 2035
9: 1, 20358: 1, 20357: 1, 20356: 1, 20355: 1, 20354: 1, 20353: 1, 2035
2: 1, 20351: 1, 20350: 1, 20349: 1, 20348: 1, 20347: 1, 20346: 1, 2034
5: 1, 20344: 1, 20343: 1, 20342: 1, 20341: 1, 20340: 1, 20339: 1, 2033
8: 1, 20337: 1, 20336: 1, 20335: 1, 20334: 1, 20333: 1, 20332: 1, 2033
1: 1, 20330: 1, 20329: 1, 20328: 1, 20327: 1, 20326: 1, 20325: 1, 2032
4: 1, 20323: 1, 20322: 1, 20321: 1, 20320: 1, 20319: 1, 20318: 1, 2031
7: 1, 20316: 1, 20315: 1, 20314: 1, 20313: 1, 20312: 1, 20311: 1, 2031
0: 1, 20309: 1, 20308: 1, 20307: 1, 20306: 1, 20305: 1, 20304: 1, 2030
3: 1, 20302: 1, 20301: 1, 20300: 1, 20299: 1, 20298: 1, 20297: 1, 2029
6: 1, 20295: 1, 20294: 1, 20293: 1, 20292: 1, 20291: 1, 20290: 1, 2028
9: 1, 20288: 1, 20287: 1, 20286: 1, 20285: 1, 20284: 1, 20283: 1, 2028
2: 1, 20281: 1, 20280: 1, 20279: 1, 20278: 1, 20277: 1, 20276: 1, 2027
5: 1, 20274: 1, 20273: 1, 20272: 1, 20271: 1, 20270: 1, 20269: 1, 2026
8: 1, 20267: 1, 20266: 1, 20265: 1, 20264: 1, 20263: 1, 20262: 1, 2026
1: 1, 20260: 1, 20259: 1, 20258: 1, 20257: 1, 20256: 1, 20255: 1, 2025
4: 1, 20253: 1, 20252: 1, 20251: 1, 20250: 1, 20249: 1, 20248: 1, 2024
7: 1, 20246: 1, 20245: 1, 20244: 1, 20243: 1, 20242: 1, 20241: 1, 2024
0: 1, 20239: 1, 20238: 1, 20237: 1, 20236: 1, 20235: 1, 20234: 1, 2023
3: 1, 20232: 1, 20231: 1, 20230: 1, 20229: 1, 20228: 1, 20227: 1, 2022
6: 1, 20225: 1, 20224: 1, 20223: 1, 20222: 1, 20221: 1, 20220: 1, 2021
9: 1, 20218: 1, 20217: 1, 20216: 1, 20215: 1, 20214: 1, 20213: 1, 2021
2: 1, 20211: 1, 20210: 1, 20209: 1, 20208: 1, 20207: 1, 20206: 1, 2020
5: 1, 20204: 1, 20203: 1, 20202: 1, 20201: 1, 20200: 1, 20199: 1, 2019
8: 1, 20197: 1, 20196: 1, 20195: 1, 20194: 1, 20193: 1, 20192: 1, 2019
1: 1, 20190: 1, 20189: 1, 20188: 1, 20187: 1, 20186: 1, 20185: 1, 2018
4: 1, 20183: 1, 20182: 1, 20181: 1, 20180: 1, 20179: 1, 20178: 1, 2017
7: 1, 20176: 1, 20175: 1, 20174: 1, 20173: 1, 20172: 1, 20171: 1, 2017

0: 1, 20169: 1, 20168: 1, 20167: 1, 20166: 1, 20165: 1, 20164: 1, 2016
3: 1, 20162: 1, 20161: 1, 20160: 1, 20159: 1, 20158: 1, 20157: 1, 2015
6: 1, 20155: 1, 20154: 1, 20153: 1, 20152: 1, 20151: 1, 20150: 1, 2014
9: 1, 20148: 1, 20147: 1, 20146: 1, 20145: 1, 20144: 1, 20143: 1, 2014
2: 1, 20141: 1, 20140: 1, 20139: 1, 20138: 1, 20137: 1, 20136: 1, 2013
5: 1, 20134: 1, 20133: 1, 20132: 1, 20131: 1, 20130: 1, 20129: 1, 2012
8: 1, 20127: 1, 20126: 1, 20125: 1, 20124: 1, 20123: 1, 20122: 1, 2012
1: 1, 20120: 1, 20119: 1, 20118: 1, 20117: 1, 20116: 1, 20115: 1, 2011
4: 1, 20113: 1, 20112: 1, 20111: 1, 20110: 1, 20109: 1, 20108: 1, 2010
7: 1, 20106: 1, 20105: 1, 20104: 1, 20103: 1, 20102: 1, 20101: 1, 2010
0: 1, 20099: 1, 20098: 1, 20097: 1, 20096: 1, 20095: 1, 20094: 1, 2009
3: 1, 20092: 1, 20091: 1, 20090: 1, 20089: 1, 20088: 1, 20087: 1, 2008
6: 1, 20085: 1, 20084: 1, 20083: 1, 20082: 1, 20081: 1, 20080: 1, 2007
9: 1, 20078: 1, 20077: 1, 20076: 1, 20075: 1, 20074: 1, 20073: 1, 2007
2: 1, 20071: 1, 20070: 1, 20069: 1, 20068: 1, 20067: 1, 20066: 1, 2006
5: 1, 20064: 1, 20063: 1, 20062: 1, 20061: 1, 20060: 1, 20059: 1, 2005
8: 1, 20057: 1, 20056: 1, 20055: 1, 20054: 1, 20053: 1, 20052: 1, 2005
1: 1, 20050: 1, 20049: 1, 20048: 1, 20047: 1, 20046: 1, 20045: 1, 2004
4: 1, 20043: 1, 20042: 1, 20041: 1, 20040: 1, 20039: 1, 20038: 1, 2003
7: 1, 20036: 1, 20035: 1, 20034: 1, 20033: 1, 20032: 1, 20031: 1, 2003
0: 1, 20029: 1, 20028: 1, 20027: 1, 20026: 1, 20025: 1, 20024: 1, 2002
3: 1, 20022: 1, 20021: 1, 20020: 1, 20019: 1, 20018: 1, 20017: 1, 2001
6: 1, 20015: 1, 20014: 1, 20013: 1, 20012: 1, 20011: 1, 20010: 1, 2000
9: 1, 20008: 1, 20007: 1, 20006: 1, 20005: 1, 20004: 1, 20003: 1, 2000
2: 1, 20001: 1, 20000: 1, 19999: 1, 19998: 1, 19997: 1, 19996: 1, 1999
5: 1, 19994: 1, 19993: 1, 19992: 1, 19991: 1, 19990: 1, 19989: 1, 1998
8: 1, 19987: 1, 19986: 1, 19985: 1, 19984: 1, 19983: 1, 19982: 1, 1998
1: 1, 19980: 1, 19979: 1, 19978: 1, 19977: 1, 19976: 1, 19975: 1, 1997
4: 1, 19973: 1, 19972: 1, 19971: 1, 19970: 1, 19969: 1, 19968: 1, 1996
7: 1, 19966: 1, 19965: 1, 19964: 1, 19963: 1, 19962: 1, 19961: 1, 1996
0: 1, 19959: 1, 19958: 1, 19957: 1, 19956: 1, 19955: 1, 19954: 1, 1995
3: 1, 19952: 1, 19951: 1, 19950: 1, 19949: 1, 19948: 1, 19947: 1, 1994
6: 1, 19945: 1, 19944: 1, 19943: 1, 19942: 1, 19941: 1, 19940: 1, 1993
9: 1, 19938: 1, 19937: 1, 19936: 1, 19935: 1, 19934: 1, 19933: 1, 1993
2: 1, 19931: 1, 19930: 1, 19929: 1, 19928: 1, 19927: 1, 19926: 1, 1992
5: 1, 19924: 1, 19923: 1, 19922: 1, 19921: 1, 19920: 1, 19919: 1, 1991
8: 1, 19917: 1, 19916: 1, 19915: 1, 19914: 1, 19913: 1, 19912: 1, 1991
1: 1, 19910: 1, 19909: 1, 19908: 1, 19907: 1, 19906: 1, 19905: 1, 1990
4: 1, 19903: 1, 19902: 1, 19901: 1, 19900: 1, 19899: 1, 19898: 1, 1989

7: 1, 19896: 1, 19895: 1, 19894: 1, 19893: 1, 19892: 1, 19891: 1, 1989
0: 1, 19889: 1, 19888: 1, 19887: 1, 19886: 1, 19885: 1, 19884: 1, 1988
3: 1, 19882: 1, 19881: 1, 19880: 1, 19879: 1, 19878: 1, 19877: 1, 1987
6: 1, 19875: 1, 19874: 1, 19873: 1, 19872: 1, 19871: 1, 19870: 1, 1986
9: 1, 19868: 1, 19867: 1, 19866: 1, 19865: 1, 19864: 1, 19863: 1, 1986
2: 1, 19861: 1, 19860: 1, 19859: 1, 19858: 1, 19857: 1, 19856: 1, 1985
5: 1, 19854: 1, 19853: 1, 19852: 1, 19851: 1, 19850: 1, 19849: 1, 1984
8: 1, 19847: 1, 19846: 1, 19845: 1, 19844: 1, 19843: 1, 19842: 1, 1984
1: 1, 19840: 1, 19839: 1, 19838: 1, 19837: 1, 19836: 1, 19835: 1, 1983
4: 1, 19833: 1, 19832: 1, 19831: 1, 19830: 1, 19829: 1, 19828: 1, 1982
7: 1, 19826: 1, 19825: 1, 19824: 1, 19823: 1, 19822: 1, 19821: 1, 1982
0: 1, 19819: 1, 19818: 1, 19817: 1, 19816: 1, 19815: 1, 19814: 1, 1981
3: 1, 19812: 1, 19811: 1, 19810: 1, 19809: 1, 19808: 1, 19807: 1, 1980
6: 1, 19805: 1, 19804: 1, 19803: 1, 19802: 1, 19801: 1, 19800: 1, 1979
9: 1, 19798: 1, 19797: 1, 19796: 1, 19795: 1, 19794: 1, 19793: 1, 1979
2: 1, 19791: 1, 19790: 1, 19789: 1, 19788: 1, 19787: 1, 19786: 1, 1978
5: 1, 19784: 1, 19783: 1, 19782: 1, 19781: 1, 19780: 1, 19779: 1, 1977
8: 1, 19777: 1, 19776: 1, 19775: 1, 19774: 1, 19773: 1, 19772: 1, 1977
1: 1, 19770: 1, 19769: 1, 19768: 1, 19767: 1, 19766: 1, 19765: 1, 1976
4: 1, 19763: 1, 19762: 1, 19761: 1, 19760: 1, 19759: 1, 19758: 1, 1975
7: 1, 19756: 1, 19755: 1, 19754: 1, 19753: 1, 19752: 1, 19751: 1, 1975
0: 1, 19749: 1, 19748: 1, 19747: 1, 19746: 1, 19745: 1, 19744: 1, 1974
3: 1, 19742: 1, 19741: 1, 19740: 1, 19739: 1, 19738: 1, 19737: 1, 1973
6: 1, 19735: 1, 19734: 1, 19733: 1, 19732: 1, 19731: 1, 19730: 1, 1972
9: 1, 19728: 1, 19727: 1, 19726: 1, 19725: 1, 19724: 1, 19723: 1, 1972
2: 1, 19721: 1, 19720: 1, 19719: 1, 19718: 1, 19717: 1, 19716: 1, 1971
5: 1, 19714: 1, 19713: 1, 19712: 1, 19711: 1, 19710: 1, 19709: 1, 1970
8: 1, 19707: 1, 19706: 1, 19705: 1, 19704: 1, 19703: 1, 19702: 1, 1970
1: 1, 19700: 1, 19699: 1, 19698: 1, 19697: 1, 19696: 1, 19695: 1, 1969
4: 1, 19693: 1, 19692: 1, 19691: 1, 19690: 1, 19689: 1, 19688: 1, 1968
7: 1, 19686: 1, 19685: 1, 19684: 1, 19683: 1, 19682: 1, 19681: 1, 1968
0: 1, 19679: 1, 19678: 1, 19677: 1, 19676: 1, 19675: 1, 19674: 1, 1967
3: 1, 19672: 1, 19671: 1, 19670: 1, 19669: 1, 19668: 1, 19667: 1, 1966
6: 1, 19665: 1, 19664: 1, 19663: 1, 19662: 1, 19661: 1, 19660: 1, 1965
9: 1, 19658: 1, 19657: 1, 19656: 1, 19655: 1, 19654: 1, 19653: 1, 1965
2: 1, 19651: 1, 19650: 1, 19649: 1, 19648: 1, 19647: 1, 19646: 1, 1964
5: 1, 19644: 1, 19643: 1, 19642: 1, 19641: 1, 19640: 1, 19639: 1, 1963
8: 1, 19637: 1, 19636: 1, 19635: 1, 19634: 1, 19633: 1, 19632: 1, 1963
1: 1, 19630: 1, 19629: 1, 19628: 1, 19627: 1, 19626: 1, 19625: 1, 1962

4: 1, 19623: 1, 19622: 1, 19621: 1, 19620: 1, 19619: 1, 19618: 1, 1961
7: 1, 19616: 1, 19615: 1, 19614: 1, 19613: 1, 19612: 1, 19611: 1, 1961
0: 1, 19609: 1, 19608: 1, 19607: 1, 19606: 1, 19605: 1, 19604: 1, 1960
3: 1, 19602: 1, 19601: 1, 19600: 1, 19599: 1, 19598: 1, 19597: 1, 1959
6: 1, 19595: 1, 19594: 1, 19593: 1, 19592: 1, 19591: 1, 19590: 1, 1958
9: 1, 19588: 1, 19587: 1, 19586: 1, 19585: 1, 19584: 1, 19583: 1, 1958
2: 1, 19581: 1, 19580: 1, 19579: 1, 19578: 1, 19577: 1, 19576: 1, 1957
5: 1, 19574: 1, 19573: 1, 19572: 1, 19571: 1, 19570: 1, 19569: 1, 1956
8: 1, 19567: 1, 19566: 1, 19565: 1, 19564: 1, 19563: 1, 19562: 1, 1956
1: 1, 19560: 1, 19559: 1, 19558: 1, 19557: 1, 19556: 1, 19555: 1, 1955
4: 1, 19553: 1, 19552: 1, 19551: 1, 19550: 1, 19549: 1, 19548: 1, 1954
7: 1, 19546: 1, 19545: 1, 19544: 1, 19543: 1, 19542: 1, 19541: 1, 1954
0: 1, 19539: 1, 19538: 1, 19537: 1, 19536: 1, 19535: 1, 19534: 1, 1953
3: 1, 19532: 1, 19531: 1, 19530: 1, 19529: 1, 19528: 1, 19527: 1, 1952
6: 1, 19525: 1, 19524: 1, 19523: 1, 19522: 1, 19521: 1, 19520: 1, 1951
9: 1, 19518: 1, 19517: 1, 19516: 1, 19515: 1, 19514: 1, 19513: 1, 1951
2: 1, 19511: 1, 19510: 1, 19509: 1, 19508: 1, 19507: 1, 19506: 1, 1950
5: 1, 19504: 1, 19503: 1, 19502: 1, 19501: 1, 19500: 1, 19499: 1, 1949
8: 1, 19497: 1, 19496: 1, 19495: 1, 19494: 1, 19493: 1, 19492: 1, 1949
1: 1, 19490: 1, 19489: 1, 19488: 1, 19487: 1, 19486: 1, 19485: 1, 1948
4: 1, 19483: 1, 19482: 1, 19481: 1, 19480: 1, 19479: 1, 19478: 1, 1947
7: 1, 19476: 1, 19475: 1, 19474: 1, 19473: 1, 19472: 1, 19471: 1, 1947
0: 1, 19469: 1, 19468: 1, 19467: 1, 19466: 1, 19465: 1, 19464: 1, 1946
3: 1, 19462: 1, 19461: 1, 19460: 1, 19459: 1, 19458: 1, 19457: 1, 1945
6: 1, 19455: 1, 19454: 1, 19453: 1, 19452: 1, 19451: 1, 19450: 1, 1944
9: 1, 19448: 1, 19447: 1, 19446: 1, 19445: 1, 19444: 1, 19443: 1, 1944
2: 1, 19441: 1, 19440: 1, 19439: 1, 19438: 1, 19437: 1, 19436: 1, 1943
5: 1, 19434: 1, 19433: 1, 19432: 1, 19431: 1, 19430: 1, 19429: 1, 1942
8: 1, 19427: 1, 19426: 1, 19425: 1, 19424: 1, 19423: 1, 19422: 1, 1942
1: 1, 19420: 1, 19419: 1, 19418: 1, 19417: 1, 19416: 1, 19415: 1, 1941
4: 1, 19413: 1, 19412: 1, 19411: 1, 19410: 1, 19409: 1, 19408: 1, 1940
7: 1, 19406: 1, 19405: 1, 19404: 1, 19403: 1, 19402: 1, 19401: 1, 1940
0: 1, 19399: 1, 19398: 1, 19397: 1, 19396: 1, 19395: 1, 19394: 1, 1939
3: 1, 19392: 1, 19391: 1, 19390: 1, 19389: 1, 19388: 1, 19387: 1, 1938
6: 1, 19385: 1, 19384: 1, 19383: 1, 19382: 1, 19381: 1, 19380: 1, 1937
9: 1, 19378: 1, 19377: 1, 19376: 1, 19375: 1, 19374: 1, 19373: 1, 1937
2: 1, 19371: 1, 19370: 1, 19369: 1, 19368: 1, 19367: 1, 19366: 1, 1936
5: 1, 19364: 1, 19363: 1, 19362: 1, 19361: 1, 19360: 1, 19359: 1, 1935
8: 1, 19357: 1, 19356: 1, 19355: 1, 19354: 1, 19353: 1, 19352: 1, 1935

1: 1, 19350: 1, 19349: 1, 19348: 1, 19347: 1, 19346: 1, 19345: 1, 1934
4: 1, 19343: 1, 19342: 1, 19341: 1, 19340: 1, 19339: 1, 19338: 1, 1933
7: 1, 19336: 1, 19335: 1, 19334: 1, 19333: 1, 19332: 1, 19331: 1, 1933
0: 1, 19329: 1, 19328: 1, 19327: 1, 19326: 1, 19325: 1, 19324: 1, 1932
3: 1, 19322: 1, 19321: 1, 19320: 1, 19319: 1, 19318: 1, 19317: 1, 1931
6: 1, 19315: 1, 19314: 1, 19313: 1, 19312: 1, 19311: 1, 19310: 1, 1930
9: 1, 19308: 1, 19307: 1, 19306: 1, 19305: 1, 19304: 1, 19303: 1, 1930
2: 1, 19301: 1, 19300: 1, 19299: 1, 19298: 1, 19297: 1, 19296: 1, 1929
5: 1, 19294: 1, 19293: 1, 19292: 1, 19291: 1, 19290: 1, 19289: 1, 1928
8: 1, 19287: 1, 19286: 1, 19285: 1, 19284: 1, 19283: 1, 19282: 1, 1928
1: 1, 19280: 1, 19279: 1, 19278: 1, 19277: 1, 19276: 1, 19275: 1, 1927
4: 1, 19273: 1, 19272: 1, 19271: 1, 19270: 1, 19269: 1, 19268: 1, 1926
7: 1, 19266: 1, 19265: 1, 19264: 1, 19263: 1, 19262: 1, 19261: 1, 1926
0: 1, 19259: 1, 19258: 1, 19257: 1, 19256: 1, 19255: 1, 19254: 1, 1925
3: 1, 19252: 1, 19251: 1, 19250: 1, 19249: 1, 19248: 1, 19247: 1, 1924
6: 1, 19245: 1, 19244: 1, 19243: 1, 19242: 1, 19241: 1, 19240: 1, 1923
9: 1, 19238: 1, 19237: 1, 19236: 1, 19235: 1, 19234: 1, 19233: 1, 1923
2: 1, 19231: 1, 19230: 1, 19229: 1, 19228: 1, 19227: 1, 19226: 1, 1922
5: 1, 19224: 1, 19223: 1, 19222: 1, 19221: 1, 19220: 1, 19219: 1, 1921
8: 1, 19217: 1, 19216: 1, 19215: 1, 19214: 1, 19213: 1, 19212: 1, 1921
1: 1, 19210: 1, 19209: 1, 19208: 1, 19207: 1, 19206: 1, 19205: 1, 1920
4: 1, 19203: 1, 19202: 1, 19201: 1, 19200: 1, 19199: 1, 19198: 1, 1919
7: 1, 19196: 1, 19195: 1, 19194: 1, 19193: 1, 19192: 1, 19191: 1, 1919
0: 1, 19189: 1, 19188: 1, 19187: 1, 19186: 1, 19185: 1, 19184: 1, 1918
3: 1, 19182: 1, 19181: 1, 19180: 1, 19179: 1, 19178: 1, 19177: 1, 1917
6: 1, 19175: 1, 19174: 1, 19173: 1, 19172: 1, 19171: 1, 19170: 1, 1916
9: 1, 19168: 1, 19167: 1, 19166: 1, 19165: 1, 19164: 1, 19163: 1, 1916
2: 1, 19161: 1, 19160: 1, 19159: 1, 19158: 1, 19157: 1, 19156: 1, 1915
5: 1, 19154: 1, 19153: 1, 19152: 1, 19151: 1, 19150: 1, 19149: 1, 1914
8: 1, 19147: 1, 19146: 1, 19145: 1, 19144: 1, 19143: 1, 19142: 1, 1914
1: 1, 19140: 1, 19139: 1, 19138: 1, 19137: 1, 19136: 1, 19135: 1, 1913
4: 1, 19133: 1, 19132: 1, 19131: 1, 19130: 1, 19129: 1, 19128: 1, 1912
7: 1, 19126: 1, 19125: 1, 19124: 1, 19123: 1, 19122: 1, 19121: 1, 1912
0: 1, 19119: 1, 19118: 1, 19117: 1, 19116: 1, 19115: 1, 19114: 1, 1911
3: 1, 19112: 1, 19111: 1, 19110: 1, 19109: 1, 19108: 1, 19107: 1, 1910
6: 1, 19105: 1, 19104: 1, 19103: 1, 19102: 1, 19101: 1, 19100: 1, 1909
9: 1, 19098: 1, 19097: 1, 19096: 1, 19095: 1, 19094: 1, 19093: 1, 1909
2: 1, 19091: 1, 19090: 1, 19089: 1, 19088: 1, 19087: 1, 19086: 1, 1908
5: 1, 19084: 1, 19083: 1, 19082: 1, 19081: 1, 19080: 1, 19079: 1, 1907

8: 1, 19077: 1, 19076: 1, 19075: 1, 19074: 1, 19073: 1, 19072: 1, 1907
1: 1, 19070: 1, 19069: 1, 19068: 1, 19067: 1, 19066: 1, 19065: 1, 1906
4: 1, 19063: 1, 19062: 1, 19061: 1, 19060: 1, 19059: 1, 19058: 1, 1905
7: 1, 19056: 1, 19055: 1, 19054: 1, 19053: 1, 19052: 1, 19051: 1, 1905
0: 1, 19049: 1, 19048: 1, 19047: 1, 19046: 1, 19045: 1, 19044: 1, 1904
3: 1, 19042: 1, 19041: 1, 19040: 1, 19039: 1, 19038: 1, 19037: 1, 1903
6: 1, 19035: 1, 19034: 1, 19033: 1, 19032: 1, 19031: 1, 19030: 1, 1902
9: 1, 19028: 1, 19027: 1, 19026: 1, 19025: 1, 19024: 1, 19023: 1, 1902
2: 1, 19021: 1, 19020: 1, 19019: 1, 19018: 1, 19017: 1, 19016: 1, 1901
5: 1, 19014: 1, 19013: 1, 19012: 1, 19011: 1, 19010: 1, 19009: 1, 1900
8: 1, 19007: 1, 19006: 1, 19005: 1, 19004: 1, 19003: 1, 19002: 1, 1900
1: 1, 19000: 1, 18999: 1, 18998: 1, 18997: 1, 18996: 1, 18995: 1, 1899
4: 1, 18993: 1, 18992: 1, 18991: 1, 18990: 1, 18989: 1, 18988: 1, 1898
7: 1, 18986: 1, 18985: 1, 18984: 1, 18983: 1, 18982: 1, 18981: 1, 1898
0: 1, 18979: 1, 18978: 1, 18977: 1, 18976: 1, 18975: 1, 18974: 1, 1897
3: 1, 18972: 1, 18971: 1, 18970: 1, 18969: 1, 18968: 1, 18967: 1, 1896
6: 1, 18965: 1, 18964: 1, 18963: 1, 18962: 1, 18961: 1, 18960: 1, 1895
9: 1, 18958: 1, 18957: 1, 18956: 1, 18955: 1, 18954: 1, 18953: 1, 1895
2: 1, 18951: 1, 18950: 1, 18949: 1, 18948: 1, 18947: 1, 18946: 1, 1894
5: 1, 18944: 1, 18943: 1, 18942: 1, 18941: 1, 18940: 1, 18939: 1, 1893
8: 1, 18937: 1, 18936: 1, 18935: 1, 18934: 1, 18933: 1, 18932: 1, 1893
1: 1, 18930: 1, 18929: 1, 18928: 1, 18927: 1, 18926: 1, 18925: 1, 1892
4: 1, 18923: 1, 18922: 1, 18921: 1, 18920: 1, 18919: 1, 18918: 1, 1891
7: 1, 18916: 1, 18915: 1, 18914: 1, 18913: 1, 18912: 1, 18911: 1, 1891
0: 1, 18909: 1, 18908: 1, 18907: 1, 18906: 1, 18905: 1, 18904: 1, 1890
3: 1, 18902: 1, 18901: 1, 18900: 1, 18899: 1, 18898: 1, 18897: 1, 1889
6: 1, 18895: 1, 18894: 1, 18893: 1, 18892: 1, 18891: 1, 18890: 1, 1888
9: 1, 18888: 1, 18887: 1, 18886: 1, 18885: 1, 18884: 1, 18883: 1, 1888
2: 1, 18881: 1, 18880: 1, 18879: 1, 18878: 1, 18877: 1, 18876: 1, 1887
5: 1, 18874: 1, 18873: 1, 18872: 1, 18871: 1, 18870: 1, 18869: 1, 1886
8: 1, 18867: 1, 18866: 1, 18865: 1, 18864: 1, 18863: 1, 18862: 1, 1886
1: 1, 18860: 1, 18859: 1, 18858: 1, 18857: 1, 18856: 1, 18855: 1, 1885
4: 1, 18853: 1, 18852: 1, 18851: 1, 18850: 1, 18849: 1, 18848: 1, 1884
7: 1, 18846: 1, 18845: 1, 18844: 1, 18843: 1, 18842: 1, 18841: 1, 1884
0: 1, 18839: 1, 18838: 1, 18837: 1, 18836: 1, 18835: 1, 18834: 1, 1883
3: 1, 18832: 1, 18831: 1, 18830: 1, 18829: 1, 18828: 1, 18827: 1, 1882
6: 1, 18825: 1, 18824: 1, 18823: 1, 18822: 1, 18821: 1, 18820: 1, 1881
9: 1, 18818: 1, 18817: 1, 18816: 1, 18815: 1, 18814: 1, 18813: 1, 1881
2: 1, 18811: 1, 18810: 1, 18809: 1, 18808: 1, 18807: 1, 18806: 1, 1880

5: 1, 18804: 1, 18803: 1, 18802: 1, 18801: 1, 18800: 1, 18799: 1, 1879
8: 1, 18797: 1, 18796: 1, 18795: 1, 18794: 1, 18793: 1, 18792: 1, 1879
1: 1, 18790: 1, 18789: 1, 18788: 1, 18787: 1, 18786: 1, 18785: 1, 1878
4: 1, 18783: 1, 18782: 1, 18781: 1, 18780: 1, 18779: 1, 18778: 1, 1877
7: 1, 18776: 1, 18775: 1, 18774: 1, 18773: 1, 18772: 1, 18771: 1, 1877
0: 1, 18769: 1, 18768: 1, 18767: 1, 18766: 1, 18765: 1, 18764: 1, 1876
3: 1, 18762: 1, 18761: 1, 18760: 1, 18759: 1, 18758: 1, 18757: 1, 1875
6: 1, 18755: 1, 18754: 1, 18753: 1, 18752: 1, 18751: 1, 18750: 1, 1874
9: 1, 18748: 1, 18747: 1, 18746: 1, 18745: 1, 18744: 1, 18743: 1, 1874
2: 1, 18741: 1, 18740: 1, 18739: 1, 18738: 1, 18737: 1, 18736: 1, 1873
5: 1, 18734: 1, 18733: 1, 18732: 1, 18731: 1, 18730: 1, 18729: 1, 1872
8: 1, 18727: 1, 18726: 1, 18725: 1, 18724: 1, 18723: 1, 18722: 1, 1872
1: 1, 18720: 1, 18719: 1, 18718: 1, 18717: 1, 18716: 1, 18715: 1, 1871
4: 1, 18713: 1, 18712: 1, 18711: 1, 18710: 1, 18709: 1, 18708: 1, 1870
7: 1, 18706: 1, 18705: 1, 18704: 1, 18703: 1, 18702: 1, 18701: 1, 1870
0: 1, 18699: 1, 18698: 1, 18697: 1, 18696: 1, 18695: 1, 18694: 1, 1869
3: 1, 18692: 1, 18691: 1, 18690: 1, 18689: 1, 18688: 1, 18687: 1, 1868
6: 1, 18685: 1, 18684: 1, 18683: 1, 18682: 1, 18681: 1, 18680: 1, 1867
9: 1, 18678: 1, 18677: 1, 18676: 1, 18675: 1, 18674: 1, 18673: 1, 1867
2: 1, 18671: 1, 18670: 1, 18669: 1, 18668: 1, 18667: 1, 18666: 1, 1866
5: 1, 18664: 1, 18663: 1, 18662: 1, 18661: 1, 18660: 1, 18659: 1, 1865
8: 1, 18657: 1, 18656: 1, 18655: 1, 18654: 1, 18653: 1, 18652: 1, 1865
1: 1, 18650: 1, 18649: 1, 18648: 1, 18647: 1, 18646: 1, 18645: 1, 1864
4: 1, 18643: 1, 18642: 1, 18641: 1, 18640: 1, 18639: 1, 18638: 1, 1863
7: 1, 18636: 1, 18635: 1, 18634: 1, 18633: 1, 18632: 1, 18631: 1, 1863
0: 1, 18629: 1, 18628: 1, 18627: 1, 18626: 1, 18625: 1, 18624: 1, 1862
3: 1, 18622: 1, 18621: 1, 18620: 1, 18619: 1, 18618: 1, 18617: 1, 1861
6: 1, 18615: 1, 18614: 1, 18613: 1, 18612: 1, 18611: 1, 18610: 1, 1860
9: 1, 18608: 1, 18607: 1, 18606: 1, 18605: 1, 18604: 1, 18603: 1, 1860
2: 1, 18601: 1, 18600: 1, 18599: 1, 18598: 1, 18597: 1, 18596: 1, 1859
5: 1, 18594: 1, 18593: 1, 18592: 1, 18591: 1, 18590: 1, 18589: 1, 1858
8: 1, 18587: 1, 18586: 1, 18585: 1, 18584: 1, 18583: 1, 18582: 1, 1858
1: 1, 18580: 1, 18579: 1, 18578: 1, 18577: 1, 18576: 1, 18575: 1, 1857
4: 1, 18573: 1, 18572: 1, 18571: 1, 18570: 1, 18569: 1, 18568: 1, 1856
7: 1, 18566: 1, 18565: 1, 18564: 1, 18563: 1, 18562: 1, 18561: 1, 1856
0: 1, 18559: 1, 18558: 1, 18557: 1, 18556: 1, 18555: 1, 18554: 1, 1855
3: 1, 18552: 1, 18551: 1, 18550: 1, 18549: 1, 18548: 1, 18547: 1, 1854
6: 1, 18545: 1, 18544: 1, 18543: 1, 18542: 1, 18541: 1, 18540: 1, 1853
9: 1, 18538: 1, 18537: 1, 18536: 1, 18535: 1, 18534: 1, 18533: 1, 1853

2: 1, 18531: 1, 18530: 1, 18529: 1, 18528: 1, 18527: 1, 18526: 1, 1852
5: 1, 18524: 1, 18523: 1, 18522: 1, 18521: 1, 18520: 1, 18519: 1, 1851
8: 1, 18517: 1, 18516: 1, 18515: 1, 18514: 1, 18513: 1, 18512: 1, 1851
1: 1, 18510: 1, 18509: 1, 18508: 1, 18507: 1, 18506: 1, 18505: 1, 1850
4: 1, 18503: 1, 18502: 1, 18501: 1, 18500: 1, 18499: 1, 18498: 1, 1849
7: 1, 18496: 1, 18495: 1, 18494: 1, 18493: 1, 18492: 1, 18491: 1, 1849
0: 1, 18489: 1, 18488: 1, 18487: 1, 18486: 1, 18485: 1, 18484: 1, 1848
3: 1, 18482: 1, 18481: 1, 18480: 1, 18479: 1, 18478: 1, 18477: 1, 1847
6: 1, 18475: 1, 18474: 1, 18473: 1, 18472: 1, 18471: 1, 18470: 1, 1846
9: 1, 18468: 1, 18467: 1, 18466: 1, 18465: 1, 18464: 1, 18463: 1, 1846
2: 1, 18461: 1, 18460: 1, 18459: 1, 18458: 1, 18457: 1, 18456: 1, 1845
5: 1, 18454: 1, 18453: 1, 18452: 1, 18451: 1, 18450: 1, 18449: 1, 1844
8: 1, 18447: 1, 18446: 1, 18445: 1, 18444: 1, 18443: 1, 18442: 1, 1844
1: 1, 18440: 1, 18439: 1, 18438: 1, 18437: 1, 18436: 1, 18435: 1, 1843
4: 1, 18433: 1, 18432: 1, 18431: 1, 18430: 1, 18429: 1, 18428: 1, 1842
7: 1, 18426: 1, 18425: 1, 18424: 1, 18423: 1, 18422: 1, 18421: 1, 1842
0: 1, 18419: 1, 18418: 1, 18417: 1, 18416: 1, 18415: 1, 18414: 1, 1841
3: 1, 18412: 1, 18411: 1, 18410: 1, 18409: 1, 18408: 1, 18407: 1, 1840
6: 1, 18405: 1, 18404: 1, 18403: 1, 18402: 1, 18401: 1, 18400: 1, 1839
9: 1, 18398: 1, 18397: 1, 18396: 1, 18395: 1, 18394: 1, 18393: 1, 1839
2: 1, 18391: 1, 18390: 1, 18389: 1, 18388: 1, 18387: 1, 18386: 1, 1838
5: 1, 18384: 1, 18383: 1, 18382: 1, 18381: 1, 18380: 1, 18379: 1, 1837
8: 1, 18377: 1, 18376: 1, 18375: 1, 18374: 1, 18373: 1, 18372: 1, 1837
1: 1, 18370: 1, 18369: 1, 18368: 1, 18367: 1, 18366: 1, 18365: 1, 1836
4: 1, 18363: 1, 18362: 1, 18361: 1, 18360: 1, 18359: 1, 18358: 1, 1835
7: 1, 18356: 1, 18355: 1, 18354: 1, 18353: 1, 18352: 1, 18351: 1, 1835
0: 1, 18349: 1, 18348: 1, 18347: 1, 18346: 1, 18345: 1, 18344: 1, 1834
3: 1, 18342: 1, 18341: 1, 18340: 1, 18339: 1, 18338: 1, 18337: 1, 1833
6: 1, 18335: 1, 18334: 1, 18333: 1, 18332: 1, 18331: 1, 18330: 1, 1832
9: 1, 18328: 1, 18327: 1, 18326: 1, 18325: 1, 18324: 1, 18323: 1, 1832
2: 1, 18321: 1, 18320: 1, 18319: 1, 18318: 1, 18317: 1, 18316: 1, 1831
5: 1, 18314: 1, 18313: 1, 18312: 1, 18311: 1, 18310: 1, 18309: 1, 1830
8: 1, 18307: 1, 18306: 1, 18305: 1, 18304: 1, 18303: 1, 18302: 1, 1830
1: 1, 18300: 1, 18299: 1, 18298: 1, 18297: 1, 18296: 1, 18295: 1, 1829
4: 1, 18293: 1, 18292: 1, 18291: 1, 18290: 1, 18289: 1, 18288: 1, 1828
7: 1, 18286: 1, 18285: 1, 18284: 1, 18283: 1, 18282: 1, 18281: 1, 1828
0: 1, 18279: 1, 18278: 1, 18277: 1, 18276: 1, 18275: 1, 18274: 1, 1827
3: 1, 18272: 1, 18271: 1, 18270: 1, 18269: 1, 18268: 1, 18267: 1, 1826
6: 1, 18265: 1, 18264: 1, 18263: 1, 18262: 1, 18261: 1, 18260: 1, 1825

9: 1, 18258: 1, 18257: 1, 18256: 1, 18255: 1, 18254: 1, 18253: 1, 1825
2: 1, 18251: 1, 18250: 1, 18249: 1, 18248: 1, 18247: 1, 18246: 1, 1824
5: 1, 18244: 1, 18243: 1, 18242: 1, 18241: 1, 18240: 1, 18239: 1, 1823
8: 1, 18237: 1, 18236: 1, 18235: 1, 18234: 1, 18233: 1, 18232: 1, 1823
1: 1, 18230: 1, 18229: 1, 18228: 1, 18227: 1, 18226: 1, 18225: 1, 1822
4: 1, 18223: 1, 18222: 1, 18221: 1, 18220: 1, 18219: 1, 18218: 1, 1821
7: 1, 18216: 1, 18215: 1, 18214: 1, 18213: 1, 18212: 1, 18211: 1, 1821
0: 1, 18209: 1, 18208: 1, 18207: 1, 18206: 1, 18205: 1, 18204: 1, 1820
3: 1, 18202: 1, 18201: 1, 18200: 1, 18199: 1, 18198: 1, 18197: 1, 1819
6: 1, 18195: 1, 18194: 1, 18193: 1, 18192: 1, 18191: 1, 18190: 1, 1818
9: 1, 18188: 1, 18187: 1, 18186: 1, 18185: 1, 18184: 1, 18183: 1, 1818
2: 1, 18181: 1, 18180: 1, 18179: 1, 18178: 1, 18177: 1, 18176: 1, 1817
5: 1, 18174: 1, 18173: 1, 18172: 1, 18171: 1, 18170: 1, 18169: 1, 1816
8: 1, 18167: 1, 18166: 1, 18165: 1, 18164: 1, 18163: 1, 18162: 1, 1816
1: 1, 18160: 1, 18159: 1, 18158: 1, 18157: 1, 18156: 1, 18155: 1, 1815
4: 1, 18153: 1, 18152: 1, 18151: 1, 18150: 1, 18149: 1, 18148: 1, 1814
7: 1, 18146: 1, 18145: 1, 18144: 1, 18143: 1, 18142: 1, 18141: 1, 1814
0: 1, 18139: 1, 18138: 1, 18137: 1, 18136: 1, 18135: 1, 18134: 1, 1813
3: 1, 18132: 1, 18131: 1, 18130: 1, 18129: 1, 18128: 1, 18127: 1, 1812
6: 1, 18125: 1, 18124: 1, 18123: 1, 18122: 1, 18121: 1, 18120: 1, 1811
9: 1, 18118: 1, 18117: 1, 18116: 1, 18115: 1, 18114: 1, 18113: 1, 1811
2: 1, 18111: 1, 18110: 1, 18109: 1, 18108: 1, 18107: 1, 18106: 1, 1810
5: 1, 18104: 1, 18103: 1, 18102: 1, 18101: 1, 18100: 1, 18099: 1, 1809
8: 1, 18097: 1, 18096: 1, 18095: 1, 18094: 1, 18093: 1, 18092: 1, 1809
1: 1, 18090: 1, 18089: 1, 18088: 1, 18087: 1, 18086: 1, 18085: 1, 1808
4: 1, 18083: 1, 18082: 1, 18081: 1, 18080: 1, 18079: 1, 18078: 1, 1807
7: 1, 18076: 1, 18075: 1, 18074: 1, 18073: 1, 18072: 1, 18071: 1, 1807
0: 1, 18069: 1, 18068: 1, 18067: 1, 18066: 1, 18065: 1, 18064: 1, 1806
3: 1, 18062: 1, 18061: 1, 18060: 1, 18059: 1, 18058: 1, 18057: 1, 1805
6: 1, 18055: 1, 18054: 1, 18053: 1, 18052: 1, 18051: 1, 18050: 1, 1804
9: 1, 18048: 1, 18047: 1, 18046: 1, 18045: 1, 18044: 1, 18043: 1, 1804
2: 1, 18041: 1, 18040: 1, 18039: 1, 18038: 1, 18037: 1, 18036: 1, 1803
5: 1, 18034: 1, 18033: 1, 18032: 1, 18031: 1, 18030: 1, 18029: 1, 1802
8: 1, 18027: 1, 18026: 1, 18025: 1, 18024: 1, 18023: 1, 18022: 1, 1802
1: 1, 18020: 1, 18019: 1, 18018: 1, 18017: 1, 18016: 1, 18015: 1, 1801
4: 1, 18013: 1, 18012: 1, 18011: 1, 18010: 1, 18009: 1, 18008: 1, 1800
7: 1, 18006: 1, 18005: 1, 18004: 1, 18003: 1, 18002: 1, 18001: 1, 1800
0: 1, 17999: 1, 17998: 1, 17997: 1, 17996: 1, 17995: 1, 17994: 1, 1799
3: 1, 17992: 1, 17991: 1, 17990: 1, 17989: 1, 17988: 1, 17987: 1, 1798

6: 1, 17985: 1, 17984: 1, 17983: 1, 17982: 1, 17981: 1, 17980: 1, 1797
9: 1, 17978: 1, 17977: 1, 17976: 1, 17975: 1, 17974: 1, 17973: 1, 1797
2: 1, 17971: 1, 17970: 1, 17969: 1, 17968: 1, 17967: 1, 17966: 1, 1796
5: 1, 17964: 1, 17963: 1, 17962: 1, 17961: 1, 17960: 1, 17959: 1, 1795
8: 1, 17957: 1, 17956: 1, 17955: 1, 17954: 1, 17953: 1, 17952: 1, 1795
1: 1, 17950: 1, 17949: 1, 17948: 1, 17947: 1, 17946: 1, 17945: 1, 1794
4: 1, 17943: 1, 17942: 1, 17941: 1, 17940: 1, 17939: 1, 17938: 1, 1793
7: 1, 17936: 1, 17935: 1, 17934: 1, 17933: 1, 17932: 1, 17931: 1, 1793
0: 1, 17929: 1, 17928: 1, 17927: 1, 17926: 1, 17925: 1, 17924: 1, 1792
3: 1, 17922: 1, 17921: 1, 17920: 1, 17919: 1, 17918: 1, 17917: 1, 1791
6: 1, 17915: 1, 17914: 1, 17913: 1, 17912: 1, 17911: 1, 17910: 1, 1790
9: 1, 17908: 1, 17907: 1, 17906: 1, 17905: 1, 17904: 1, 17903: 1, 1790
2: 1, 17901: 1, 17900: 1, 17899: 1, 17898: 1, 17897: 1, 17896: 1, 1789
5: 1, 17894: 1, 17893: 1, 17892: 1, 17891: 1, 17890: 1, 17889: 1, 1788
8: 1, 17887: 1, 17886: 1, 17885: 1, 17884: 1, 17883: 1, 17882: 1, 1788
1: 1, 17880: 1, 17879: 1, 17878: 1, 17877: 1, 17876: 1, 17875: 1, 1787
4: 1, 17873: 1, 17872: 1, 17871: 1, 17870: 1, 17869: 1, 17868: 1, 1786
7: 1, 17866: 1, 17865: 1, 17864: 1, 17863: 1, 17862: 1, 17861: 1, 1786
0: 1, 17859: 1, 17858: 1, 17857: 1, 17856: 1, 17855: 1, 17854: 1, 1785
3: 1, 17852: 1, 17851: 1, 17850: 1, 17849: 1, 17848: 1, 17847: 1, 1784
6: 1, 17845: 1, 17844: 1, 17843: 1, 17842: 1, 17841: 1, 17840: 1, 1783
9: 1, 17838: 1, 17837: 1, 17836: 1, 17835: 1, 17834: 1, 17833: 1, 1783
2: 1, 17831: 1, 17830: 1, 17829: 1, 17828: 1, 17827: 1, 17826: 1, 1782
5: 1, 17824: 1, 17823: 1, 17822: 1, 17821: 1, 17820: 1, 17819: 1, 1781
8: 1, 17817: 1, 17816: 1, 17815: 1, 17814: 1, 17813: 1, 17812: 1, 1781
1: 1, 17810: 1, 17809: 1, 17808: 1, 17807: 1, 17806: 1, 17805: 1, 1780
4: 1, 17803: 1, 17802: 1, 17801: 1, 17800: 1, 17799: 1, 17798: 1, 1779
7: 1, 17796: 1, 17795: 1, 17794: 1, 17793: 1, 17792: 1, 17791: 1, 1779
0: 1, 17789: 1, 17788: 1, 17787: 1, 17786: 1, 17785: 1, 17784: 1, 1778
3: 1, 17782: 1, 17781: 1, 17780: 1, 17779: 1, 17778: 1, 17777: 1, 1777
6: 1, 17775: 1, 17774: 1, 17773: 1, 17772: 1, 17771: 1, 17770: 1, 1776
9: 1, 17768: 1, 17767: 1, 17766: 1, 17765: 1, 17764: 1, 17763: 1, 1776
2: 1, 17761: 1, 17760: 1, 17759: 1, 17758: 1, 17757: 1, 17756: 1, 1775
5: 1, 17754: 1, 17753: 1, 17752: 1, 17751: 1, 17750: 1, 17749: 1, 1774
8: 1, 17747: 1, 17746: 1, 17745: 1, 17744: 1, 17743: 1, 17742: 1, 1774
1: 1, 17740: 1, 17739: 1, 17738: 1, 17737: 1, 17736: 1, 17735: 1, 1773
4: 1, 17733: 1, 17732: 1, 17731: 1, 17730: 1, 17729: 1, 17728: 1, 1772
7: 1, 17726: 1, 17725: 1, 17724: 1, 17723: 1, 17722: 1, 17721: 1, 1772
0: 1, 17719: 1, 17718: 1, 17717: 1, 17716: 1, 17715: 1, 17714: 1, 1771

3: 1, 17712: 1, 17711: 1, 17710: 1, 17709: 1, 17708: 1, 17707: 1, 1770
6: 1, 17705: 1, 17704: 1, 17703: 1, 17702: 1, 17701: 1, 17700: 1, 1769
9: 1, 17698: 1, 17697: 1, 17696: 1, 17695: 1, 17694: 1, 17693: 1, 1769
2: 1, 17691: 1, 17690: 1, 17689: 1, 17688: 1, 17687: 1, 17686: 1, 1768
5: 1, 17684: 1, 17683: 1, 17682: 1, 17681: 1, 17680: 1, 17679: 1, 1767
8: 1, 17677: 1, 17676: 1, 17675: 1, 17674: 1, 17673: 1, 17672: 1, 1767
1: 1, 17670: 1, 17669: 1, 17668: 1, 17667: 1, 17666: 1, 17665: 1, 1766
4: 1, 17663: 1, 17662: 1, 17661: 1, 17660: 1, 17659: 1, 17658: 1, 1765
7: 1, 17656: 1, 17655: 1, 17654: 1, 17653: 1, 17652: 1, 17651: 1, 1765
0: 1, 17649: 1, 17648: 1, 17647: 1, 17646: 1, 17645: 1, 17644: 1, 1764
3: 1, 17642: 1, 17641: 1, 17640: 1, 17639: 1, 17638: 1, 17637: 1, 1763
6: 1, 17635: 1, 17634: 1, 17633: 1, 17632: 1, 17631: 1, 17630: 1, 1762
9: 1, 17628: 1, 17627: 1, 17626: 1, 17625: 1, 17624: 1, 17623: 1, 1762
2: 1, 17621: 1, 17620: 1, 17619: 1, 17618: 1, 17617: 1, 17616: 1, 1761
5: 1, 17614: 1, 17613: 1, 17612: 1, 17611: 1, 17610: 1, 17609: 1, 1760
8: 1, 17607: 1, 17606: 1, 17605: 1, 17604: 1, 17603: 1, 17602: 1, 1760
1: 1, 17600: 1, 17599: 1, 17598: 1, 17597: 1, 17596: 1, 17595: 1, 1759
4: 1, 17593: 1, 17592: 1, 17591: 1, 17590: 1, 17589: 1, 17588: 1, 1758
7: 1, 17586: 1, 17585: 1, 17584: 1, 17583: 1, 17582: 1, 17581: 1, 1758
0: 1, 17579: 1, 17578: 1, 17577: 1, 17576: 1, 17575: 1, 17574: 1, 1757
3: 1, 17572: 1, 17571: 1, 17570: 1, 17569: 1, 17568: 1, 17567: 1, 1756
6: 1, 17565: 1, 17564: 1, 17563: 1, 17562: 1, 17561: 1, 17560: 1, 1755
9: 1, 17558: 1, 17557: 1, 17556: 1, 17555: 1, 17554: 1, 17553: 1, 1755
2: 1, 17551: 1, 17550: 1, 17549: 1, 17548: 1, 17547: 1, 17546: 1, 1754
5: 1, 17544: 1, 17543: 1, 17542: 1, 17541: 1, 17540: 1, 17539: 1, 1753
8: 1, 17537: 1, 17536: 1, 17535: 1, 17534: 1, 17533: 1, 17532: 1, 1753
1: 1, 17530: 1, 17529: 1, 17528: 1, 17527: 1, 17526: 1, 17525: 1, 1752
4: 1, 17523: 1, 17522: 1, 17521: 1, 17520: 1, 17519: 1, 17518: 1, 1751
7: 1, 17516: 1, 17515: 1, 17514: 1, 17513: 1, 17512: 1, 17511: 1, 1751
0: 1, 17509: 1, 17508: 1, 17507: 1, 17506: 1, 17505: 1, 17504: 1, 1750
3: 1, 17502: 1, 17501: 1, 17500: 1, 17499: 1, 17498: 1, 17497: 1, 1749
6: 1, 17495: 1, 17494: 1, 17493: 1, 17492: 1, 17491: 1, 17490: 1, 1748
9: 1, 17488: 1, 17487: 1, 17486: 1, 17485: 1, 17484: 1, 17483: 1, 1748
2: 1, 17481: 1, 17480: 1, 17479: 1, 17478: 1, 17477: 1, 17476: 1, 1747
5: 1, 17474: 1, 17473: 1, 17472: 1, 17471: 1, 17470: 1, 17469: 1, 1746
8: 1, 17467: 1, 17466: 1, 17465: 1, 17464: 1, 17463: 1, 17462: 1, 1746
1: 1, 17460: 1, 17459: 1, 17458: 1, 17457: 1, 17456: 1, 17455: 1, 1745
4: 1, 17453: 1, 17452: 1, 17451: 1, 17450: 1, 17449: 1, 17448: 1, 1744
7: 1, 17446: 1, 17445: 1, 17444: 1, 17443: 1, 17442: 1, 17441: 1, 1744

0: 1, 17439: 1, 17438: 1, 17437: 1, 17436: 1, 17435: 1, 17434: 1, 1743
3: 1, 17432: 1, 17431: 1, 17430: 1, 17429: 1, 17428: 1, 17427: 1, 1742
6: 1, 17425: 1, 17424: 1, 17423: 1, 17422: 1, 17421: 1, 17420: 1, 1741
9: 1, 17418: 1, 17417: 1, 17416: 1, 17415: 1, 17414: 1, 17413: 1, 1741
2: 1, 17411: 1, 17410: 1, 17409: 1, 17408: 1, 17407: 1, 17406: 1, 1740
5: 1, 17404: 1, 17403: 1, 17402: 1, 17401: 1, 17400: 1, 17399: 1, 1739
8: 1, 17397: 1, 17396: 1, 17395: 1, 17394: 1, 17393: 1, 17392: 1, 1739
1: 1, 17390: 1, 17389: 1, 17388: 1, 17387: 1, 17386: 1, 17385: 1, 1738
4: 1, 17383: 1, 17382: 1, 17381: 1, 17380: 1, 17379: 1, 17378: 1, 1737
7: 1, 17376: 1, 17375: 1, 17374: 1, 17373: 1, 17372: 1, 17371: 1, 1737
0: 1, 17369: 1, 17368: 1, 17367: 1, 17366: 1, 17365: 1, 17364: 1, 1736
3: 1, 17362: 1, 17361: 1, 17360: 1, 17359: 1, 17358: 1, 17357: 1, 1735
6: 1, 17355: 1, 17354: 1, 17353: 1, 17352: 1, 17351: 1, 17350: 1, 1734
9: 1, 17348: 1, 17347: 1, 17346: 1, 17345: 1, 17344: 1, 17343: 1, 1734
2: 1, 17341: 1, 17340: 1, 17339: 1, 17338: 1, 17337: 1, 17336: 1, 1733
5: 1, 17334: 1, 17333: 1, 17332: 1, 17331: 1, 17330: 1, 17329: 1, 1732
8: 1, 17327: 1, 17326: 1, 17325: 1, 17324: 1, 17323: 1, 17322: 1, 1732
1: 1, 17320: 1, 17319: 1, 17318: 1, 17317: 1, 17316: 1, 17315: 1, 1731
4: 1, 17313: 1, 17312: 1, 17311: 1, 17310: 1, 17309: 1, 17308: 1, 1730
7: 1, 17306: 1, 17305: 1, 17304: 1, 17303: 1, 17302: 1, 17301: 1, 1730
0: 1, 17299: 1, 17298: 1, 17297: 1, 17296: 1, 17295: 1, 17294: 1, 1729
3: 1, 17292: 1, 17291: 1, 17290: 1, 17289: 1, 17288: 1, 17287: 1, 1728
6: 1, 17285: 1, 17284: 1, 17283: 1, 17282: 1, 17281: 1, 17280: 1, 1727
9: 1, 17278: 1, 17277: 1, 17276: 1, 17275: 1, 17274: 1, 17273: 1, 1727
2: 1, 17271: 1, 17270: 1, 17269: 1, 17268: 1, 17267: 1, 17266: 1, 1726
5: 1, 17264: 1, 17263: 1, 17262: 1, 17261: 1, 17260: 1, 17259: 1, 1725
8: 1, 17257: 1, 17256: 1, 17255: 1, 17254: 1, 17253: 1, 17252: 1, 1725
1: 1, 17250: 1, 17249: 1, 17248: 1, 17247: 1, 17246: 1, 17245: 1, 1724
4: 1, 17243: 1, 17242: 1, 17241: 1, 17240: 1, 17239: 1, 17238: 1, 1723
7: 1, 17236: 1, 17235: 1, 17234: 1, 17233: 1, 17232: 1, 17231: 1, 1723
0: 1, 17229: 1, 17228: 1, 17227: 1, 17226: 1, 17225: 1, 17224: 1, 1722
3: 1, 17222: 1, 17221: 1, 17220: 1, 17219: 1, 17218: 1, 17217: 1, 1721
6: 1, 17215: 1, 17214: 1, 17213: 1, 17212: 1, 17211: 1, 17210: 1, 1720
9: 1, 17208: 1, 17207: 1, 17206: 1, 17205: 1, 17204: 1, 17203: 1, 1720
2: 1, 17201: 1, 17200: 1, 17199: 1, 17198: 1, 17197: 1, 17196: 1, 1719
5: 1, 17194: 1, 17193: 1, 17192: 1, 17191: 1, 17190: 1, 17189: 1, 1718
8: 1, 17187: 1, 17186: 1, 17185: 1, 17184: 1, 17183: 1, 17182: 1, 1718
1: 1, 17180: 1, 17179: 1, 17178: 1, 17177: 1, 17176: 1, 17175: 1, 1717
4: 1, 17173: 1, 17172: 1, 17171: 1, 17170: 1, 17169: 1, 17168: 1, 1716

7: 1, 17166: 1, 17165: 1, 17164: 1, 17163: 1, 17162: 1, 17161: 1, 1716
0: 1, 17159: 1, 17158: 1, 17157: 1, 17156: 1, 17155: 1, 17154: 1, 1715
3: 1, 17152: 1, 17151: 1, 17150: 1, 17149: 1, 17148: 1, 17147: 1, 1714
6: 1, 17145: 1, 17144: 1, 17143: 1, 17142: 1, 17141: 1, 17140: 1, 1713
9: 1, 17138: 1, 17137: 1, 17136: 1, 17135: 1, 17134: 1, 17133: 1, 1713
2: 1, 17131: 1, 17130: 1, 17129: 1, 17128: 1, 17127: 1, 17126: 1, 1712
5: 1, 17124: 1, 17123: 1, 17122: 1, 17121: 1, 17120: 1, 17119: 1, 1711
8: 1, 17117: 1, 17116: 1, 17115: 1, 17114: 1, 17113: 1, 17112: 1, 1711
1: 1, 17110: 1, 17109: 1, 17108: 1, 17107: 1, 17106: 1, 17105: 1, 1710
4: 1, 17103: 1, 17102: 1, 17101: 1, 17100: 1, 17099: 1, 17098: 1, 1709
7: 1, 17096: 1, 17095: 1, 17094: 1, 17093: 1, 17092: 1, 17091: 1, 1709
0: 1, 17089: 1, 17088: 1, 17087: 1, 17086: 1, 17085: 1, 17084: 1, 1708
3: 1, 17082: 1, 17081: 1, 17080: 1, 17079: 1, 17078: 1, 17077: 1, 1707
6: 1, 17075: 1, 17074: 1, 17073: 1, 17072: 1, 17071: 1, 17070: 1, 1706
9: 1, 17068: 1, 17067: 1, 17066: 1, 17065: 1, 17064: 1, 17063: 1, 1706
2: 1, 17061: 1, 17060: 1, 17059: 1, 17058: 1, 17057: 1, 17056: 1, 1705
5: 1, 17054: 1, 17053: 1, 17052: 1, 17051: 1, 17050: 1, 17049: 1, 1704
8: 1, 17047: 1, 17046: 1, 17045: 1, 17044: 1, 17043: 1, 17042: 1, 1704
1: 1, 17040: 1, 17039: 1, 17038: 1, 17037: 1, 17036: 1, 17035: 1, 1703
4: 1, 17033: 1, 17032: 1, 17031: 1, 17030: 1, 17029: 1, 17028: 1, 1702
7: 1, 17026: 1, 17025: 1, 17024: 1, 17023: 1, 17022: 1, 17021: 1, 1702
0: 1, 17019: 1, 17018: 1, 17017: 1, 17016: 1, 17015: 1, 17014: 1, 1701
3: 1, 17012: 1, 17011: 1, 17010: 1, 17009: 1, 17008: 1, 17007: 1, 1700
6: 1, 17005: 1, 17004: 1, 17003: 1, 17002: 1, 17001: 1, 17000: 1, 1699
9: 1, 16998: 1, 16997: 1, 16996: 1, 16995: 1, 16994: 1, 16993: 1, 1699
2: 1, 16991: 1, 16990: 1, 16989: 1, 16988: 1, 16987: 1, 16986: 1, 1698
5: 1, 16984: 1, 16983: 1, 16982: 1, 16981: 1, 16980: 1, 16979: 1, 1697
8: 1, 16977: 1, 16976: 1, 16975: 1, 16974: 1, 16973: 1, 16972: 1, 1697
1: 1, 16970: 1, 16969: 1, 16968: 1, 16967: 1, 16966: 1, 16965: 1, 1696
4: 1, 16963: 1, 16962: 1, 16961: 1, 16960: 1, 16959: 1, 16958: 1, 1695
7: 1, 16956: 1, 16955: 1, 16954: 1, 16953: 1, 16952: 1, 16951: 1, 1695
0: 1, 16949: 1, 16948: 1, 16947: 1, 16946: 1, 16945: 1, 16944: 1, 1694
3: 1, 16942: 1, 16941: 1, 16940: 1, 16939: 1, 16938: 1, 16937: 1, 1693
6: 1, 16935: 1, 16934: 1, 16933: 1, 16932: 1, 16931: 1, 16930: 1, 1692
9: 1, 16928: 1, 16927: 1, 16926: 1, 16925: 1, 16924: 1, 16923: 1, 1692
2: 1, 16921: 1, 16920: 1, 16919: 1, 16918: 1, 16917: 1, 16916: 1, 1691
5: 1, 16914: 1, 16913: 1, 16912: 1, 16911: 1, 16910: 1, 16909: 1, 1690
8: 1, 16907: 1, 16906: 1, 16905: 1, 16904: 1, 16903: 1, 16902: 1, 1690
1: 1, 16900: 1, 16899: 1, 16898: 1, 16897: 1, 16896: 1, 16895: 1, 1689

4: 1, 16893: 1, 16892: 1, 16891: 1, 16890: 1, 16889: 1, 16888: 1, 1688
7: 1, 16886: 1, 16885: 1, 16884: 1, 16883: 1, 16882: 1, 16881: 1, 1688
0: 1, 16879: 1, 16878: 1, 16877: 1, 16876: 1, 16875: 1, 16874: 1, 1687
3: 1, 16872: 1, 16871: 1, 16870: 1, 16869: 1, 16868: 1, 16867: 1, 1686
6: 1, 16865: 1, 16864: 1, 16863: 1, 16862: 1, 16861: 1, 16860: 1, 1685
9: 1, 16858: 1, 16857: 1, 16856: 1, 16855: 1, 16854: 1, 16853: 1, 1685
2: 1, 16851: 1, 16850: 1, 16849: 1, 16848: 1, 16847: 1, 16846: 1, 1684
5: 1, 16844: 1, 16843: 1, 16842: 1, 16841: 1, 16840: 1, 16839: 1, 1683
8: 1, 16837: 1, 16836: 1, 16835: 1, 16834: 1, 16833: 1, 16832: 1, 1683
1: 1, 16830: 1, 16829: 1, 16828: 1, 16827: 1, 16826: 1, 16825: 1, 1682
4: 1, 16823: 1, 16822: 1, 16821: 1, 16820: 1, 16819: 1, 16818: 1, 1681
7: 1, 16816: 1, 16815: 1, 16814: 1, 16813: 1, 16812: 1, 16811: 1, 1681
0: 1, 16809: 1, 16808: 1, 16807: 1, 16806: 1, 16805: 1, 16804: 1, 1680
3: 1, 16802: 1, 16801: 1, 16800: 1, 16799: 1, 16798: 1, 16797: 1, 1679
6: 1, 16795: 1, 16794: 1, 16793: 1, 16792: 1, 16791: 1, 16790: 1, 1678
9: 1, 16788: 1, 16787: 1, 16786: 1, 16785: 1, 16784: 1, 16783: 1, 1678
2: 1, 16781: 1, 16780: 1, 16779: 1, 16778: 1, 16777: 1, 16776: 1, 1677
5: 1, 16774: 1, 16773: 1, 16772: 1, 16771: 1, 16770: 1, 16769: 1, 1676
8: 1, 16767: 1, 16766: 1, 16765: 1, 16764: 1, 16763: 1, 16762: 1, 1676
1: 1, 16760: 1, 16759: 1, 16758: 1, 16757: 1, 16756: 1, 16755: 1, 1675
4: 1, 16753: 1, 16752: 1, 16751: 1, 16750: 1, 16749: 1, 16748: 1, 1674
7: 1, 16746: 1, 16745: 1, 16744: 1, 16743: 1, 16742: 1, 16741: 1, 1674
0: 1, 16739: 1, 16738: 1, 16737: 1, 16736: 1, 16735: 1, 16734: 1, 1673
3: 1, 16732: 1, 16731: 1, 16730: 1, 16729: 1, 16728: 1, 16727: 1, 1672
6: 1, 16725: 1, 16724: 1, 16723: 1, 16722: 1, 16721: 1, 16720: 1, 1671
9: 1, 16718: 1, 16717: 1, 16716: 1, 16715: 1, 16714: 1, 16713: 1, 1671
2: 1, 16711: 1, 16710: 1, 16709: 1, 16708: 1, 16707: 1, 16706: 1, 1670
5: 1, 16704: 1, 16703: 1, 16702: 1, 16701: 1, 16700: 1, 16699: 1, 1669
8: 1, 16697: 1, 16696: 1, 16695: 1, 16694: 1, 16693: 1, 16692: 1, 1669
1: 1, 16690: 1, 16689: 1, 16688: 1, 16687: 1, 16686: 1, 16685: 1, 1668
4: 1, 16683: 1, 16682: 1, 16681: 1, 16680: 1, 16679: 1, 16678: 1, 1667
7: 1, 16676: 1, 16675: 1, 16674: 1, 16673: 1, 16672: 1, 16671: 1, 1667
0: 1, 16669: 1, 16668: 1, 16667: 1, 16666: 1, 16665: 1, 16664: 1, 1666
3: 1, 16662: 1, 16661: 1, 16660: 1, 16659: 1, 16658: 1, 16657: 1, 1665
6: 1, 16655: 1, 16654: 1, 16653: 1, 16652: 1, 16651: 1, 16650: 1, 1664
9: 1, 16648: 1, 16647: 1, 16646: 1, 16645: 1, 16644: 1, 16643: 1, 1664
2: 1, 16641: 1, 16640: 1, 16639: 1, 16638: 1, 16637: 1, 16636: 1, 1663
5: 1, 16634: 1, 16633: 1, 16632: 1, 16631: 1, 16630: 1, 16629: 1, 1662
8: 1, 16627: 1, 16626: 1, 16625: 1, 16624: 1, 16623: 1, 16622: 1, 1662

1: 1, 16620: 1, 16619: 1, 16618: 1, 16617: 1, 16616: 1, 16615: 1, 1661
4: 1, 16613: 1, 16612: 1, 16611: 1, 16610: 1, 16609: 1, 16608: 1, 1660
7: 1, 16606: 1, 16605: 1, 16604: 1, 16603: 1, 16602: 1, 16601: 1, 1660
0: 1, 16599: 1, 16598: 1, 16597: 1, 16596: 1, 16595: 1, 16594: 1, 1659
3: 1, 16592: 1, 16591: 1, 16590: 1, 16589: 1, 16588: 1, 16587: 1, 1658
6: 1, 16585: 1, 16584: 1, 16583: 1, 16582: 1, 16581: 1, 16580: 1, 1657
9: 1, 16578: 1, 16577: 1, 16576: 1, 16575: 1, 16574: 1, 16573: 1, 1657
2: 1, 16571: 1, 16570: 1, 16569: 1, 16568: 1, 16567: 1, 16566: 1, 1656
5: 1, 16564: 1, 16563: 1, 16562: 1, 16561: 1, 16560: 1, 16559: 1, 1655
8: 1, 16557: 1, 16556: 1, 16555: 1, 16554: 1, 16553: 1, 16552: 1, 1655
1: 1, 16550: 1, 16549: 1, 16548: 1, 16547: 1, 16546: 1, 16545: 1, 1654
4: 1, 16543: 1, 16542: 1, 16541: 1, 16540: 1, 16539: 1, 16538: 1, 1653
7: 1, 16536: 1, 16535: 1, 16534: 1, 16533: 1, 16532: 1, 16531: 1, 1653
0: 1, 16529: 1, 16528: 1, 16527: 1, 16526: 1, 16525: 1, 16524: 1, 1652
3: 1, 16522: 1, 16521: 1, 16520: 1, 16519: 1, 16518: 1, 16517: 1, 1651
6: 1, 16515: 1, 16514: 1, 16513: 1, 16512: 1, 16511: 1, 16510: 1, 1650
9: 1, 16508: 1, 16507: 1, 16506: 1, 16505: 1, 16504: 1, 16503: 1, 1650
2: 1, 16501: 1, 16500: 1, 16499: 1, 16498: 1, 16497: 1, 16496: 1, 1649
5: 1, 16494: 1, 16493: 1, 16492: 1, 16491: 1, 16490: 1, 16489: 1, 1648
8: 1, 16487: 1, 16486: 1, 16485: 1, 16484: 1, 16483: 1, 16482: 1, 1648
1: 1, 16480: 1, 16479: 1, 16478: 1, 16477: 1, 16476: 1, 16475: 1, 1647
4: 1, 16473: 1, 16472: 1, 16471: 1, 16470: 1, 16469: 1, 16468: 1, 1646
7: 1, 16466: 1, 16465: 1, 16464: 1, 16463: 1, 16462: 1, 16461: 1, 1646
0: 1, 16459: 1, 16458: 1, 16457: 1, 16456: 1, 16455: 1, 16454: 1, 1645
3: 1, 16452: 1, 16451: 1, 16450: 1, 16449: 1, 16448: 1, 16447: 1, 1644
6: 1, 16445: 1, 16444: 1, 16443: 1, 16442: 1, 16441: 1, 16440: 1, 1643
9: 1, 16438: 1, 16437: 1, 16436: 1, 16435: 1, 16434: 1, 16433: 1, 1643
2: 1, 16431: 1, 16430: 1, 16429: 1, 16428: 1, 16427: 1, 16426: 1, 1642
5: 1, 16424: 1, 16423: 1, 16422: 1, 16421: 1, 16420: 1, 16419: 1, 1641
8: 1, 16417: 1, 16416: 1, 16415: 1, 16414: 1, 16413: 1, 16412: 1, 1641
1: 1, 16410: 1, 16409: 1, 16408: 1, 16407: 1, 16406: 1, 16405: 1, 1640
4: 1, 16403: 1, 16402: 1, 16401: 1, 16400: 1, 16399: 1, 16398: 1, 1639
7: 1, 16396: 1, 16395: 1, 16394: 1, 16393: 1, 16392: 1, 16391: 1, 1639
0: 1, 16389: 1, 16388: 1, 16387: 1, 16386: 1, 16385: 1, 16384: 1, 1638
3: 1, 16382: 1, 16381: 1, 16380: 1, 16379: 1, 16378: 1, 16377: 1, 1637
6: 1, 16375: 1, 16374: 1, 16373: 1, 16372: 1, 16371: 1, 16370: 1, 1636
9: 1, 16368: 1, 16367: 1, 16366: 1, 16365: 1, 16364: 1, 16363: 1, 1636
2: 1, 16361: 1, 16360: 1, 16359: 1, 16358: 1, 16357: 1, 16356: 1, 1635
5: 1, 16354: 1, 16353: 1, 16352: 1, 16351: 1, 16350: 1, 16349: 1, 1634

8: 1, 16347: 1, 16346: 1, 16345: 1, 16344: 1, 16343: 1, 16342: 1, 1634
1: 1, 16340: 1, 16339: 1, 16338: 1, 16337: 1, 16336: 1, 16335: 1, 1633
4: 1, 16333: 1, 16332: 1, 16331: 1, 16330: 1, 16329: 1, 16328: 1, 1632
7: 1, 16326: 1, 16325: 1, 16324: 1, 16323: 1, 16322: 1, 16321: 1, 1632
0: 1, 16319: 1, 16318: 1, 16317: 1, 16316: 1, 16315: 1, 16314: 1, 1631
3: 1, 16312: 1, 16311: 1, 16310: 1, 16309: 1, 16308: 1, 16307: 1, 1630
6: 1, 16305: 1, 16304: 1, 16303: 1, 16302: 1, 16301: 1, 16300: 1, 1629
9: 1, 16298: 1, 16297: 1, 16296: 1, 16295: 1, 16294: 1, 16293: 1, 1629
2: 1, 16291: 1, 16290: 1, 16289: 1, 16288: 1, 16287: 1, 16286: 1, 1628
5: 1, 16284: 1, 16283: 1, 16282: 1, 16281: 1, 16280: 1, 16279: 1, 1627
8: 1, 16277: 1, 16276: 1, 16275: 1, 16274: 1, 16273: 1, 16272: 1, 1627
1: 1, 16270: 1, 16269: 1, 16268: 1, 16267: 1, 16266: 1, 16265: 1, 1626
4: 1, 16263: 1, 16262: 1, 16261: 1, 16260: 1, 16259: 1, 16258: 1, 1625
7: 1, 16256: 1, 16255: 1, 16254: 1, 16253: 1, 16252: 1, 16251: 1, 1625
0: 1, 16249: 1, 16248: 1, 16247: 1, 16246: 1, 16245: 1, 16244: 1, 1624
3: 1, 16242: 1, 16241: 1, 16240: 1, 16239: 1, 16238: 1, 16237: 1, 1623
6: 1, 16235: 1, 16234: 1, 16233: 1, 16232: 1, 16231: 1, 16230: 1, 1622
9: 1, 16228: 1, 16227: 1, 16226: 1, 16225: 1, 16224: 1, 16223: 1, 1622
2: 1, 16221: 1, 16220: 1, 16219: 1, 16218: 1, 16217: 1, 16216: 1, 1621
5: 1, 16214: 1, 16213: 1, 16212: 1, 16211: 1, 16210: 1, 16209: 1, 1620
8: 1, 16207: 1, 16206: 1, 16205: 1, 16204: 1, 16203: 1, 16202: 1, 1620
1: 1, 16200: 1, 16199: 1, 16198: 1, 16197: 1, 16196: 1, 16195: 1, 1619
4: 1, 16193: 1, 16192: 1, 16191: 1, 16190: 1, 16189: 1, 16188: 1, 1618
7: 1, 16186: 1, 16185: 1, 16184: 1, 16183: 1, 16182: 1, 16181: 1, 1618
0: 1, 16179: 1, 16178: 1, 16177: 1, 16176: 1, 16175: 1, 16174: 1, 1617
3: 1, 16172: 1, 16171: 1, 16170: 1, 16169: 1, 16168: 1, 16167: 1, 1616
6: 1, 16165: 1, 16164: 1, 16163: 1, 16162: 1, 16161: 1, 16160: 1, 1615
9: 1, 16158: 1, 16157: 1, 16156: 1, 16155: 1, 16154: 1, 16153: 1, 1615
2: 1, 16151: 1, 16150: 1, 16149: 1, 16148: 1, 16147: 1, 16146: 1, 1614
5: 1, 16144: 1, 16143: 1, 16142: 1, 16141: 1, 16140: 1, 16139: 1, 1613
8: 1, 16137: 1, 16136: 1, 16135: 1, 16134: 1, 16133: 1, 16132: 1, 1613
1: 1, 16130: 1, 16129: 1, 16128: 1, 16127: 1, 16126: 1, 16125: 1, 1612
4: 1, 16123: 1, 16122: 1, 16121: 1, 16120: 1, 16119: 1, 16118: 1, 1611
7: 1, 16116: 1, 16115: 1, 16114: 1, 16113: 1, 16112: 1, 16111: 1, 1611
0: 1, 16109: 1, 16108: 1, 16107: 1, 16106: 1, 16105: 1, 16104: 1, 1610
3: 1, 16102: 1, 16101: 1, 16100: 1, 16099: 1, 16098: 1, 16097: 1, 1609
6: 1, 16095: 1, 16094: 1, 16093: 1, 16092: 1, 16091: 1, 16090: 1, 1608
9: 1, 16088: 1, 16087: 1, 16086: 1, 16085: 1, 16084: 1, 16083: 1, 1608
2: 1, 16081: 1, 16080: 1, 16079: 1, 16078: 1, 16077: 1, 16076: 1, 1607

5: 1, 16074: 1, 16073: 1, 16072: 1, 16071: 1, 16070: 1, 16069: 1, 1606
8: 1, 16067: 1, 16066: 1, 16065: 1, 16064: 1, 16063: 1, 16062: 1, 1606
1: 1, 16060: 1, 16059: 1, 16058: 1, 16057: 1, 16056: 1, 16055: 1, 1605
4: 1, 16053: 1, 16052: 1, 16051: 1, 16050: 1, 16049: 1, 16048: 1, 1604
7: 1, 16046: 1, 16045: 1, 16044: 1, 16043: 1, 16042: 1, 16041: 1, 1604
0: 1, 16039: 1, 16038: 1, 16037: 1, 16036: 1, 16035: 1, 16034: 1, 1603
3: 1, 16032: 1, 16031: 1, 16030: 1, 16029: 1, 16028: 1, 16027: 1, 1602
6: 1, 16025: 1, 16024: 1, 16023: 1, 16022: 1, 16021: 1, 16020: 1, 1601
9: 1, 16018: 1, 16017: 1, 16016: 1, 16015: 1, 16014: 1, 16013: 1, 1601
2: 1, 16011: 1, 16010: 1, 16009: 1, 16008: 1, 16007: 1, 16006: 1, 1600
5: 1, 16004: 1, 16003: 1, 16002: 1, 16001: 1, 16000: 1, 15999: 1, 1599
8: 1, 15997: 1, 15996: 1, 15995: 1, 15994: 1, 15993: 1, 15992: 1, 1599
1: 1, 15990: 1, 15989: 1, 15988: 1, 15987: 1, 15986: 1, 15985: 1, 1598
4: 1, 15983: 1, 15982: 1, 15981: 1, 15980: 1, 15979: 1, 15978: 1, 1597
7: 1, 15976: 1, 15975: 1, 15974: 1, 15973: 1, 15972: 1, 15971: 1, 1597
0: 1, 15969: 1, 15968: 1, 15967: 1, 15966: 1, 15965: 1, 15964: 1, 1596
3: 1, 15962: 1, 15961: 1, 15960: 1, 15959: 1, 15958: 1, 15957: 1, 1595
6: 1, 15955: 1, 15954: 1, 15953: 1, 15952: 1, 15951: 1, 15950: 1, 1594
9: 1, 15948: 1, 15947: 1, 15946: 1, 15945: 1, 15944: 1, 15943: 1, 1594
2: 1, 15941: 1, 15940: 1, 15939: 1, 15938: 1, 15937: 1, 15936: 1, 1593
5: 1, 15934: 1, 15933: 1, 15932: 1, 15931: 1, 15930: 1, 15929: 1, 1592
8: 1, 15927: 1, 15926: 1, 15925: 1, 15924: 1, 15923: 1, 15922: 1, 1592
1: 1, 15920: 1, 15919: 1, 15918: 1, 15917: 1, 15916: 1, 15915: 1, 1591
4: 1, 15913: 1, 15912: 1, 15911: 1, 15910: 1, 15909: 1, 15908: 1, 1590
7: 1, 15906: 1, 15905: 1, 15904: 1, 15903: 1, 15902: 1, 15901: 1, 1590
0: 1, 15899: 1, 15898: 1, 15897: 1, 15896: 1, 15895: 1, 15894: 1, 1589
3: 1, 15892: 1, 15891: 1, 15890: 1, 15889: 1, 15888: 1, 15887: 1, 1588
6: 1, 15885: 1, 15884: 1, 15883: 1, 15882: 1, 15881: 1, 15880: 1, 1587
9: 1, 15878: 1, 15877: 1, 15876: 1, 15875: 1, 15874: 1, 15873: 1, 1587
2: 1, 15871: 1, 15870: 1, 15869: 1, 15868: 1, 15867: 1, 15866: 1, 1586
5: 1, 15864: 1, 15863: 1, 15862: 1, 15861: 1, 15860: 1, 15859: 1, 1585
8: 1, 15857: 1, 15856: 1, 15855: 1, 15854: 1, 15853: 1, 15852: 1, 1585
1: 1, 15850: 1, 15849: 1, 15848: 1, 15847: 1, 15846: 1, 15845: 1, 1584
4: 1, 15843: 1, 15842: 1, 15841: 1, 15840: 1, 15839: 1, 15838: 1, 1583
7: 1, 15836: 1, 15835: 1, 15834: 1, 15833: 1, 15832: 1, 15831: 1, 1583
0: 1, 15829: 1, 15828: 1, 15827: 1, 15826: 1, 15825: 1, 15824: 1, 1582
3: 1, 15822: 1, 15821: 1, 15820: 1, 15819: 1, 15818: 1, 15817: 1, 1581
6: 1, 15815: 1, 15814: 1, 15813: 1, 15812: 1, 15811: 1, 15810: 1, 1580
9: 1, 15808: 1, 15807: 1, 15806: 1, 15805: 1, 15804: 1, 15803: 1, 1580

2: 1, 15801: 1, 15800: 1, 15799: 1, 15798: 1, 15797: 1, 15796: 1, 1579
5: 1, 15794: 1, 15793: 1, 15792: 1, 15791: 1, 15790: 1, 15789: 1, 1578
8: 1, 15787: 1, 15786: 1, 15785: 1, 15784: 1, 15783: 1, 15782: 1, 1578
1: 1, 15780: 1, 15779: 1, 15778: 1, 15777: 1, 15776: 1, 15775: 1, 1577
4: 1, 15773: 1, 15772: 1, 15771: 1, 15770: 1, 15769: 1, 15768: 1, 1576
7: 1, 15766: 1, 15765: 1, 15764: 1, 15763: 1, 15762: 1, 15761: 1, 1576
0: 1, 15759: 1, 15758: 1, 15757: 1, 15756: 1, 15755: 1, 15754: 1, 1575
3: 1, 15752: 1, 15751: 1, 15750: 1, 15749: 1, 15748: 1, 15747: 1, 1574
6: 1, 15745: 1, 15744: 1, 15743: 1, 15742: 1, 15741: 1, 15740: 1, 1573
9: 1, 15738: 1, 15737: 1, 15736: 1, 15735: 1, 15734: 1, 15733: 1, 1573
2: 1, 15731: 1, 15730: 1, 15729: 1, 15728: 1, 15727: 1, 15726: 1, 1572
5: 1, 15724: 1, 15723: 1, 15722: 1, 15721: 1, 15720: 1, 15719: 1, 1571
8: 1, 15717: 1, 15716: 1, 15715: 1, 15714: 1, 15713: 1, 15712: 1, 1571
1: 1, 15710: 1, 15709: 1, 15708: 1, 15707: 1, 15706: 1, 15705: 1, 1570
4: 1, 15703: 1, 15702: 1, 15701: 1, 15700: 1, 15699: 1, 15698: 1, 1569
7: 1, 15696: 1, 15695: 1, 15694: 1, 15693: 1, 15692: 1, 15691: 1, 1569
0: 1, 15689: 1, 15688: 1, 15687: 1, 15686: 1, 15685: 1, 15684: 1, 1568
3: 1, 15682: 1, 15681: 1, 15680: 1, 15679: 1, 15678: 1, 15677: 1, 1567
6: 1, 15675: 1, 15674: 1, 15673: 1, 15672: 1, 15671: 1, 15670: 1, 1566
9: 1, 15668: 1, 15667: 1, 15666: 1, 15665: 1, 15664: 1, 15663: 1, 1566
2: 1, 15661: 1, 15660: 1, 15659: 1, 15658: 1, 15657: 1, 15656: 1, 1565
5: 1, 15654: 1, 15653: 1, 15652: 1, 15651: 1, 15650: 1, 15649: 1, 1564
8: 1, 15647: 1, 15646: 1, 15645: 1, 15644: 1, 15643: 1, 15642: 1, 1564
1: 1, 15640: 1, 15639: 1, 15638: 1, 15637: 1, 15636: 1, 15635: 1, 1563
4: 1, 15633: 1, 15632: 1, 15631: 1, 15630: 1, 15629: 1, 15628: 1, 1562
7: 1, 15626: 1, 15625: 1, 15624: 1, 15623: 1, 15622: 1, 15621: 1, 1562
0: 1, 15619: 1, 15618: 1, 15617: 1, 15616: 1, 15615: 1, 15614: 1, 1561
3: 1, 15612: 1, 15611: 1, 15610: 1, 15609: 1, 15608: 1, 15607: 1, 1560
6: 1, 15605: 1, 15604: 1, 15603: 1, 15602: 1, 15601: 1, 15600: 1, 1559
9: 1, 15598: 1, 15597: 1, 15596: 1, 15595: 1, 15594: 1, 15593: 1, 1559
2: 1, 15591: 1, 15590: 1, 15589: 1, 15588: 1, 15587: 1, 15586: 1, 1558
5: 1, 15584: 1, 15583: 1, 15582: 1, 15581: 1, 15580: 1, 15579: 1, 1557
8: 1, 15577: 1, 15576: 1, 15575: 1, 15574: 1, 15573: 1, 15572: 1, 1557
1: 1, 15570: 1, 15569: 1, 15568: 1, 15567: 1, 15566: 1, 15565: 1, 1556
4: 1, 15563: 1, 15562: 1, 15561: 1, 15560: 1, 15559: 1, 15558: 1, 1555
7: 1, 15556: 1, 15555: 1, 15554: 1, 15553: 1, 15552: 1, 15551: 1, 1555
0: 1, 15549: 1, 15548: 1, 15547: 1, 15546: 1, 15545: 1, 15544: 1, 1554
3: 1, 15542: 1, 15541: 1, 15540: 1, 15539: 1, 15538: 1, 15537: 1, 1553
6: 1, 15535: 1, 15534: 1, 15533: 1, 15532: 1, 15531: 1, 15530: 1, 1552

9: 1, 15528: 1, 15527: 1, 15526: 1, 15525: 1, 15524: 1, 15523: 1, 1552
2: 1, 15521: 1, 15520: 1, 15519: 1, 15518: 1, 15517: 1, 15516: 1, 1551
5: 1, 15514: 1, 15513: 1, 15512: 1, 15511: 1, 15510: 1, 15509: 1, 1550
8: 1, 15507: 1, 15506: 1, 15505: 1, 15504: 1, 15503: 1, 15502: 1, 1550
1: 1, 15500: 1, 15499: 1, 15498: 1, 15497: 1, 15496: 1, 15495: 1, 1549
4: 1, 15493: 1, 15492: 1, 15491: 1, 15490: 1, 15489: 1, 15488: 1, 1548
7: 1, 15486: 1, 15485: 1, 15484: 1, 15483: 1, 15482: 1, 15481: 1, 1548
0: 1, 15479: 1, 15478: 1, 15477: 1, 15476: 1, 15475: 1, 15474: 1, 1547
3: 1, 15472: 1, 15471: 1, 15470: 1, 15469: 1, 15468: 1, 15467: 1, 1546
6: 1, 15465: 1, 15464: 1, 15463: 1, 15462: 1, 15461: 1, 15460: 1, 1545
9: 1, 15458: 1, 15457: 1, 15456: 1, 15455: 1, 15454: 1, 15453: 1, 1545
2: 1, 15451: 1, 15450: 1, 15449: 1, 15448: 1, 15447: 1, 15446: 1, 1544
5: 1, 15444: 1, 15443: 1, 15442: 1, 15441: 1, 15440: 1, 15439: 1, 1543
8: 1, 15437: 1, 15436: 1, 15435: 1, 15434: 1, 15433: 1, 15432: 1, 1543
1: 1, 15430: 1, 15429: 1, 15428: 1, 15427: 1, 15426: 1, 15425: 1, 1542
4: 1, 15423: 1, 15422: 1, 15421: 1, 15420: 1, 15419: 1, 15418: 1, 1541
7: 1, 15416: 1, 15415: 1, 15414: 1, 15413: 1, 15412: 1, 15411: 1, 1541
0: 1, 15409: 1, 15408: 1, 15407: 1, 15406: 1, 15405: 1, 15404: 1, 1540
3: 1, 15402: 1, 15401: 1, 15400: 1, 15399: 1, 15398: 1, 15397: 1, 1539
6: 1, 15395: 1, 15394: 1, 15393: 1, 15392: 1, 15391: 1, 15390: 1, 1538
9: 1, 15388: 1, 15387: 1, 15386: 1, 15385: 1, 15384: 1, 15383: 1, 1538
2: 1, 15381: 1, 15380: 1, 15379: 1, 15378: 1, 15377: 1, 15376: 1, 1537
5: 1, 15374: 1, 15373: 1, 15372: 1, 15371: 1, 15370: 1, 15369: 1, 1536
8: 1, 15367: 1, 15366: 1, 15365: 1, 15364: 1, 15363: 1, 15362: 1, 1536
1: 1, 15360: 1, 15359: 1, 15358: 1, 15357: 1, 15356: 1, 15355: 1, 1535
4: 1, 15353: 1, 15352: 1, 15351: 1, 15350: 1, 15349: 1, 15348: 1, 1534
7: 1, 15346: 1, 15345: 1, 15344: 1, 15343: 1, 15342: 1, 15341: 1, 1534
0: 1, 15339: 1, 15338: 1, 15337: 1, 15336: 1, 15335: 1, 15334: 1, 1533
3: 1, 15332: 1, 15331: 1, 15330: 1, 15329: 1, 15328: 1, 15327: 1, 1532
6: 1, 15325: 1, 15324: 1, 15323: 1, 15322: 1, 15321: 1, 15320: 1, 1531
9: 1, 15318: 1, 15317: 1, 15316: 1, 15315: 1, 15314: 1, 15313: 1, 1531
2: 1, 15311: 1, 15310: 1, 15309: 1, 15308: 1, 15307: 1, 15306: 1, 1530
5: 1, 15304: 1, 15303: 1, 15302: 1, 15301: 1, 15300: 1, 15299: 1, 1529
8: 1, 15297: 1, 15296: 1, 15295: 1, 15294: 1, 15293: 1, 15292: 1, 1529
1: 1, 15290: 1, 15289: 1, 15288: 1, 15287: 1, 15286: 1, 15285: 1, 1528
4: 1, 15283: 1, 15282: 1, 15281: 1, 15280: 1, 15279: 1, 15278: 1, 1527
7: 1, 15276: 1, 15275: 1, 15274: 1, 15273: 1, 15272: 1, 15271: 1, 1527
0: 1, 15269: 1, 15268: 1, 15267: 1, 15266: 1, 15265: 1, 15264: 1, 1526
3: 1, 15262: 1, 15261: 1, 15260: 1, 15259: 1, 15258: 1, 15257: 1, 1525

6: 1, 15255: 1, 15254: 1, 15253: 1, 15252: 1, 15251: 1, 15250: 1, 1524
9: 1, 15248: 1, 15247: 1, 15246: 1, 15245: 1, 15244: 1, 15243: 1, 1524
2: 1, 15241: 1, 15240: 1, 15239: 1, 15238: 1, 15237: 1, 15236: 1, 1523
5: 1, 15234: 1, 15233: 1, 15232: 1, 15231: 1, 15230: 1, 15229: 1, 1522
8: 1, 15227: 1, 15226: 1, 15225: 1, 15224: 1, 15223: 1, 15222: 1, 1522
1: 1, 15220: 1, 15219: 1, 15218: 1, 15217: 1, 15216: 1, 15215: 1, 1521
4: 1, 15213: 1, 15212: 1, 15211: 1, 15210: 1, 15209: 1, 15208: 1, 1520
7: 1, 15206: 1, 15205: 1, 15204: 1, 15203: 1, 15202: 1, 15201: 1, 1520
0: 1, 15199: 1, 15198: 1, 15197: 1, 15196: 1, 15195: 1, 15194: 1, 1519
3: 1, 15192: 1, 15191: 1, 15190: 1, 15189: 1, 15188: 1, 15187: 1, 1518
6: 1, 15185: 1, 15184: 1, 15183: 1, 15182: 1, 15181: 1, 15180: 1, 1517
9: 1, 15178: 1, 15177: 1, 15176: 1, 15175: 1, 15174: 1, 15173: 1, 1517
2: 1, 15171: 1, 15170: 1, 15169: 1, 15168: 1, 15167: 1, 15166: 1, 1516
5: 1, 15164: 1, 15163: 1, 15162: 1, 15161: 1, 15160: 1, 15159: 1, 1515
8: 1, 15157: 1, 15156: 1, 15155: 1, 15154: 1, 15153: 1, 15152: 1, 1515
1: 1, 15150: 1, 15149: 1, 15148: 1, 15147: 1, 15146: 1, 15145: 1, 1514
4: 1, 15143: 1, 15142: 1, 15141: 1, 15140: 1, 15139: 1, 15138: 1, 1513
7: 1, 15136: 1, 15135: 1, 15134: 1, 15133: 1, 15132: 1, 15131: 1, 1513
0: 1, 15129: 1, 15128: 1, 15127: 1, 15126: 1, 15125: 1, 15124: 1, 1512
3: 1, 15122: 1, 15121: 1, 15120: 1, 15119: 1, 15118: 1, 15117: 1, 1511
6: 1, 15115: 1, 15114: 1, 15113: 1, 15112: 1, 15111: 1, 15110: 1, 1510
9: 1, 15108: 1, 15107: 1, 15106: 1, 15105: 1, 15104: 1, 15103: 1, 1510
2: 1, 15101: 1, 15100: 1, 15099: 1, 15098: 1, 15097: 1, 15096: 1, 1509
5: 1, 15094: 1, 15093: 1, 15092: 1, 15091: 1, 15090: 1, 15089: 1, 1508
8: 1, 15087: 1, 15086: 1, 15085: 1, 15084: 1, 15083: 1, 15082: 1, 1508
1: 1, 15080: 1, 15079: 1, 15078: 1, 15077: 1, 15076: 1, 15075: 1, 1507
4: 1, 15073: 1, 15072: 1, 15071: 1, 15070: 1, 15069: 1, 15068: 1, 1506
7: 1, 15066: 1, 15065: 1, 15064: 1, 15063: 1, 15062: 1, 15061: 1, 1506
0: 1, 15059: 1, 15058: 1, 15057: 1, 15056: 1, 15055: 1, 15054: 1, 1505
3: 1, 15052: 1, 15051: 1, 15050: 1, 15049: 1, 15048: 1, 15047: 1, 1504
6: 1, 15045: 1, 15044: 1, 15043: 1, 15042: 1, 15041: 1, 15040: 1, 1503
9: 1, 15038: 1, 15037: 1, 15036: 1, 15035: 1, 15034: 1, 15033: 1, 1503
2: 1, 15031: 1, 15030: 1, 15029: 1, 15028: 1, 15027: 1, 15026: 1, 1502
5: 1, 15024: 1, 15023: 1, 15022: 1, 15021: 1, 15020: 1, 15019: 1, 1501
8: 1, 15017: 1, 15016: 1, 15015: 1, 15014: 1, 15013: 1, 15012: 1, 1501
1: 1, 15010: 1, 15009: 1, 15008: 1, 15007: 1, 15006: 1, 15005: 1, 1500
4: 1, 15003: 1, 15002: 1, 15001: 1, 15000: 1, 14999: 1, 14998: 1, 1499
7: 1, 14996: 1, 14995: 1, 14994: 1, 14993: 1, 14992: 1, 14991: 1, 1499
0: 1, 14989: 1, 14988: 1, 14987: 1, 14986: 1, 14985: 1, 14984: 1, 1498

3: 1, 14982: 1, 14981: 1, 14980: 1, 14979: 1, 14978: 1, 14977: 1, 1497
6: 1, 14975: 1, 14974: 1, 14973: 1, 14972: 1, 14971: 1, 14970: 1, 1496
9: 1, 14968: 1, 14967: 1, 14966: 1, 14965: 1, 14964: 1, 14963: 1, 1496
2: 1, 14961: 1, 14960: 1, 14959: 1, 14958: 1, 14957: 1, 14956: 1, 1495
5: 1, 14954: 1, 14953: 1, 14952: 1, 14951: 1, 14950: 1, 14949: 1, 1494
8: 1, 14947: 1, 14946: 1, 14945: 1, 14944: 1, 14943: 1, 14942: 1, 1494
1: 1, 14940: 1, 14939: 1, 14938: 1, 14937: 1, 14936: 1, 14935: 1, 1493
4: 1, 14933: 1, 14932: 1, 14931: 1, 14930: 1, 14929: 1, 14928: 1, 1492
7: 1, 14926: 1, 14925: 1, 14924: 1, 14923: 1, 14922: 1, 14921: 1, 1492
0: 1, 14919: 1, 14918: 1, 14917: 1, 14916: 1, 14915: 1, 14914: 1, 1491
3: 1, 14912: 1, 14911: 1, 14910: 1, 14909: 1, 14908: 1, 14907: 1, 1490
6: 1, 14905: 1, 14904: 1, 14903: 1, 14902: 1, 14901: 1, 14900: 1, 1489
9: 1, 14898: 1, 14897: 1, 14896: 1, 14895: 1, 14894: 1, 14893: 1, 1489
2: 1, 14891: 1, 14890: 1, 14889: 1, 14888: 1, 14887: 1, 14886: 1, 1488
5: 1, 14884: 1, 14883: 1, 14882: 1, 14881: 1, 14880: 1, 14879: 1, 1487
8: 1, 14877: 1, 14876: 1, 14875: 1, 14874: 1, 14873: 1, 14872: 1, 1487
1: 1, 14870: 1, 14869: 1, 14868: 1, 14867: 1, 14866: 1, 14865: 1, 1486
4: 1, 14863: 1, 14862: 1, 14861: 1, 14860: 1, 14859: 1, 14858: 1, 1485
7: 1, 14856: 1, 14855: 1, 14854: 1, 14853: 1, 14852: 1, 14851: 1, 1485
0: 1, 14849: 1, 14848: 1, 14847: 1, 14846: 1, 14845: 1, 14844: 1, 1484
3: 1, 14842: 1, 14841: 1, 14840: 1, 14839: 1, 14838: 1, 14837: 1, 1483
6: 1, 14835: 1, 14834: 1, 14833: 1, 14832: 1, 14831: 1, 14830: 1, 1482
9: 1, 14828: 1, 14827: 1, 14826: 1, 14825: 1, 14824: 1, 14823: 1, 1482
2: 1, 14821: 1, 14820: 1, 14819: 1, 14818: 1, 14817: 1, 14816: 1, 1481
5: 1, 14814: 1, 14813: 1, 14812: 1, 14811: 1, 14810: 1, 14809: 1, 1480
8: 1, 14807: 1, 14806: 1, 14805: 1, 14804: 1, 14803: 1, 14802: 1, 1480
1: 1, 14800: 1, 14799: 1, 14798: 1, 14797: 1, 14796: 1, 14795: 1, 1479
4: 1, 14793: 1, 14792: 1, 14791: 1, 14790: 1, 14789: 1, 14788: 1, 1478
7: 1, 14786: 1, 14785: 1, 14784: 1, 14783: 1, 14782: 1, 14781: 1, 1478
0: 1, 14779: 1, 14778: 1, 14777: 1, 14776: 1, 14775: 1, 14774: 1, 1477
3: 1, 14772: 1, 14771: 1, 14770: 1, 14769: 1, 14768: 1, 14767: 1, 1476
6: 1, 14765: 1, 14764: 1, 14763: 1, 14762: 1, 14761: 1, 14760: 1, 1475
9: 1, 14758: 1, 14757: 1, 14756: 1, 14755: 1, 14754: 1, 14753: 1, 1475
2: 1, 14751: 1, 14750: 1, 14749: 1, 14748: 1, 14747: 1, 14746: 1, 1474
5: 1, 14744: 1, 14743: 1, 14742: 1, 14741: 1, 14740: 1, 14739: 1, 1473
8: 1, 14737: 1, 14736: 1, 14735: 1, 14734: 1, 14733: 1, 14732: 1, 1473
1: 1, 14730: 1, 14729: 1, 14728: 1, 14727: 1, 14726: 1, 14725: 1, 1472
4: 1, 14723: 1, 14722: 1, 14721: 1, 14720: 1, 14719: 1, 14718: 1, 1471
7: 1, 14716: 1, 14715: 1, 14714: 1, 14713: 1, 14712: 1, 14711: 1, 1471

0: 1, 14709: 1, 14708: 1, 14707: 1, 14706: 1, 14705: 1, 14704: 1, 1470
3: 1, 14702: 1, 14701: 1, 14700: 1, 14699: 1, 14698: 1, 14697: 1, 1469
6: 1, 14695: 1, 14694: 1, 14693: 1, 14692: 1, 14691: 1, 14690: 1, 1468
9: 1, 14688: 1, 14687: 1, 14686: 1, 14685: 1, 14684: 1, 14683: 1, 1468
2: 1, 14681: 1, 14680: 1, 14679: 1, 14678: 1, 14677: 1, 14676: 1, 1467
5: 1, 14674: 1, 14673: 1, 14672: 1, 14671: 1, 14670: 1, 14669: 1, 1466
8: 1, 14667: 1, 14666: 1, 14665: 1, 14664: 1, 14663: 1, 14662: 1, 1466
1: 1, 14660: 1, 14659: 1, 14658: 1, 14657: 1, 14656: 1, 14655: 1, 1465
4: 1, 14653: 1, 14652: 1, 14651: 1, 14650: 1, 14649: 1, 14648: 1, 1464
7: 1, 14646: 1, 14645: 1, 14644: 1, 14643: 1, 14642: 1, 14641: 1, 1464
0: 1, 14639: 1, 14638: 1, 14637: 1, 14636: 1, 14635: 1, 14634: 1, 1463
3: 1, 14632: 1, 14631: 1, 14630: 1, 14629: 1, 14628: 1, 14627: 1, 1462
6: 1, 14625: 1, 14624: 1, 14623: 1, 14622: 1, 14621: 1, 14620: 1, 1461
9: 1, 14618: 1, 14617: 1, 14616: 1, 14615: 1, 14614: 1, 14613: 1, 1461
2: 1, 14611: 1, 14610: 1, 14609: 1, 14608: 1, 14607: 1, 14606: 1, 1460
5: 1, 14604: 1, 14603: 1, 14602: 1, 14601: 1, 14600: 1, 14599: 1, 1459
8: 1, 14597: 1, 14596: 1, 14595: 1, 14594: 1, 14593: 1, 14592: 1, 1459
1: 1, 14590: 1, 14589: 1, 14588: 1, 14587: 1, 14586: 1, 14585: 1, 1458
4: 1, 14583: 1, 14582: 1, 14581: 1, 14580: 1, 14579: 1, 14578: 1, 1457
7: 1, 14576: 1, 14575: 1, 14574: 1, 14573: 1, 14572: 1, 14571: 1, 1457
0: 1, 14569: 1, 14568: 1, 14567: 1, 14566: 1, 14565: 1, 14564: 1, 1456
3: 1, 14562: 1, 14561: 1, 14560: 1, 14559: 1, 14558: 1, 14557: 1, 1455
6: 1, 14555: 1, 14554: 1, 14553: 1, 14552: 1, 14551: 1, 14550: 1, 1454
9: 1, 14548: 1, 14547: 1, 14546: 1, 14545: 1, 14544: 1, 14543: 1, 1454
2: 1, 14541: 1, 14540: 1, 14539: 1, 14538: 1, 14537: 1, 14536: 1, 1453
5: 1, 14534: 1, 14533: 1, 14532: 1, 14531: 1, 14530: 1, 14529: 1, 1452
8: 1, 14527: 1, 14526: 1, 14525: 1, 14524: 1, 14523: 1, 14522: 1, 1452
1: 1, 14520: 1, 14519: 1, 14518: 1, 14517: 1, 14516: 1, 14515: 1, 1451
4: 1, 14513: 1, 14512: 1, 14511: 1, 14510: 1, 14509: 1, 14508: 1, 1450
7: 1, 14506: 1, 14505: 1, 14504: 1, 14503: 1, 14502: 1, 14501: 1, 1450
0: 1, 14499: 1, 14498: 1, 14497: 1, 14496: 1, 14495: 1, 14494: 1, 1449
3: 1, 14492: 1, 14491: 1, 14490: 1, 14489: 1, 14488: 1, 14487: 1, 1448
6: 1, 14485: 1, 14484: 1, 14483: 1, 14482: 1, 14481: 1, 14480: 1, 1447
9: 1, 14478: 1, 14477: 1, 14476: 1, 14475: 1, 14474: 1, 14473: 1, 1447
2: 1, 14471: 1, 14470: 1, 14469: 1, 14468: 1, 14467: 1, 14466: 1, 1446
5: 1, 14464: 1, 14463: 1, 14462: 1, 14461: 1, 14460: 1, 14459: 1, 1445
8: 1, 14457: 1, 14456: 1, 14455: 1, 14454: 1, 14453: 1, 14452: 1, 1445
1: 1, 14450: 1, 14449: 1, 14448: 1, 14447: 1, 14446: 1, 14445: 1, 1444
4: 1, 14443: 1, 14442: 1, 14441: 1, 14440: 1, 14439: 1, 14438: 1, 1443

4: 1, 14163: 1, 14162: 1, 14161: 1, 14160: 1, 14159: 1, 14158: 1, 1415
7: 1, 14156: 1, 14155: 1, 14154: 1, 14153: 1, 14152: 1, 14151: 1, 1415
0: 1, 14149: 1, 14148: 1, 14147: 1, 14146: 1, 14145: 1, 14144: 1, 1414
3: 1, 14142: 1, 14141: 1, 14140: 1, 14139: 1, 14138: 1, 14137: 1, 1413
6: 1, 14135: 1, 14134: 1, 14133: 1, 14132: 1, 14131: 1, 14130: 1, 1412
9: 1, 14128: 1, 14127: 1, 14126: 1, 14125: 1, 14124: 1, 14123: 1, 1412
2: 1, 14121: 1, 14120: 1, 14119: 1, 14118: 1, 14117: 1, 14116: 1, 1411
5: 1, 14114: 1, 14113: 1, 14112: 1, 14111: 1, 14110: 1, 14109: 1, 1410
8: 1, 14107: 1, 14106: 1, 14105: 1, 14104: 1, 14103: 1, 14102: 1, 1410
1: 1, 14100: 1, 14099: 1, 14098: 1, 14097: 1, 14096: 1, 14095: 1, 1409
4: 1, 14093: 1, 14092: 1, 14091: 1, 14090: 1, 14089: 1, 14088: 1, 1408
7: 1, 14086: 1, 14085: 1, 14084: 1, 14083: 1, 14082: 1, 14081: 1, 1408
0: 1, 14079: 1, 14078: 1, 14077: 1, 14076: 1, 14075: 1, 14074: 1, 1407
3: 1, 14072: 1, 14071: 1, 14070: 1, 14069: 1, 14068: 1, 14067: 1, 1406
6: 1, 14065: 1, 14064: 1, 14063: 1, 14062: 1, 14061: 1, 14060: 1, 1405
9: 1, 14058: 1, 14057: 1, 14056: 1, 14055: 1, 14054: 1, 14053: 1, 1405
2: 1, 14051: 1, 14050: 1, 14049: 1, 14048: 1, 14047: 1, 14046: 1, 1404
5: 1, 14044: 1, 14043: 1, 14042: 1, 14041: 1, 14040: 1, 14039: 1, 1403
8: 1, 14037: 1, 14036: 1, 14035: 1, 14034: 1, 14033: 1, 14032: 1, 1403
1: 1, 14030: 1, 14029: 1, 14028: 1, 14027: 1, 14026: 1, 14025: 1, 1402
4: 1, 14023: 1, 14022: 1, 14021: 1, 14020: 1, 14019: 1, 14018: 1, 1401
7: 1, 14016: 1, 14015: 1, 14014: 1, 14013: 1, 14012: 1, 14011: 1, 1401
0: 1, 14009: 1, 14008: 1, 14007: 1, 14006: 1, 14005: 1, 14004: 1, 1400
3: 1, 14002: 1, 14001: 1, 14000: 1, 13999: 1, 13998: 1, 13997: 1, 1399
6: 1, 13995: 1, 13994: 1, 13993: 1, 13992: 1, 13991: 1, 13990: 1, 1398
9: 1, 13988: 1, 13987: 1, 13986: 1, 13985: 1, 13984: 1, 13983: 1, 1398
2: 1, 13981: 1, 13980: 1, 13979: 1, 13978: 1, 13977: 1, 13976: 1, 1397
5: 1, 13974: 1, 13973: 1, 13972: 1, 13971: 1, 13970: 1, 13969: 1, 1396
8: 1, 13967: 1, 13966: 1, 13965: 1, 13964: 1, 13963: 1, 13962: 1, 1396
1: 1, 13960: 1, 13959: 1, 13958: 1, 13957: 1, 13956: 1, 13955: 1, 1395
4: 1, 13953: 1, 13952: 1, 13951: 1, 13950: 1, 13949: 1, 13948: 1, 1394
7: 1, 13946: 1, 13945: 1, 13944: 1, 13943: 1, 13942: 1, 13941: 1, 1394
0: 1, 13939: 1, 13938: 1, 13937: 1, 13936: 1, 13935: 1, 13934: 1, 1393
3: 1, 13932: 1, 13931: 1, 13930: 1, 13929: 1, 13928: 1, 13927: 1, 1392
6: 1, 13925: 1, 13924: 1, 13923: 1, 13922: 1, 13921: 1, 13920: 1, 1391
9: 1, 13918: 1, 13917: 1, 13916: 1, 13915: 1, 13914: 1, 13913: 1, 1391
2: 1, 13911: 1, 13910: 1, 13909: 1, 13908: 1, 13907: 1, 13906: 1, 1390
5: 1, 13904: 1, 13903: 1, 13902: 1, 13901: 1, 13900: 1, 13899: 1, 1389
8: 1, 13897: 1, 13896: 1, 13895: 1, 13894: 1, 13893: 1, 13892: 1, 1389

1: 1, 13890: 1, 13889: 1, 13888: 1, 13887: 1, 13886: 1, 13885: 1, 1388
4: 1, 13883: 1, 13882: 1, 13881: 1, 13880: 1, 13879: 1, 13878: 1, 1387
7: 1, 13876: 1, 13875: 1, 13874: 1, 13873: 1, 13872: 1, 13871: 1, 1387
0: 1, 13869: 1, 13868: 1, 13867: 1, 13866: 1, 13865: 1, 13864: 1, 1386
3: 1, 13862: 1, 13861: 1, 13860: 1, 13859: 1, 13858: 1, 13857: 1, 1385
6: 1, 13855: 1, 13854: 1, 13853: 1, 13852: 1, 13851: 1, 13850: 1, 1384
9: 1, 13848: 1, 13847: 1, 13846: 1, 13845: 1, 13844: 1, 13843: 1, 1384
2: 1, 13841: 1, 13840: 1, 13839: 1, 13838: 1, 13837: 1, 13836: 1, 1383
5: 1, 13834: 1, 13833: 1, 13832: 1, 13831: 1, 13830: 1, 13829: 1, 1382
8: 1, 13827: 1, 13826: 1, 13825: 1, 13824: 1, 13823: 1, 13822: 1, 1382
1: 1, 13820: 1, 13819: 1, 13818: 1, 13817: 1, 13816: 1, 13815: 1, 1381
4: 1, 13813: 1, 13812: 1, 13811: 1, 13810: 1, 13809: 1, 13808: 1, 1380
7: 1, 13806: 1, 13805: 1, 13804: 1, 13803: 1, 13802: 1, 13801: 1, 1380
0: 1, 13799: 1, 13798: 1, 13797: 1, 13796: 1, 13795: 1, 13794: 1, 1379
3: 1, 13792: 1, 13791: 1, 13790: 1, 13789: 1, 13788: 1, 13787: 1, 1378
6: 1, 13785: 1, 13784: 1, 13783: 1, 13782: 1, 13781: 1, 13780: 1, 1377
9: 1, 13778: 1, 13777: 1, 13776: 1, 13775: 1, 13774: 1, 13773: 1, 1377
2: 1, 13771: 1, 13770: 1, 13769: 1, 13768: 1, 13767: 1, 13766: 1, 1376
5: 1, 13764: 1, 13763: 1, 13762: 1, 13761: 1, 13760: 1, 13759: 1, 1375
8: 1, 13757: 1, 13756: 1, 13755: 1, 13754: 1, 13753: 1, 13752: 1, 1375
1: 1, 13750: 1, 13749: 1, 13748: 1, 13747: 1, 13746: 1, 13745: 1, 1374
4: 1, 13743: 1, 13742: 1, 13741: 1, 13740: 1, 13739: 1, 13738: 1, 1373
7: 1, 13736: 1, 13735: 1, 13734: 1, 13733: 1, 13732: 1, 13731: 1, 1373
0: 1, 13729: 1, 13728: 1, 13727: 1, 13726: 1, 13725: 1, 13724: 1, 1372
3: 1, 13722: 1, 13721: 1, 13720: 1, 13719: 1, 13718: 1, 13717: 1, 1371
6: 1, 13715: 1, 13714: 1, 13713: 1, 13712: 1, 13711: 1, 13710: 1, 1370
9: 1, 13708: 1, 13707: 1, 13706: 1, 13705: 1, 13704: 1, 13703: 1, 1370
2: 1, 13701: 1, 13700: 1, 13699: 1, 13698: 1, 13697: 1, 13696: 1, 1369
5: 1, 13694: 1, 13693: 1, 13692: 1, 13691: 1, 13690: 1, 13689: 1, 1368
8: 1, 13687: 1, 13686: 1, 13685: 1, 13684: 1, 13683: 1, 13682: 1, 1368
1: 1, 13680: 1, 13679: 1, 13678: 1, 13677: 1, 13676: 1, 13675: 1, 1367
4: 1, 13673: 1, 13672: 1, 13671: 1, 13670: 1, 13669: 1, 13668: 1, 1366
7: 1, 13666: 1, 13665: 1, 13664: 1, 13663: 1, 13662: 1, 13661: 1, 1366
0: 1, 13659: 1, 13658: 1, 13657: 1, 13656: 1, 13655: 1, 13654: 1, 1365
3: 1, 13652: 1, 13651: 1, 13650: 1, 13649: 1, 13648: 1, 13647: 1, 1364
6: 1, 13645: 1, 13644: 1, 13643: 1, 13642: 1, 13641: 1, 13640: 1, 1363
9: 1, 13638: 1, 13637: 1, 13636: 1, 13635: 1, 13634: 1, 13633: 1, 1363
2: 1, 13631: 1, 13630: 1, 13629: 1, 13628: 1, 13627: 1, 13626: 1, 1362
5: 1, 13624: 1, 13623: 1, 13622: 1, 13621: 1, 13620: 1, 13619: 1, 1361

8: 1, 13617: 1, 13616: 1, 13615: 1, 13614: 1, 13613: 1, 13612: 1, 1361
1: 1, 13610: 1, 13609: 1, 13608: 1, 13607: 1, 13606: 1, 13605: 1, 1360
4: 1, 13603: 1, 13602: 1, 13601: 1, 13600: 1, 13599: 1, 13598: 1, 1359
7: 1, 13596: 1, 13595: 1, 13594: 1, 13593: 1, 13592: 1, 13591: 1, 1359
0: 1, 13589: 1, 13588: 1, 13587: 1, 13586: 1, 13585: 1, 13584: 1, 1358
3: 1, 13582: 1, 13581: 1, 13580: 1, 13579: 1, 13578: 1, 13577: 1, 1357
6: 1, 13575: 1, 13574: 1, 13573: 1, 13572: 1, 13571: 1, 13570: 1, 1356
9: 1, 13568: 1, 13567: 1, 13566: 1, 13565: 1, 13564: 1, 13563: 1, 1356
2: 1, 13561: 1, 13560: 1, 13559: 1, 13558: 1, 13557: 1, 13556: 1, 1355
5: 1, 13554: 1, 13553: 1, 13552: 1, 13551: 1, 13550: 1, 13549: 1, 1354
8: 1, 13547: 1, 13546: 1, 13545: 1, 13544: 1, 13543: 1, 13542: 1, 1354
1: 1, 13540: 1, 13539: 1, 13538: 1, 13537: 1, 13536: 1, 13535: 1, 1353
4: 1, 13533: 1, 13532: 1, 13531: 1, 13530: 1, 13529: 1, 13528: 1, 1352
7: 1, 13526: 1, 13525: 1, 13524: 1, 13523: 1, 13522: 1, 13521: 1, 1352
0: 1, 13519: 1, 13518: 1, 13517: 1, 13516: 1, 13515: 1, 13514: 1, 1351
3: 1, 13512: 1, 13511: 1, 13510: 1, 13509: 1, 13508: 1, 13507: 1, 1350
6: 1, 13505: 1, 13504: 1, 13503: 1, 13502: 1, 13501: 1, 13500: 1, 1349
9: 1, 13498: 1, 13497: 1, 13496: 1, 13495: 1, 13494: 1, 13493: 1, 1349
2: 1, 13491: 1, 13490: 1, 13489: 1, 13488: 1, 13487: 1, 13486: 1, 1348
5: 1, 13484: 1, 13483: 1, 13482: 1, 13481: 1, 13480: 1, 13479: 1, 1347
8: 1, 13477: 1, 13476: 1, 13475: 1, 13474: 1, 13473: 1, 13472: 1, 1347
1: 1, 13470: 1, 13469: 1, 13468: 1, 13467: 1, 13466: 1, 13465: 1, 1346
4: 1, 13463: 1, 13462: 1, 13461: 1, 13460: 1, 13459: 1, 13458: 1, 1345
7: 1, 13456: 1, 13455: 1, 13454: 1, 13453: 1, 13452: 1, 13451: 1, 1345
0: 1, 13449: 1, 13448: 1, 13447: 1, 13446: 1, 13445: 1, 13444: 1, 1344
3: 1, 13442: 1, 13441: 1, 13440: 1, 13439: 1, 13438: 1, 13437: 1, 1343
6: 1, 13435: 1, 13434: 1, 13433: 1, 13432: 1, 13431: 1, 13430: 1, 1342
9: 1, 13428: 1, 13427: 1, 13426: 1, 13425: 1, 13424: 1, 13423: 1, 1342
2: 1, 13421: 1, 13420: 1, 13419: 1, 13418: 1, 13417: 1, 13416: 1, 1341
5: 1, 13414: 1, 13413: 1, 13412: 1, 13411: 1, 13410: 1, 13409: 1, 1340
8: 1, 13407: 1, 13406: 1, 13405: 1, 13404: 1, 13403: 1, 13402: 1, 1340
1: 1, 13400: 1, 13399: 1, 13398: 1, 13397: 1, 13396: 1, 13395: 1, 1339
4: 1, 13393: 1, 13392: 1, 13391: 1, 13390: 1, 13389: 1, 13388: 1, 1338
7: 1, 13386: 1, 13385: 1, 13384: 1, 13383: 1, 13382: 1, 13381: 1, 1338
0: 1, 13379: 1, 13378: 1, 13377: 1, 13376: 1, 13375: 1, 13374: 1, 1337
3: 1, 13372: 1, 13371: 1, 13370: 1, 13369: 1, 13368: 1, 13367: 1, 1336
6: 1, 13365: 1, 13364: 1, 13363: 1, 13362: 1, 13361: 1, 13360: 1, 1335
9: 1, 13358: 1, 13357: 1, 13356: 1, 13355: 1, 13354: 1, 13353: 1, 1335
2: 1, 13351: 1, 13350: 1, 13349: 1, 13348: 1, 13347: 1, 13346: 1, 1334

5: 1, 13344: 1, 13343: 1, 13342: 1, 13341: 1, 13340: 1, 13339: 1, 1333
8: 1, 13337: 1, 13336: 1, 13335: 1, 13334: 1, 13333: 1, 13332: 1, 1333
1: 1, 13330: 1, 13329: 1, 13328: 1, 13327: 1, 13326: 1, 13325: 1, 1332
4: 1, 13323: 1, 13322: 1, 13321: 1, 13320: 1, 13319: 1, 13318: 1, 1331
7: 1, 13316: 1, 13315: 1, 13314: 1, 13313: 1, 13312: 1, 13311: 1, 1331
0: 1, 13309: 1, 13308: 1, 13307: 1, 13306: 1, 13305: 1, 13304: 1, 1330
3: 1, 13302: 1, 13301: 1, 13300: 1, 13299: 1, 13298: 1, 13297: 1, 1329
6: 1, 13295: 1, 13294: 1, 13293: 1, 13292: 1, 13291: 1, 13290: 1, 1328
9: 1, 13288: 1, 13287: 1, 13286: 1, 13285: 1, 13284: 1, 13283: 1, 1328
2: 1, 13281: 1, 13280: 1, 13279: 1, 13278: 1, 13277: 1, 13276: 1, 1327
5: 1, 13274: 1, 13273: 1, 13272: 1, 13271: 1, 13270: 1, 13269: 1, 1326
8: 1, 13267: 1, 13266: 1, 13265: 1, 13264: 1, 13263: 1, 13262: 1, 1326
1: 1, 13260: 1, 13259: 1, 13258: 1, 13257: 1, 13256: 1, 13255: 1, 1325
4: 1, 13253: 1, 13252: 1, 13251: 1, 13250: 1, 13249: 1, 13248: 1, 1324
7: 1, 13246: 1, 13245: 1, 13244: 1, 13243: 1, 13242: 1, 13241: 1, 1324
0: 1, 13239: 1, 13238: 1, 13237: 1, 13236: 1, 13235: 1, 13234: 1, 1323
3: 1, 13232: 1, 13231: 1, 13230: 1, 13229: 1, 13228: 1, 13227: 1, 1322
6: 1, 13225: 1, 13224: 1, 13223: 1, 13222: 1, 13221: 1, 13220: 1, 1321
9: 1, 13218: 1, 13217: 1, 13216: 1, 13215: 1, 13214: 1, 13213: 1, 1321
2: 1, 13211: 1, 13210: 1, 13209: 1, 13208: 1, 13207: 1, 13206: 1, 1320
5: 1, 13204: 1, 13203: 1, 13202: 1, 13201: 1, 13200: 1, 13199: 1, 1319
8: 1, 13197: 1, 13196: 1, 13195: 1, 13194: 1, 13193: 1, 13192: 1, 1319
1: 1, 13190: 1, 13189: 1, 13188: 1, 13187: 1, 13186: 1, 13185: 1, 1318
4: 1, 13183: 1, 13182: 1, 13181: 1, 13180: 1, 13179: 1, 13178: 1, 1317
7: 1, 13176: 1, 13175: 1, 13174: 1, 13173: 1, 13172: 1, 13171: 1, 1317
0: 1, 13169: 1, 13168: 1, 13167: 1, 13166: 1, 13165: 1, 13164: 1, 1316
3: 1, 13162: 1, 13161: 1, 13160: 1, 13159: 1, 13158: 1, 13157: 1, 1315
6: 1, 13155: 1, 13154: 1, 13153: 1, 13152: 1, 13151: 1, 13150: 1, 1314
9: 1, 13148: 1, 13147: 1, 13146: 1, 13145: 1, 13144: 1, 13143: 1, 1314
2: 1, 13141: 1, 13140: 1, 13139: 1, 13138: 1, 13137: 1, 13136: 1, 1313
5: 1, 13134: 1, 13133: 1, 13132: 1, 13131: 1, 13130: 1, 13129: 1, 1312
8: 1, 13127: 1, 13126: 1, 13125: 1, 13124: 1, 13123: 1, 13122: 1, 1312
1: 1, 13120: 1, 13119: 1, 13118: 1, 13117: 1, 13116: 1, 13115: 1, 1311
4: 1, 13113: 1, 13112: 1, 13111: 1, 13110: 1, 13109: 1, 13108: 1, 1310
7: 1, 13106: 1, 13105: 1, 13104: 1, 13103: 1, 13102: 1, 13101: 1, 1310
0: 1, 13099: 1, 13098: 1, 13097: 1, 13096: 1, 13095: 1, 13094: 1, 1309
3: 1, 13092: 1, 13091: 1, 13090: 1, 13089: 1, 13088: 1, 13087: 1, 1308
6: 1, 13085: 1, 13084: 1, 13083: 1, 13082: 1, 13081: 1, 13080: 1, 1307
9: 1, 13078: 1, 13077: 1, 13076: 1, 13075: 1, 13074: 1, 13073: 1, 1307

2: 1, 13071: 1, 13070: 1, 13069: 1, 13068: 1, 13067: 1, 13066: 1, 1306
5: 1, 13064: 1, 13063: 1, 13062: 1, 13061: 1, 13060: 1, 13059: 1, 1305
8: 1, 13057: 1, 13056: 1, 13055: 1, 13054: 1, 13053: 1, 13052: 1, 1305
1: 1, 13050: 1, 13049: 1, 13048: 1, 13047: 1, 13046: 1, 13045: 1, 1304
4: 1, 13043: 1, 13042: 1, 13041: 1, 13040: 1, 13039: 1, 13038: 1, 1303
7: 1, 13036: 1, 13035: 1, 13034: 1, 13033: 1, 13032: 1, 13031: 1, 1303
0: 1, 13029: 1, 13028: 1, 13027: 1, 13026: 1, 13025: 1, 13024: 1, 1302
3: 1, 13022: 1, 13021: 1, 13020: 1, 13019: 1, 13018: 1, 13017: 1, 1301
6: 1, 13015: 1, 13014: 1, 13013: 1, 13012: 1, 13011: 1, 13010: 1, 1300
9: 1, 13008: 1, 13007: 1, 13006: 1, 13005: 1, 13004: 1, 13003: 1, 1300
2: 1, 13001: 1, 13000: 1, 12999: 1, 12998: 1, 12997: 1, 12996: 1, 1299
5: 1, 12994: 1, 12993: 1, 12992: 1, 12991: 1, 12990: 1, 12989: 1, 1298
8: 1, 12987: 1, 12986: 1, 12985: 1, 12984: 1, 12983: 1, 12982: 1, 1298
1: 1, 12980: 1, 12979: 1, 12978: 1, 12977: 1, 12976: 1, 12975: 1, 1297
4: 1, 12973: 1, 12972: 1, 12971: 1, 12970: 1, 12969: 1, 12968: 1, 1296
7: 1, 12966: 1, 12965: 1, 12964: 1, 12963: 1, 12962: 1, 12961: 1, 1296
0: 1, 12959: 1, 12958: 1, 12957: 1, 12956: 1, 12955: 1, 12954: 1, 1295
3: 1, 12952: 1, 12951: 1, 12950: 1, 12949: 1, 12948: 1, 12947: 1, 1294
6: 1, 12945: 1, 12944: 1, 12943: 1, 12942: 1, 12941: 1, 12940: 1, 1293
9: 1, 12938: 1, 12937: 1, 12936: 1, 12935: 1, 12934: 1, 12933: 1, 1293
2: 1, 12931: 1, 12930: 1, 12929: 1, 12928: 1, 12927: 1, 12926: 1, 1292
5: 1, 12924: 1, 12923: 1, 12922: 1, 12921: 1, 12920: 1, 12919: 1, 1291
8: 1, 12917: 1, 12916: 1, 12915: 1, 12914: 1, 12913: 1, 12912: 1, 1291
1: 1, 12910: 1, 12909: 1, 12908: 1, 12907: 1, 12906: 1, 12905: 1, 1290
4: 1, 12903: 1, 12902: 1, 12901: 1, 12900: 1, 12899: 1, 12898: 1, 1289
7: 1, 12896: 1, 12895: 1, 12894: 1, 12893: 1, 12892: 1, 12891: 1, 1289
0: 1, 12889: 1, 12888: 1, 12887: 1, 12886: 1, 12885: 1, 12884: 1, 1288
3: 1, 12882: 1, 12881: 1, 12880: 1, 12879: 1, 12878: 1, 12877: 1, 1287
6: 1, 12875: 1, 12874: 1, 12873: 1, 12872: 1, 12871: 1, 12870: 1, 1286
9: 1, 12868: 1, 12867: 1, 12866: 1, 12865: 1, 12864: 1, 12863: 1, 1286
2: 1, 12861: 1, 12860: 1, 12859: 1, 12858: 1, 12857: 1, 12856: 1, 1285
5: 1, 12854: 1, 12853: 1, 12852: 1, 12851: 1, 12850: 1, 12849: 1, 1284
8: 1, 12847: 1, 12846: 1, 12845: 1, 12844: 1, 12843: 1, 12842: 1, 1284
1: 1, 12840: 1, 12839: 1, 12838: 1, 12837: 1, 12836: 1, 12835: 1, 1283
4: 1, 12833: 1, 12832: 1, 12831: 1, 12830: 1, 12829: 1, 12828: 1, 1282
7: 1, 12826: 1, 12825: 1, 12824: 1, 12823: 1, 12822: 1, 12821: 1, 1282
0: 1, 12819: 1, 12818: 1, 12817: 1, 12816: 1, 12815: 1, 12814: 1, 1281
3: 1, 12812: 1, 12811: 1, 12810: 1, 12809: 1, 12808: 1, 12807: 1, 1280
6: 1, 12805: 1, 12804: 1, 12803: 1, 12802: 1, 12801: 1, 12800: 1, 1279

9: 1, 12798: 1, 12797: 1, 12796: 1, 12795: 1, 12794: 1, 12793: 1, 1279
2: 1, 12791: 1, 12790: 1, 12789: 1, 12788: 1, 12787: 1, 12786: 1, 1278
5: 1, 12784: 1, 12783: 1, 12782: 1, 12781: 1, 12780: 1, 12779: 1, 1277
8: 1, 12777: 1, 12776: 1, 12775: 1, 12774: 1, 12773: 1, 12772: 1, 1277
1: 1, 12770: 1, 12769: 1, 12768: 1, 12767: 1, 12766: 1, 12765: 1, 1276
4: 1, 12763: 1, 12762: 1, 12761: 1, 12760: 1, 12759: 1, 12758: 1, 1275
7: 1, 12756: 1, 12755: 1, 12754: 1, 12753: 1, 12752: 1, 12751: 1, 1275
0: 1, 12749: 1, 12748: 1, 12747: 1, 12746: 1, 12745: 1, 12744: 1, 1274
3: 1, 12742: 1, 12741: 1, 12740: 1, 12739: 1, 12738: 1, 12737: 1, 1273
6: 1, 12735: 1, 12734: 1, 12733: 1, 12732: 1, 12731: 1, 12730: 1, 1272
9: 1, 12728: 1, 12727: 1, 12726: 1, 12725: 1, 12724: 1, 12723: 1, 1272
2: 1, 12721: 1, 12720: 1, 12719: 1, 12718: 1, 12717: 1, 12716: 1, 1271
5: 1, 12714: 1, 12713: 1, 12712: 1, 12711: 1, 12710: 1, 12709: 1, 1270
8: 1, 12707: 1, 12706: 1, 12705: 1, 12704: 1, 12703: 1, 12702: 1, 1270
1: 1, 12700: 1, 12699: 1, 12698: 1, 12697: 1, 12696: 1, 12695: 1, 1269
4: 1, 12693: 1, 12692: 1, 12691: 1, 12690: 1, 12689: 1, 12688: 1, 1268
7: 1, 12686: 1, 12685: 1, 12684: 1, 12683: 1, 12682: 1, 12681: 1, 1268
0: 1, 12679: 1, 12678: 1, 12677: 1, 12676: 1, 12675: 1, 12674: 1, 1267
3: 1, 12672: 1, 12671: 1, 12670: 1, 12669: 1, 12668: 1, 12667: 1, 1266
6: 1, 12665: 1, 12664: 1, 12663: 1, 12662: 1, 12661: 1, 12660: 1, 1265
9: 1, 12658: 1, 12657: 1, 12656: 1, 12655: 1, 12654: 1, 12653: 1, 1265
2: 1, 12651: 1, 12650: 1, 12649: 1, 12648: 1, 12647: 1, 12646: 1, 1264
5: 1, 12644: 1, 12643: 1, 12642: 1, 12641: 1, 12640: 1, 12639: 1, 1263
8: 1, 12637: 1, 12636: 1, 12635: 1, 12634: 1, 12633: 1, 12632: 1, 1263
1: 1, 12630: 1, 12629: 1, 12628: 1, 12627: 1, 12626: 1, 12625: 1, 1262
4: 1, 12623: 1, 12622: 1, 12621: 1, 12620: 1, 12619: 1, 12618: 1, 1261
7: 1, 12616: 1, 12615: 1, 12614: 1, 12613: 1, 12612: 1, 12611: 1, 1261
0: 1, 12609: 1, 12608: 1, 12607: 1, 12606: 1, 12605: 1, 12604: 1, 1260
3: 1, 12602: 1, 12601: 1, 12600: 1, 12599: 1, 12598: 1, 12597: 1, 1259
6: 1, 12595: 1, 12594: 1, 12593: 1, 12592: 1, 12591: 1, 12590: 1, 1258
9: 1, 12588: 1, 12587: 1, 12586: 1, 12585: 1, 12584: 1, 12583: 1, 1258
2: 1, 12581: 1, 12580: 1, 12579: 1, 12578: 1, 12577: 1, 12576: 1, 1257
5: 1, 12574: 1, 12573: 1, 12572: 1, 12571: 1, 12570: 1, 12569: 1, 1256
8: 1, 12567: 1, 12566: 1, 12565: 1, 12564: 1, 12563: 1, 12562: 1, 1256
1: 1, 12560: 1, 12559: 1, 12558: 1, 12557: 1, 12556: 1, 12555: 1, 1255
4: 1, 12553: 1, 12552: 1, 12551: 1, 12550: 1, 12549: 1, 12548: 1, 1254
7: 1, 12546: 1, 12545: 1, 12544: 1, 12543: 1, 12542: 1, 12541: 1, 1254
0: 1, 12539: 1, 12538: 1, 12537: 1, 12536: 1, 12535: 1, 12534: 1, 1253
3: 1, 12532: 1, 12531: 1, 12530: 1, 12529: 1, 12528: 1, 12527: 1, 1252

6: 1, 12525: 1, 12524: 1, 12523: 1, 12522: 1, 12521: 1, 12520: 1, 1251
9: 1, 12518: 1, 12517: 1, 12516: 1, 12515: 1, 12514: 1, 12513: 1, 1251
2: 1, 12511: 1, 12510: 1, 12509: 1, 12508: 1, 12507: 1, 12506: 1, 1250
5: 1, 12504: 1, 12503: 1, 12502: 1, 12501: 1, 12500: 1, 12499: 1, 1249
8: 1, 12497: 1, 12496: 1, 12495: 1, 12494: 1, 12493: 1, 12492: 1, 1249
1: 1, 12490: 1, 12489: 1, 12488: 1, 12487: 1, 12486: 1, 12485: 1, 1248
4: 1, 12483: 1, 12482: 1, 12481: 1, 12480: 1, 12479: 1, 12478: 1, 1247
7: 1, 12476: 1, 12475: 1, 12474: 1, 12473: 1, 12472: 1, 12471: 1, 1247
0: 1, 12469: 1, 12468: 1, 12467: 1, 12466: 1, 12465: 1, 12464: 1, 1246
3: 1, 12462: 1, 12461: 1, 12460: 1, 12459: 1, 12458: 1, 12457: 1, 1245
6: 1, 12455: 1, 12454: 1, 12453: 1, 12452: 1, 12451: 1, 12450: 1, 1244
9: 1, 12448: 1, 12447: 1, 12446: 1, 12445: 1, 12444: 1, 12443: 1, 1244
2: 1, 12441: 1, 12440: 1, 12439: 1, 12438: 1, 12437: 1, 12436: 1, 1243
5: 1, 12434: 1, 12433: 1, 12432: 1, 12431: 1, 12430: 1, 12429: 1, 1242
8: 1, 12427: 1, 12426: 1, 12425: 1, 12424: 1, 12423: 1, 12422: 1, 1242
1: 1, 12420: 1, 12419: 1, 12418: 1, 12417: 1, 12416: 1, 12415: 1, 1241
4: 1, 12413: 1, 12412: 1, 12411: 1, 12410: 1, 12409: 1, 12408: 1, 1240
7: 1, 12406: 1, 12405: 1, 12404: 1, 12403: 1, 12402: 1, 12401: 1, 1240
0: 1, 12399: 1, 12398: 1, 12397: 1, 12396: 1, 12395: 1, 12394: 1, 1239
3: 1, 12392: 1, 12391: 1, 12390: 1, 12389: 1, 12388: 1, 12387: 1, 1238
6: 1, 12385: 1, 12384: 1, 12383: 1, 12382: 1, 12381: 1, 12380: 1, 1237
9: 1, 12378: 1, 12377: 1, 12376: 1, 12375: 1, 12374: 1, 12373: 1, 1237
2: 1, 12371: 1, 12370: 1, 12369: 1, 12368: 1, 12367: 1, 12366: 1, 1236
5: 1, 12364: 1, 12363: 1, 12362: 1, 12361: 1, 12360: 1, 12359: 1, 1235
8: 1, 12357: 1, 12356: 1, 12355: 1, 12354: 1, 12353: 1, 12352: 1, 1235
1: 1, 12350: 1, 12349: 1, 12348: 1, 12347: 1, 12346: 1, 12345: 1, 1234
4: 1, 12343: 1, 12342: 1, 12341: 1, 12340: 1, 12339: 1, 12338: 1, 1233
7: 1, 12336: 1, 12335: 1, 12334: 1, 12333: 1, 12332: 1, 12331: 1, 1233
0: 1, 12329: 1, 12328: 1, 12327: 1, 12326: 1, 12325: 1, 12324: 1, 1232
3: 1, 12322: 1, 12321: 1, 12320: 1, 12319: 1, 12318: 1, 12317: 1, 1231
6: 1, 12315: 1, 12314: 1, 12313: 1, 12312: 1, 12311: 1, 12310: 1, 1230
9: 1, 12308: 1, 12307: 1, 12306: 1, 12305: 1, 12304: 1, 12303: 1, 1230
2: 1, 12301: 1, 12300: 1, 12299: 1, 12298: 1, 12297: 1, 12296: 1, 1229
5: 1, 12294: 1, 12293: 1, 12292: 1, 12291: 1, 12290: 1, 12289: 1, 1228
8: 1, 12287: 1, 12286: 1, 12285: 1, 12284: 1, 12283: 1, 12282: 1, 1228
1: 1, 12280: 1, 12279: 1, 12278: 1, 12277: 1, 12276: 1, 12275: 1, 1227
4: 1, 12273: 1, 12272: 1, 12271: 1, 12270: 1, 12269: 1, 12268: 1, 1226
7: 1, 12266: 1, 12265: 1, 12264: 1, 12263: 1, 12262: 1, 12261: 1, 1226
0: 1, 12259: 1, 12258: 1, 12257: 1, 12256: 1, 12255: 1, 12254: 1, 1225

3: 1, 12252: 1, 12251: 1, 12250: 1, 12249: 1, 12248: 1, 12247: 1, 1224
6: 1, 12245: 1, 12244: 1, 12243: 1, 12242: 1, 12241: 1, 12240: 1, 1223
9: 1, 12238: 1, 12237: 1, 12236: 1, 12235: 1, 12234: 1, 12233: 1, 1223
2: 1, 12231: 1, 12230: 1, 12229: 1, 12228: 1, 12227: 1, 12226: 1, 1222
5: 1, 12224: 1, 12223: 1, 12222: 1, 12221: 1, 12220: 1, 12219: 1, 1221
8: 1, 12217: 1, 12216: 1, 12215: 1, 12214: 1, 12213: 1, 12212: 1, 1221
1: 1, 12210: 1, 12209: 1, 12208: 1, 12207: 1, 12206: 1, 12205: 1, 1220
4: 1, 12203: 1, 12202: 1, 12201: 1, 12200: 1, 12199: 1, 12198: 1, 1219
7: 1, 12196: 1, 12195: 1, 12194: 1, 12193: 1, 12192: 1, 12191: 1, 1219
0: 1, 12189: 1, 12188: 1, 12187: 1, 12186: 1, 12185: 1, 12184: 1, 1218
3: 1, 12182: 1, 12181: 1, 12180: 1, 12179: 1, 12178: 1, 12177: 1, 1217
6: 1, 12175: 1, 12174: 1, 12173: 1, 12172: 1, 12171: 1, 12170: 1, 1216
9: 1, 12168: 1, 12167: 1, 12166: 1, 12165: 1, 12164: 1, 12163: 1, 1216
2: 1, 12161: 1, 12160: 1, 12159: 1, 12158: 1, 12157: 1, 12156: 1, 1215
5: 1, 12154: 1, 12153: 1, 12152: 1, 12151: 1, 12150: 1, 12149: 1, 1214
8: 1, 12147: 1, 12146: 1, 12145: 1, 12144: 1, 12143: 1, 12142: 1, 1214
1: 1, 12140: 1, 12139: 1, 12138: 1, 12137: 1, 12136: 1, 12135: 1, 1213
4: 1, 12133: 1, 12132: 1, 12131: 1, 12130: 1, 12129: 1, 12128: 1, 1212
7: 1, 12126: 1, 12125: 1, 12124: 1, 12123: 1, 12122: 1, 12121: 1, 1212
0: 1, 12119: 1, 12118: 1, 12117: 1, 12116: 1, 12115: 1, 12114: 1, 1211
3: 1, 12112: 1, 12111: 1, 12110: 1, 12109: 1, 12108: 1, 12107: 1, 1210
6: 1, 12105: 1, 12104: 1, 12103: 1, 12102: 1, 12101: 1, 12100: 1, 1209
9: 1, 12098: 1, 12097: 1, 12096: 1, 12095: 1, 12094: 1, 12093: 1, 1209
2: 1, 12091: 1, 12090: 1, 12089: 1, 12088: 1, 12087: 1, 12086: 1, 1208
5: 1, 12084: 1, 12083: 1, 12082: 1, 12081: 1, 12080: 1, 12079: 1, 1207
8: 1, 12077: 1, 12076: 1, 12075: 1, 12074: 1, 12073: 1, 12072: 1, 1207
1: 1, 12070: 1, 12069: 1, 12068: 1, 12067: 1, 12066: 1, 12065: 1, 1206
4: 1, 12063: 1, 12062: 1, 12061: 1, 12060: 1, 12059: 1, 12058: 1, 1205
7: 1, 12056: 1, 12055: 1, 12054: 1, 12053: 1, 12052: 1, 12051: 1, 1205
0: 1, 12049: 1, 12048: 1, 12047: 1, 12046: 1, 12045: 1, 12044: 1, 1204
3: 1, 12042: 1, 12041: 1, 12040: 1, 12039: 1, 12038: 1, 12037: 1, 1203
6: 1, 12035: 1, 12034: 1, 12033: 1, 12032: 1, 12031: 1, 12030: 1, 1202
9: 1, 12028: 1, 12027: 1, 12026: 1, 12025: 1, 12024: 1, 12023: 1, 1202
2: 1, 12021: 1, 12020: 1, 12019: 1, 12018: 1, 12017: 1, 12016: 1, 1201
5: 1, 12014: 1, 12013: 1, 12012: 1, 12011: 1, 12010: 1, 12009: 1, 1200
8: 1, 12007: 1, 12006: 1, 12005: 1, 12004: 1, 12003: 1, 12002: 1, 1200
1: 1, 12000: 1, 11999: 1, 11998: 1, 11997: 1, 11996: 1, 11995: 1, 1199
4: 1, 11993: 1, 11992: 1, 11991: 1, 11990: 1, 11989: 1, 11988: 1, 1198
7: 1, 11986: 1, 11985: 1, 11984: 1, 11983: 1, 11982: 1, 11981: 1, 1198

0: 1, 11979: 1, 11978: 1, 11977: 1, 11976: 1, 11975: 1, 11974: 1, 1197
3: 1, 11972: 1, 11971: 1, 11970: 1, 11969: 1, 11968: 1, 11967: 1, 1196
6: 1, 11965: 1, 11964: 1, 11963: 1, 11962: 1, 11961: 1, 11960: 1, 1195
9: 1, 11958: 1, 11957: 1, 11956: 1, 11955: 1, 11954: 1, 11953: 1, 1195
2: 1, 11951: 1, 11950: 1, 11949: 1, 11948: 1, 11947: 1, 11946: 1, 1194
5: 1, 11944: 1, 11943: 1, 11942: 1, 11941: 1, 11940: 1, 11939: 1, 1193
8: 1, 11937: 1, 11936: 1, 11935: 1, 11934: 1, 11933: 1, 11932: 1, 1193
1: 1, 11930: 1, 11929: 1, 11928: 1, 11927: 1, 11926: 1, 11925: 1, 1192
4: 1, 11923: 1, 11922: 1, 11921: 1, 11920: 1, 11919: 1, 11918: 1, 1191
7: 1, 11916: 1, 11915: 1, 11914: 1, 11913: 1, 11912: 1, 11911: 1, 1191
0: 1, 11909: 1, 11908: 1, 11907: 1, 11906: 1, 11905: 1, 11904: 1, 1190
3: 1, 11902: 1, 11901: 1, 11900: 1, 11899: 1, 11898: 1, 11897: 1, 1189
6: 1, 11895: 1, 11894: 1, 11893: 1, 11892: 1, 11891: 1, 11890: 1, 1188
9: 1, 11888: 1, 11887: 1, 11886: 1, 11885: 1, 11884: 1, 11883: 1, 1188
2: 1, 11881: 1, 11880: 1, 11879: 1, 11878: 1, 11877: 1, 11876: 1, 1187
5: 1, 11874: 1, 11873: 1, 11872: 1, 11871: 1, 11870: 1, 11869: 1, 1186
8: 1, 11867: 1, 11866: 1, 11865: 1, 11864: 1, 11863: 1, 11862: 1, 1186
1: 1, 11860: 1, 11859: 1, 11858: 1, 11857: 1, 11856: 1, 11855: 1, 1185
4: 1, 11853: 1, 11852: 1, 11851: 1, 11850: 1, 11849: 1, 11848: 1, 1184
7: 1, 11846: 1, 11845: 1, 11844: 1, 11843: 1, 11842: 1, 11841: 1, 1184
0: 1, 11839: 1, 11838: 1, 11837: 1, 11836: 1, 11835: 1, 11834: 1, 1183
3: 1, 11832: 1, 11831: 1, 11830: 1, 11829: 1, 11828: 1, 11827: 1, 1182
6: 1, 11825: 1, 11824: 1, 11823: 1, 11822: 1, 11821: 1, 11820: 1, 1181
9: 1, 11818: 1, 11817: 1, 11816: 1, 11815: 1, 11814: 1, 11813: 1, 1181
2: 1, 11811: 1, 11810: 1, 11809: 1, 11808: 1, 11807: 1, 11806: 1, 1180
5: 1, 11804: 1, 11803: 1, 11802: 1, 11801: 1, 11800: 1, 11799: 1, 1179
8: 1, 11797: 1, 11796: 1, 11795: 1, 11794: 1, 11793: 1, 11792: 1, 1179
1: 1, 11790: 1, 11789: 1, 11788: 1, 11787: 1, 11786: 1, 11785: 1, 1178
4: 1, 11783: 1, 11782: 1, 11781: 1, 11780: 1, 11779: 1, 11778: 1, 1177
7: 1, 11776: 1, 11775: 1, 11774: 1, 11773: 1, 11772: 1, 11771: 1, 1177
0: 1, 11769: 1, 11768: 1, 11767: 1, 11766: 1, 11765: 1, 11764: 1, 1176
3: 1, 11762: 1, 11761: 1, 11760: 1, 11759: 1, 11758: 1, 11757: 1, 1175
6: 1, 11755: 1, 11754: 1, 11753: 1, 11752: 1, 11751: 1, 11750: 1, 1174
9: 1, 11748: 1, 11747: 1, 11746: 1, 11745: 1, 11744: 1, 11743: 1, 1174
2: 1, 11741: 1, 11740: 1, 11739: 1, 11738: 1, 11737: 1, 11736: 1, 1173
5: 1, 11734: 1, 11733: 1, 11732: 1, 11731: 1, 11730: 1, 11729: 1, 1172
8: 1, 11727: 1, 11726: 1, 11725: 1, 11724: 1, 11723: 1, 11722: 1, 1172
1: 1, 11720: 1, 11719: 1, 11718: 1, 11717: 1, 11716: 1, 11715: 1, 1171
4: 1, 11713: 1, 11712: 1, 11711: 1, 11710: 1, 11709: 1, 11708: 1, 1170

7: 1, 11706: 1, 11705: 1, 11704: 1, 11703: 1, 11702: 1, 11701: 1, 1170
0: 1, 11699: 1, 11698: 1, 11697: 1, 11696: 1, 11695: 1, 11694: 1, 1169
3: 1, 11692: 1, 11691: 1, 11690: 1, 11689: 1, 11688: 1, 11687: 1, 1168
6: 1, 11685: 1, 11684: 1, 11683: 1, 11682: 1, 11681: 1, 11680: 1, 1167
9: 1, 11678: 1, 11677: 1, 11676: 1, 11675: 1, 11674: 1, 11673: 1, 1167
2: 1, 11671: 1, 11670: 1, 11669: 1, 11668: 1, 11667: 1, 11666: 1, 1166
5: 1, 11664: 1, 11663: 1, 11662: 1, 11661: 1, 11660: 1, 11659: 1, 1165
8: 1, 11657: 1, 11656: 1, 11655: 1, 11654: 1, 11653: 1, 11652: 1, 1165
1: 1, 11650: 1, 11649: 1, 11648: 1, 11647: 1, 11646: 1, 11645: 1, 1164
4: 1, 11643: 1, 11642: 1, 11641: 1, 11640: 1, 11639: 1, 11638: 1, 1163
7: 1, 11636: 1, 11635: 1, 11634: 1, 11633: 1, 11632: 1, 11631: 1, 1163
0: 1, 11629: 1, 11628: 1, 11627: 1, 11626: 1, 11625: 1, 11624: 1, 1162
3: 1, 11622: 1, 11621: 1, 11620: 1, 11619: 1, 11618: 1, 11617: 1, 1161
6: 1, 11615: 1, 11614: 1, 11613: 1, 11612: 1, 11611: 1, 11610: 1, 1160
9: 1, 11608: 1, 11607: 1, 11606: 1, 11605: 1, 11604: 1, 11603: 1, 1160
2: 1, 11601: 1, 11600: 1, 11599: 1, 11598: 1, 11597: 1, 11596: 1, 1159
5: 1, 11594: 1, 11593: 1, 11592: 1, 11591: 1, 11590: 1, 11589: 1, 1158
8: 1, 11587: 1, 11586: 1, 11585: 1, 11584: 1, 11583: 1, 11582: 1, 1158
1: 1, 11580: 1, 11579: 1, 11578: 1, 11577: 1, 11576: 1, 11575: 1, 1157
4: 1, 11573: 1, 11572: 1, 11571: 1, 11570: 1, 11569: 1, 11568: 1, 1156
7: 1, 11566: 1, 11565: 1, 11564: 1, 11563: 1, 11562: 1, 11561: 1, 1156
0: 1, 11559: 1, 11558: 1, 11557: 1, 11556: 1, 11555: 1, 11554: 1, 1155
3: 1, 11552: 1, 11551: 1, 11550: 1, 11549: 1, 11548: 1, 11547: 1, 1154
6: 1, 11545: 1, 11544: 1, 11543: 1, 11542: 1, 11541: 1, 11540: 1, 1153
9: 1, 11538: 1, 11537: 1, 11536: 1, 11535: 1, 11534: 1, 11533: 1, 1153
2: 1, 11531: 1, 11530: 1, 11529: 1, 11528: 1, 11527: 1, 11526: 1, 1152
5: 1, 11524: 1, 11523: 1, 11522: 1, 11521: 1, 11520: 1, 11519: 1, 1151
8: 1, 11517: 1, 11516: 1, 11515: 1, 11514: 1, 11513: 1, 11512: 1, 1151
1: 1, 11510: 1, 11509: 1, 11508: 1, 11507: 1, 11506: 1, 11505: 1, 1150
4: 1, 11503: 1, 11502: 1, 11501: 1, 11500: 1, 11499: 1, 11498: 1, 1149
7: 1, 11496: 1, 11495: 1, 11494: 1, 11493: 1, 11492: 1, 11491: 1, 1149
0: 1, 11489: 1, 11488: 1, 11487: 1, 11486: 1, 11485: 1, 11484: 1, 1148
3: 1, 11482: 1, 11481: 1, 11480: 1, 11479: 1, 11478: 1, 11477: 1, 1147
6: 1, 11475: 1, 11474: 1, 11473: 1, 11472: 1, 11471: 1, 11470: 1, 1146
9: 1, 11468: 1, 11467: 1, 11466: 1, 11465: 1, 11464: 1, 11463: 1, 1146
2: 1, 11461: 1, 11460: 1, 11459: 1, 11458: 1, 11457: 1, 11456: 1, 1145
5: 1, 11454: 1, 11453: 1, 11452: 1, 11451: 1, 11450: 1, 11449: 1, 1144
8: 1, 11447: 1, 11446: 1, 11445: 1, 11444: 1, 11443: 1, 11442: 1, 1144
1: 1, 11440: 1, 11439: 1, 11438: 1, 11437: 1, 11436: 1, 11435: 1, 1143

4: 1, 11433: 1, 11432: 1, 11431: 1, 11430: 1, 11429: 1, 11428: 1, 1142
7: 1, 11426: 1, 11425: 1, 11424: 1, 11423: 1, 11422: 1, 11421: 1, 1142
0: 1, 11419: 1, 11418: 1, 11417: 1, 11416: 1, 11415: 1, 11414: 1, 1141
3: 1, 11412: 1, 11411: 1, 11410: 1, 11409: 1, 11408: 1, 11407: 1, 1140
6: 1, 11405: 1, 11404: 1, 11403: 1, 11402: 1, 11401: 1, 11400: 1, 1139
9: 1, 11398: 1, 11397: 1, 11396: 1, 11395: 1, 11394: 1, 11393: 1, 1139
2: 1, 11391: 1, 11390: 1, 11389: 1, 11388: 1, 11387: 1, 11386: 1, 1138
5: 1, 11384: 1, 11383: 1, 11382: 1, 11381: 1, 11380: 1, 11379: 1, 1137
8: 1, 11377: 1, 11376: 1, 11375: 1, 11374: 1, 11373: 1, 11372: 1, 1137
1: 1, 11370: 1, 11369: 1, 11368: 1, 11367: 1, 11366: 1, 11365: 1, 1136
4: 1, 11363: 1, 11362: 1, 11361: 1, 11360: 1, 11359: 1, 11358: 1, 1135
7: 1, 11356: 1, 11355: 1, 11354: 1, 11353: 1, 11352: 1, 11351: 1, 1135
0: 1, 11349: 1, 11348: 1, 11347: 1, 11346: 1, 11345: 1, 11344: 1, 1134
3: 1, 11342: 1, 11341: 1, 11340: 1, 11339: 1, 11338: 1, 11337: 1, 1133
6: 1, 11335: 1, 11334: 1, 11333: 1, 11332: 1, 11331: 1, 11330: 1, 1132
9: 1, 11328: 1, 11327: 1, 11326: 1, 11325: 1, 11324: 1, 11323: 1, 1132
2: 1, 11321: 1, 11320: 1, 11319: 1, 11318: 1, 11317: 1, 11316: 1, 1131
5: 1, 11314: 1, 11313: 1, 11312: 1, 11311: 1, 11310: 1, 11309: 1, 1130
8: 1, 11307: 1, 11306: 1, 11305: 1, 11304: 1, 11303: 1, 11302: 1, 1130
1: 1, 11300: 1, 11299: 1, 11298: 1, 11297: 1, 11296: 1, 11295: 1, 1129
4: 1, 11293: 1, 11292: 1, 11291: 1, 11290: 1, 11289: 1, 11288: 1, 1128
7: 1, 11286: 1, 11285: 1, 11284: 1, 11283: 1, 11282: 1, 11281: 1, 1128
0: 1, 11279: 1, 11278: 1, 11277: 1, 11276: 1, 11275: 1, 11274: 1, 1127
3: 1, 11272: 1, 11271: 1, 11270: 1, 11269: 1, 11268: 1, 11267: 1, 1126
6: 1, 11265: 1, 11264: 1, 11263: 1, 11262: 1, 11261: 1, 11260: 1, 1125
9: 1, 11258: 1, 11257: 1, 11256: 1, 11255: 1, 11254: 1, 11253: 1, 1125
2: 1, 11251: 1, 11250: 1, 11249: 1, 11248: 1, 11247: 1, 11246: 1, 1124
5: 1, 11244: 1, 11243: 1, 11242: 1, 11241: 1, 11240: 1, 11239: 1, 1123
8: 1, 11237: 1, 11236: 1, 11235: 1, 11234: 1, 11233: 1, 11232: 1, 1123
1: 1, 11230: 1, 11229: 1, 11228: 1, 11227: 1, 11226: 1, 11225: 1, 1122
4: 1, 11223: 1, 11222: 1, 11221: 1, 11220: 1, 11219: 1, 11218: 1, 1121
7: 1, 11216: 1, 11215: 1, 11214: 1, 11213: 1, 11212: 1, 11211: 1, 1121
0: 1, 11209: 1, 11208: 1, 11207: 1, 11206: 1, 11205: 1, 11204: 1, 1120
3: 1, 11202: 1, 11201: 1, 11200: 1, 11199: 1, 11198: 1, 11197: 1, 1119
6: 1, 11195: 1, 11194: 1, 11193: 1, 11192: 1, 11191: 1, 11190: 1, 1118
9: 1, 11188: 1, 11187: 1, 11186: 1, 11185: 1, 11184: 1, 11183: 1, 1118
2: 1, 11181: 1, 11180: 1, 11179: 1, 11178: 1, 11177: 1, 11176: 1, 1117
5: 1, 11174: 1, 11173: 1, 11172: 1, 11171: 1, 11170: 1, 11169: 1, 1116
8: 1, 11167: 1, 11166: 1, 11165: 1, 11164: 1, 11163: 1, 11162: 1, 1116

1: 1, 11160: 1, 11159: 1, 11158: 1, 11157: 1, 11156: 1, 11155: 1, 1115
4: 1, 11153: 1, 11152: 1, 11151: 1, 11150: 1, 11149: 1, 11148: 1, 1114
7: 1, 11146: 1, 11145: 1, 11144: 1, 11143: 1, 11142: 1, 11141: 1, 1114
0: 1, 11139: 1, 11138: 1, 11137: 1, 11136: 1, 11135: 1, 11134: 1, 1113
3: 1, 11132: 1, 11131: 1, 11130: 1, 11129: 1, 11128: 1, 11127: 1, 1112
6: 1, 11125: 1, 11124: 1, 11123: 1, 11122: 1, 11121: 1, 11120: 1, 1111
9: 1, 11118: 1, 11117: 1, 11116: 1, 11115: 1, 11114: 1, 11113: 1, 1111
2: 1, 11111: 1, 11110: 1, 11109: 1, 11108: 1, 11107: 1, 11106: 1, 1110
5: 1, 11104: 1, 11103: 1, 11102: 1, 11101: 1, 11100: 1, 11099: 1, 1109
8: 1, 11097: 1, 11096: 1, 11095: 1, 11094: 1, 11093: 1, 11092: 1, 1109
1: 1, 11090: 1, 11089: 1, 11088: 1, 11087: 1, 11086: 1, 11085: 1, 1108
4: 1, 11083: 1, 11082: 1, 11081: 1, 11080: 1, 11079: 1, 11078: 1, 1107
7: 1, 11076: 1, 11075: 1, 11074: 1, 11073: 1, 11072: 1, 11071: 1, 1107
0: 1, 11069: 1, 11068: 1, 11067: 1, 11066: 1, 11065: 1, 11064: 1, 1106
3: 1, 11062: 1, 11061: 1, 11060: 1, 11059: 1, 11058: 1, 11057: 1, 1105
6: 1, 11055: 1, 11054: 1, 11053: 1, 11052: 1, 11051: 1, 11050: 1, 1104
9: 1, 11048: 1, 11047: 1, 11046: 1, 11045: 1, 11044: 1, 11043: 1, 1104
2: 1, 11041: 1, 11040: 1, 11039: 1, 11038: 1, 11037: 1, 11036: 1, 1103
5: 1, 11034: 1, 11033: 1, 11032: 1, 11031: 1, 11030: 1, 11029: 1, 1102
8: 1, 11027: 1, 11026: 1, 11025: 1, 11024: 1, 11023: 1, 11022: 1, 1102
1: 1, 11020: 1, 11019: 1, 11018: 1, 11017: 1, 11016: 1, 11015: 1, 1101
4: 1, 11013: 1, 11012: 1, 11011: 1, 11010: 1, 11009: 1, 11008: 1, 1100
7: 1, 11006: 1, 11005: 1, 11004: 1, 11003: 1, 11002: 1, 11001: 1, 1100
0: 1, 10999: 1, 10998: 1, 10997: 1, 10996: 1, 10995: 1, 10994: 1, 1099
3: 1, 10992: 1, 10991: 1, 10990: 1, 10989: 1, 10988: 1, 10987: 1, 1098
6: 1, 10985: 1, 10984: 1, 10983: 1, 10982: 1, 10981: 1, 10980: 1, 1097
9: 1, 10978: 1, 10977: 1, 10976: 1, 10975: 1, 10974: 1, 10973: 1, 1097
2: 1, 10971: 1, 10970: 1, 10969: 1, 10968: 1, 10967: 1, 10966: 1, 1096
5: 1, 10964: 1, 10963: 1, 10962: 1, 10961: 1, 10960: 1, 10959: 1, 1095
8: 1, 10957: 1, 10956: 1, 10955: 1, 10954: 1, 10953: 1, 10952: 1, 1095
1: 1, 10950: 1, 10949: 1, 10948: 1, 10947: 1, 10946: 1, 10945: 1, 1094
4: 1, 10943: 1, 10942: 1, 10941: 1, 10940: 1, 10939: 1, 10938: 1, 1093
7: 1, 10936: 1, 10935: 1, 10934: 1, 10933: 1, 10932: 1, 10931: 1, 1093
0: 1, 10929: 1, 10928: 1, 10927: 1, 10926: 1, 10925: 1, 10924: 1, 1092
3: 1, 10922: 1, 10921: 1, 10920: 1, 10919: 1, 10918: 1, 10917: 1, 1091
6: 1, 10915: 1, 10914: 1, 10913: 1, 10912: 1, 10911: 1, 10910: 1, 1090
9: 1, 10908: 1, 10907: 1, 10906: 1, 10905: 1, 10904: 1, 10903: 1, 1090
2: 1, 10901: 1, 10900: 1, 10899: 1, 10898: 1, 10897: 1, 10896: 1, 1089
5: 1, 10894: 1, 10893: 1, 10892: 1, 10891: 1, 10890: 1, 10889: 1, 1088

8: 1, 10887: 1, 10886: 1, 10885: 1, 10884: 1, 10883: 1, 10882: 1, 1088
1: 1, 10880: 1, 10879: 1, 10878: 1, 10877: 1, 10876: 1, 10875: 1, 1087
4: 1, 10873: 1, 10872: 1, 10871: 1, 10870: 1, 10869: 1, 10868: 1, 1086
7: 1, 10866: 1, 10865: 1, 10864: 1, 10863: 1, 10862: 1, 10861: 1, 1086
0: 1, 10859: 1, 10858: 1, 10857: 1, 10856: 1, 10855: 1, 10854: 1, 1085
3: 1, 10852: 1, 10851: 1, 10850: 1, 10849: 1, 10848: 1, 10847: 1, 1084
6: 1, 10845: 1, 10844: 1, 10843: 1, 10842: 1, 10841: 1, 10840: 1, 1083
9: 1, 10838: 1, 10837: 1, 10836: 1, 10835: 1, 10834: 1, 10833: 1, 1083
2: 1, 10831: 1, 10830: 1, 10829: 1, 10828: 1, 10827: 1, 10826: 1, 1082
5: 1, 10824: 1, 10823: 1, 10822: 1, 10821: 1, 10820: 1, 10819: 1, 1081
8: 1, 10817: 1, 10816: 1, 10815: 1, 10814: 1, 10813: 1, 10812: 1, 1081
1: 1, 10810: 1, 10809: 1, 10808: 1, 10807: 1, 10806: 1, 10805: 1, 1080
4: 1, 10803: 1, 10802: 1, 10801: 1, 10800: 1, 10799: 1, 10798: 1, 1079
7: 1, 10796: 1, 10795: 1, 10794: 1, 10793: 1, 10792: 1, 10791: 1, 1079
0: 1, 10789: 1, 10788: 1, 10787: 1, 10786: 1, 10785: 1, 10784: 1, 1078
3: 1, 10782: 1, 10781: 1, 10780: 1, 10779: 1, 10778: 1, 10777: 1, 1077
6: 1, 10775: 1, 10774: 1, 10773: 1, 10772: 1, 10771: 1, 10770: 1, 1076
9: 1, 10768: 1, 10767: 1, 10766: 1, 10765: 1, 10764: 1, 10763: 1, 1076
2: 1, 10761: 1, 10760: 1, 10759: 1, 10758: 1, 10757: 1, 10756: 1, 1075
5: 1, 10754: 1, 10753: 1, 10752: 1, 10751: 1, 10750: 1, 10749: 1, 1074
8: 1, 10747: 1, 10746: 1, 10745: 1, 10744: 1, 10743: 1, 10742: 1, 1074
1: 1, 10740: 1, 10739: 1, 10738: 1, 10737: 1, 10736: 1, 10735: 1, 1073
4: 1, 10733: 1, 10732: 1, 10731: 1, 10730: 1, 10729: 1, 10728: 1, 1072
7: 1, 10726: 1, 10725: 1, 10724: 1, 10723: 1, 10722: 1, 10721: 1, 1072
0: 1, 10719: 1, 10718: 1, 10717: 1, 10716: 1, 10715: 1, 10714: 1, 1071
3: 1, 10712: 1, 10711: 1, 10710: 1, 10709: 1, 10708: 1, 10707: 1, 1070
6: 1, 10705: 1, 10704: 1, 10703: 1, 10702: 1, 10701: 1, 10700: 1, 1069
9: 1, 10698: 1, 10697: 1, 10696: 1, 10695: 1, 10694: 1, 10693: 1, 1069
2: 1, 10691: 1, 10690: 1, 10689: 1, 10688: 1, 10687: 1, 10686: 1, 1068
5: 1, 10684: 1, 10683: 1, 10682: 1, 10681: 1, 10680: 1, 10679: 1, 1067
8: 1, 10677: 1, 10676: 1, 10675: 1, 10674: 1, 10673: 1, 10672: 1, 1067
1: 1, 10670: 1, 10669: 1, 10668: 1, 10667: 1, 10666: 1, 10665: 1, 1066
4: 1, 10663: 1, 10662: 1, 10661: 1, 10660: 1, 10659: 1, 10658: 1, 1065
7: 1, 10656: 1, 10655: 1, 10654: 1, 10653: 1, 10652: 1, 10651: 1, 1065
0: 1, 10649: 1, 10648: 1, 10647: 1, 10646: 1, 10645: 1, 10644: 1, 1064
3: 1, 10642: 1, 10641: 1, 10640: 1, 10639: 1, 10638: 1, 10637: 1, 1063
6: 1, 10635: 1, 10634: 1, 10633: 1, 10632: 1, 10631: 1, 10630: 1, 1062
9: 1, 10628: 1, 10627: 1, 10626: 1, 10625: 1, 10624: 1, 10623: 1, 1062
2: 1, 10621: 1, 10620: 1, 10619: 1, 10618: 1, 10617: 1, 10616: 1, 1061

5: 1, 10614: 1, 10613: 1, 10612: 1, 10611: 1, 10610: 1, 10609: 1, 1060
8: 1, 10607: 1, 10606: 1, 10605: 1, 10604: 1, 10603: 1, 10602: 1, 1060
1: 1, 10600: 1, 10599: 1, 10598: 1, 10597: 1, 10596: 1, 10595: 1, 1059
4: 1, 10593: 1, 10592: 1, 10591: 1, 10590: 1, 10589: 1, 10588: 1, 1058
7: 1, 10586: 1, 10585: 1, 10584: 1, 10583: 1, 10582: 1, 10581: 1, 1058
0: 1, 10579: 1, 10578: 1, 10577: 1, 10576: 1, 10575: 1, 10574: 1, 1057
3: 1, 10572: 1, 10571: 1, 10570: 1, 10569: 1, 10568: 1, 10567: 1, 1056
6: 1, 10565: 1, 10564: 1, 10563: 1, 10562: 1, 10561: 1, 10560: 1, 1055
9: 1, 10558: 1, 10557: 1, 10556: 1, 10555: 1, 10554: 1, 10553: 1, 1055
2: 1, 10551: 1, 10550: 1, 10549: 1, 10548: 1, 10547: 1, 10546: 1, 1054
5: 1, 10544: 1, 10543: 1, 10542: 1, 10541: 1, 10540: 1, 10539: 1, 1053
8: 1, 10537: 1, 10536: 1, 10535: 1, 10534: 1, 10533: 1, 10532: 1, 1053
1: 1, 10530: 1, 10529: 1, 10528: 1, 10527: 1, 10526: 1, 10525: 1, 1052
4: 1, 10523: 1, 10522: 1, 10521: 1, 10520: 1, 10519: 1, 10518: 1, 1051
7: 1, 10516: 1, 10515: 1, 10514: 1, 10513: 1, 10512: 1, 10511: 1, 1051
0: 1, 10509: 1, 10508: 1, 10507: 1, 10506: 1, 10505: 1, 10504: 1, 1050
3: 1, 10502: 1, 10501: 1, 10500: 1, 10499: 1, 10498: 1, 10497: 1, 1049
6: 1, 10495: 1, 10494: 1, 10493: 1, 10492: 1, 10491: 1, 10490: 1, 1048
9: 1, 10488: 1, 10487: 1, 10486: 1, 10485: 1, 10484: 1, 10483: 1, 1048
2: 1, 10481: 1, 10480: 1, 10479: 1, 10478: 1, 10477: 1, 10476: 1, 1047
5: 1, 10474: 1, 10473: 1, 10472: 1, 10471: 1, 10470: 1, 10469: 1, 1046
8: 1, 10467: 1, 10466: 1, 10465: 1, 10464: 1, 10463: 1, 10462: 1, 1046
1: 1, 10460: 1, 10459: 1, 10458: 1, 10457: 1, 10456: 1, 10455: 1, 1045
4: 1, 10453: 1, 10452: 1, 10451: 1, 10450: 1, 10449: 1, 10448: 1, 1044
7: 1, 10446: 1, 10445: 1, 10444: 1, 10443: 1, 10442: 1, 10441: 1, 1044
0: 1, 10439: 1, 10438: 1, 10437: 1, 10436: 1, 10435: 1, 10434: 1, 1043
3: 1, 10432: 1, 10431: 1, 10430: 1, 10429: 1, 10428: 1, 10427: 1, 1042
6: 1, 10425: 1, 10424: 1, 10423: 1, 10422: 1, 10421: 1, 10420: 1, 1041
9: 1, 10418: 1, 10417: 1, 10416: 1, 10415: 1, 10414: 1, 10413: 1, 1041
2: 1, 10411: 1, 10410: 1, 10409: 1, 10408: 1, 10407: 1, 10406: 1, 1040
5: 1, 10404: 1, 10403: 1, 10402: 1, 10401: 1, 10400: 1, 10399: 1, 1039
8: 1, 10397: 1, 10396: 1, 10395: 1, 10394: 1, 10393: 1, 10392: 1, 1039
1: 1, 10390: 1, 10389: 1, 10388: 1, 10387: 1, 10386: 1, 10385: 1, 1038
4: 1, 10383: 1, 10382: 1, 10381: 1, 10380: 1, 10379: 1, 10378: 1, 1037
7: 1, 10376: 1, 10375: 1, 10374: 1, 10373: 1, 10372: 1, 10371: 1, 1037
0: 1, 10369: 1, 10368: 1, 10367: 1, 10366: 1, 10365: 1, 10364: 1, 1036
3: 1, 10362: 1, 10361: 1, 10360: 1, 10359: 1, 10358: 1, 10357: 1, 1035
6: 1, 10355: 1, 10354: 1, 10353: 1, 10352: 1, 10351: 1, 10350: 1, 1034
9: 1, 10348: 1, 10347: 1, 10346: 1, 10345: 1, 10344: 1, 10343: 1, 1034

2: 1, 10341: 1, 10340: 1, 10339: 1, 10338: 1, 10337: 1, 10336: 1, 1033
5: 1, 10334: 1, 10333: 1, 10332: 1, 10331: 1, 10330: 1, 10329: 1, 1032
8: 1, 10327: 1, 10326: 1, 10325: 1, 10324: 1, 10323: 1, 10322: 1, 1032
1: 1, 10320: 1, 10319: 1, 10318: 1, 10317: 1, 10316: 1, 10315: 1, 1031
4: 1, 10313: 1, 10312: 1, 10311: 1, 10310: 1, 10309: 1, 10308: 1, 1030
7: 1, 10306: 1, 10305: 1, 10304: 1, 10303: 1, 10302: 1, 10301: 1, 1030
0: 1, 10299: 1, 10298: 1, 10297: 1, 10296: 1, 10295: 1, 10294: 1, 1029
3: 1, 10292: 1, 10291: 1, 10290: 1, 10289: 1, 10288: 1, 10287: 1, 1028
6: 1, 10285: 1, 10284: 1, 10283: 1, 10282: 1, 10281: 1, 10280: 1, 1027
9: 1, 10278: 1, 10277: 1, 10276: 1, 10275: 1, 10274: 1, 10273: 1, 1027
2: 1, 10271: 1, 10270: 1, 10269: 1, 10268: 1, 10267: 1, 10266: 1, 1026
5: 1, 10264: 1, 10263: 1, 10262: 1, 10261: 1, 10260: 1, 10259: 1, 1025
8: 1, 10257: 1, 10256: 1, 10255: 1, 10254: 1, 10253: 1, 10252: 1, 1025
1: 1, 10250: 1, 10249: 1, 10248: 1, 10247: 1, 10246: 1, 10245: 1, 1024
4: 1, 10243: 1, 10242: 1, 10241: 1, 10240: 1, 10239: 1, 10238: 1, 1023
7: 1, 10236: 1, 10235: 1, 10234: 1, 10233: 1, 10232: 1, 10231: 1, 1023
0: 1, 10229: 1, 10228: 1, 10227: 1, 10226: 1, 10225: 1, 10224: 1, 1022
3: 1, 10222: 1, 10221: 1, 10220: 1, 10219: 1, 10218: 1, 10217: 1, 1021
6: 1, 10215: 1, 10214: 1, 10213: 1, 10212: 1, 10211: 1, 10210: 1, 1020
9: 1, 10208: 1, 10207: 1, 10206: 1, 10205: 1, 10204: 1, 10203: 1, 1020
2: 1, 10201: 1, 10200: 1, 10199: 1, 10198: 1, 10197: 1, 10196: 1, 1019
5: 1, 10194: 1, 10193: 1, 10192: 1, 10191: 1, 10190: 1, 10189: 1, 1018
8: 1, 10187: 1, 10186: 1, 10185: 1, 10184: 1, 10183: 1, 10182: 1, 1018
1: 1, 10180: 1, 10179: 1, 10178: 1, 10177: 1, 10176: 1, 10175: 1, 1017
4: 1, 10173: 1, 10172: 1, 10171: 1, 10170: 1, 10169: 1, 10168: 1, 1016
7: 1, 10166: 1, 10165: 1, 10164: 1, 10163: 1, 10162: 1, 10161: 1, 1016
0: 1, 10159: 1, 10158: 1, 10157: 1, 10156: 1, 10155: 1, 10154: 1, 1015
3: 1, 10152: 1, 10151: 1, 10150: 1, 10149: 1, 10148: 1, 10147: 1, 1014
6: 1, 10145: 1, 10144: 1, 10143: 1, 10142: 1, 10141: 1, 10140: 1, 1013
9: 1, 10138: 1, 10137: 1, 10136: 1, 10135: 1, 10134: 1, 10133: 1, 1013
2: 1, 10131: 1, 10130: 1, 10129: 1, 10128: 1, 10127: 1, 10126: 1, 1012
5: 1, 10124: 1, 10123: 1, 10122: 1, 10121: 1, 10120: 1, 10119: 1, 1011
8: 1, 10117: 1, 10116: 1, 10115: 1, 10114: 1, 10113: 1, 10112: 1, 1011
1: 1, 10110: 1, 10109: 1, 10108: 1, 10107: 1, 10106: 1, 10105: 1, 1010
4: 1, 10103: 1, 10102: 1, 10101: 1, 10100: 1, 10099: 1, 10098: 1, 1009
7: 1, 10096: 1, 10095: 1, 10094: 1, 10093: 1, 10092: 1, 10091: 1, 1009
0: 1, 10089: 1, 10088: 1, 10087: 1, 10086: 1, 10085: 1, 10084: 1, 1008
3: 1, 10082: 1, 10081: 1, 10080: 1, 10079: 1, 10078: 1, 10077: 1, 1007
6: 1, 10075: 1, 10074: 1, 10073: 1, 10072: 1, 10071: 1, 10070: 1, 1006

9: 1, 10068: 1, 10067: 1, 10066: 1, 10065: 1, 10064: 1, 10063: 1, 1006
2: 1, 10061: 1, 10060: 1, 10059: 1, 10058: 1, 10057: 1, 10056: 1, 1005
5: 1, 10054: 1, 10053: 1, 10052: 1, 10051: 1, 10050: 1, 10049: 1, 1004
8: 1, 10047: 1, 10046: 1, 10045: 1, 10044: 1, 10043: 1, 10042: 1, 1004
1: 1, 10040: 1, 10039: 1, 10038: 1, 10037: 1, 10036: 1, 10035: 1, 1003
4: 1, 10033: 1, 10032: 1, 10031: 1, 10030: 1, 10029: 1, 10028: 1, 1002
7: 1, 10026: 1, 10025: 1, 10024: 1, 10023: 1, 10022: 1, 10021: 1, 1002
0: 1, 10019: 1, 10018: 1, 10017: 1, 10016: 1, 10015: 1, 10014: 1, 1001
3: 1, 10012: 1, 10011: 1, 10010: 1, 10009: 1, 10008: 1, 10007: 1, 1000
6: 1, 10005: 1, 10004: 1, 10003: 1, 10002: 1, 10001: 1, 10000: 1, 9999:
1, 9998: 1, 9997: 1, 9996: 1, 9995: 1, 9994: 1, 9993: 1, 9992: 1, 999
1: 1, 9990: 1, 9989: 1, 9988: 1, 9987: 1, 9986: 1, 9985: 1, 9984: 1, 99
83: 1, 9982: 1, 9981: 1, 9980: 1, 9979: 1, 9978: 1, 9977: 1, 9976: 1, 9
975: 1, 9974: 1, 9973: 1, 9972: 1, 9971: 1, 9970: 1, 9969: 1, 9968: 1,
9967: 1, 9966: 1, 9965: 1, 9964: 1, 9963: 1, 9962: 1, 9961: 1, 9960:
1, 9959: 1, 9958: 1, 9957: 1, 9956: 1, 9955: 1, 9954: 1, 9953: 1, 995
2: 1, 9951: 1, 9950: 1, 9949: 1, 9948: 1, 9947: 1, 9946: 1, 9945: 1, 99
44: 1, 9943: 1, 9942: 1, 9941: 1, 9940: 1, 9939: 1, 9938: 1, 9937: 1, 9
936: 1, 9935: 1, 9934: 1, 9933: 1, 9932: 1, 9931: 1, 9930: 1, 9929: 1,
9928: 1, 9927: 1, 9926: 1, 9925: 1, 9924: 1, 9923: 1, 9922: 1, 9921:
1, 9920: 1, 9919: 1, 9918: 1, 9917: 1, 9916: 1, 9915: 1, 9914: 1, 991
3: 1, 9912: 1, 9911: 1, 9910: 1, 9909: 1, 9908: 1, 9907: 1, 9906: 1, 99
05: 1, 9904: 1, 9903: 1, 9902: 1, 9901: 1, 9900: 1, 9899: 1, 9898: 1, 9
897: 1, 9896: 1, 9895: 1, 9894: 1, 9893: 1, 9892: 1, 9891: 1, 9890: 1,
9889: 1, 9888: 1, 9887: 1, 9886: 1, 9885: 1, 9884: 1, 9883: 1, 9882:
1, 9881: 1, 9880: 1, 9879: 1, 9878: 1, 9877: 1, 9876: 1, 9875: 1, 987
4: 1, 9873: 1, 9872: 1, 9871: 1, 9870: 1, 9869: 1, 9868: 1, 9867: 1, 98
66: 1, 9865: 1, 9864: 1, 9863: 1, 9862: 1, 9861: 1, 9860: 1, 9859: 1, 9
858: 1, 9857: 1, 9856: 1, 9855: 1, 9854: 1, 9853: 1, 9852: 1, 9851: 1,
9850: 1, 9849: 1, 9848: 1, 9847: 1, 9846: 1, 9845: 1, 9844: 1, 9843:
1, 9842: 1, 9841: 1, 9840: 1, 9839: 1, 9838: 1, 9837: 1, 9836: 1, 983
5: 1, 9834: 1, 9833: 1, 9832: 1, 9831: 1, 9830: 1, 9829: 1, 9828: 1, 98
27: 1, 9826: 1, 9825: 1, 9824: 1, 9823: 1, 9822: 1, 9821: 1, 9820: 1, 9
819: 1, 9818: 1, 9817: 1, 9816: 1, 9815: 1, 9814: 1, 9813: 1, 9812: 1,
9811: 1, 9810: 1, 9809: 1, 9808: 1, 9807: 1, 9806: 1, 9805: 1, 9804:
1, 9803: 1, 9802: 1, 9801: 1, 9800: 1, 9799: 1, 9798: 1, 9797: 1, 979
6: 1, 9795: 1, 9794: 1, 9793: 1, 9792: 1, 9791: 1, 9790: 1, 9789: 1, 97
88: 1, 9787: 1, 9786: 1, 9785: 1, 9784: 1, 9783: 1, 9782: 1, 9781: 1, 9
780: 1, 9779: 1, 9778: 1, 9777: 1, 9776: 1, 9775: 1, 9774: 1, 9773: 1,

9772: 1, 9771: 1, 9770: 1, 9769: 1, 9768: 1, 9767: 1, 9766: 1, 9765:
1, 9764: 1, 9763: 1, 9762: 1, 9761: 1, 9760: 1, 9759: 1, 9758: 1, 975
7: 1, 9756: 1, 9755: 1, 9754: 1, 9753: 1, 9752: 1, 9751: 1, 9750: 1, 97
49: 1, 9748: 1, 9747: 1, 9746: 1, 9745: 1, 9744: 1, 9743: 1, 9742: 1, 9
741: 1, 9740: 1, 9739: 1, 9738: 1, 9737: 1, 9736: 1, 9735: 1, 9734: 1,
9733: 1, 9732: 1, 9731: 1, 9730: 1, 9729: 1, 9728: 1, 9727: 1, 9726:
1, 9725: 1, 9724: 1, 9723: 1, 9722: 1, 9721: 1, 9720: 1, 9719: 1, 971
8: 1, 9717: 1, 9716: 1, 9715: 1, 9714: 1, 9713: 1, 9712: 1, 9711: 1, 97
10: 1, 9709: 1, 9708: 1, 9707: 1, 9706: 1, 9705: 1, 9704: 1, 9703: 1, 9
702: 1, 9701: 1, 9700: 1, 9699: 1, 9698: 1, 9697: 1, 9696: 1, 9695: 1,
9694: 1, 9693: 1, 9692: 1, 9691: 1, 9690: 1, 9689: 1, 9688: 1, 9687:
1, 9686: 1, 9685: 1, 9684: 1, 9683: 1, 9682: 1, 9681: 1, 9680: 1, 967
9: 1, 9678: 1, 9677: 1, 9676: 1, 9675: 1, 9674: 1, 9673: 1, 9672: 1, 96
71: 1, 9670: 1, 9669: 1, 9668: 1, 9667: 1, 9666: 1, 9665: 1, 9664: 1, 9
663: 1, 9662: 1, 9661: 1, 9660: 1, 9659: 1, 9658: 1, 9657: 1, 9656: 1,
9655: 1, 9654: 1, 9653: 1, 9652: 1, 9651: 1, 9650: 1, 9649: 1, 9648:
1, 9647: 1, 9646: 1, 9645: 1, 9644: 1, 9643: 1, 9642: 1, 9641: 1, 964
0: 1, 9639: 1, 9638: 1, 9637: 1, 9636: 1, 9635: 1, 9634: 1, 9633: 1, 96
32: 1, 9631: 1, 9630: 1, 9629: 1, 9628: 1, 9627: 1, 9626: 1, 9625: 1, 9
624: 1, 9623: 1, 9622: 1, 9621: 1, 9620: 1, 9619: 1, 9618: 1, 9617: 1,
9616: 1, 9615: 1, 9614: 1, 9613: 1, 9612: 1, 9611: 1, 9610: 1, 9609:
1, 9608: 1, 9607: 1, 9606: 1, 9605: 1, 9604: 1, 9603: 1, 9602: 1, 960
1: 1, 9600: 1, 9599: 1, 9598: 1, 9597: 1, 9596: 1, 9595: 1, 9594: 1, 95
93: 1, 9592: 1, 9591: 1, 9590: 1, 9589: 1, 9588: 1, 9587: 1, 9586: 1, 9
585: 1, 9584: 1, 9583: 1, 9582: 1, 9581: 1, 9580: 1, 9579: 1, 9578: 1,
9577: 1, 9576: 1, 9575: 1, 9574: 1, 9573: 1, 9572: 1, 9571: 1, 9570:
1, 9569: 1, 9568: 1, 9567: 1, 9566: 1, 9565: 1, 9564: 1, 9563: 1, 956
2: 1, 9561: 1, 9560: 1, 9559: 1, 9558: 1, 9557: 1, 9556: 1, 9555: 1, 95
54: 1, 9553: 1, 9552: 1, 9551: 1, 9550: 1, 9549: 1, 9548: 1, 9547: 1, 9
546: 1, 9545: 1, 9544: 1, 9543: 1, 9542: 1, 9541: 1, 9540: 1, 9539: 1,
9538: 1, 9537: 1, 9536: 1, 9535: 1, 9534: 1, 9533: 1, 9532: 1, 9531:
1, 9530: 1, 9529: 1, 9528: 1, 9527: 1, 9526: 1, 9525: 1, 9524: 1, 952
3: 1, 9522: 1, 9521: 1, 9520: 1, 9519: 1, 9518: 1, 9517: 1, 9516: 1, 95
15: 1, 9514: 1, 9513: 1, 9512: 1, 9511: 1, 9510: 1, 9509: 1, 9508: 1, 9
507: 1, 9506: 1, 9505: 1, 9504: 1, 9503: 1, 9502: 1, 9501: 1, 9500: 1,
9499: 1, 9498: 1, 9497: 1, 9496: 1, 9495: 1, 9494: 1, 9493: 1, 9492:
1, 9491: 1, 9490: 1, 9489: 1, 9488: 1, 9487: 1, 9486: 1, 9485: 1, 948
4: 1, 9483: 1, 9482: 1, 9481: 1, 9480: 1, 9479: 1, 9478: 1, 9477: 1, 94
76: 1, 9475: 1, 9474: 1, 9473: 1, 9472: 1, 9471: 1, 9470: 1, 9469: 1, 9

Create PDF in your applications with the Pdfcrowd [HTML to PDF API](#)

Create PDF in your applications with the Pdfcrowd [HTML to PDF API](#)

0: 1, 8859: 1, 8858: 1, 8857: 1, 8856: 1, 8855: 1, 8854: 1, 8853: 1, 8852: 1, 8851: 1, 8850: 1, 8849: 1, 8848: 1, 8847: 1, 8846: 1, 8845: 1, 8844: 1, 8843: 1, 8842: 1, 8841: 1, 8840: 1, 8839: 1, 8838: 1, 8837: 1, 8836: 1, 8835: 1, 8834: 1, 8833: 1, 8832: 1, 8831: 1, 8830: 1, 8829: 1, 8828: 1, 8827: 1, 8826: 1, 8825: 1, 8824: 1, 8823: 1, 8822: 1, 8821: 1, 8820: 1, 8819: 1, 8818: 1, 8817: 1, 8816: 1, 8815: 1, 8814: 1, 8813: 1, 8812: 1, 8811: 1, 8810: 1, 8809: 1, 8808: 1, 8807: 1, 8806: 1, 8805: 1, 8804: 1, 8803: 1, 8802: 1, 8801: 1, 8800: 1, 8799: 1, 8798: 1, 8797: 1, 8796: 1, 8795: 1, 8794: 1, 8793: 1, 8792: 1, 8791: 1, 8790: 1, 8789: 1, 8788: 1, 8787: 1, 8786: 1, 8785: 1, 8784: 1, 8783: 1, 8782: 1, 8781: 1, 8780: 1, 8779: 1, 8778: 1, 8777: 1, 8776: 1, 8775: 1, 8774: 1, 8773: 1, 8772: 1, 8771: 1, 8770: 1, 8769: 1, 8768: 1, 8767: 1, 8766: 1, 8765: 1, 8764: 1, 8763: 1, 8762: 1, 8761: 1, 8760: 1, 8759: 1, 8758: 1, 8757: 1, 8756: 1, 8755: 1, 8754: 1, 8753: 1, 8752: 1, 8751: 1, 8750: 1, 8749: 1, 8748: 1, 8747: 1, 8746: 1, 8745: 1, 8744: 1, 8743: 1, 8742: 1, 8741: 1, 8740: 1, 8739: 1, 8738: 1, 8737: 1, 8736: 1, 8735: 1, 8734: 1, 8733: 1, 8732: 1, 8731: 1, 8730: 1, 8729: 1, 8728: 1, 8727: 1, 8726: 1, 8725: 1, 8724: 1, 8723: 1, 8722: 1, 8721: 1, 8720: 1, 8719: 1, 8718: 1, 8717: 1, 8716: 1, 8715: 1, 8714: 1, 8713: 1, 8712: 1, 8711: 1, 8710: 1, 8709: 1, 8708: 1, 8707: 1, 8706: 1, 8705: 1, 8704: 1, 8703: 1, 8702: 1, 8701: 1, 8700: 1, 8699: 1, 8698: 1, 8697: 1, 8696: 1, 8695: 1, 8694: 1, 8693: 1, 8692: 1, 8691: 1, 8690: 1, 8689: 1, 8688: 1, 8687: 1, 8686: 1, 8685: 1, 8684: 1, 8683: 1, 8682: 1, 8681: 1, 8680: 1, 8679: 1, 8678: 1, 8677: 1, 8676: 1, 8675: 1, 8674: 1, 8673: 1, 8672: 1, 8671: 1, 8670: 1, 8669: 1, 8668: 1, 8667: 1, 8666: 1, 8665: 1, 8664: 1, 8663: 1, 8662: 1, 8661: 1, 8660: 1, 8659: 1, 8658: 1, 8657: 1, 8656: 1, 8655: 1, 8654: 1, 8653: 1, 8652: 1, 8651: 1, 8650: 1, 8649: 1, 8648: 1, 8647: 1, 8646: 1, 8645: 1, 8644: 1, 8643: 1, 8642: 1, 8641: 1, 8640: 1, 8639: 1, 8638: 1, 8637: 1, 8636: 1, 8635: 1, 8634: 1, 8633: 1, 8632: 1, 8631: 1, 8630: 1, 8629: 1, 8628: 1, 8627: 1, 8626: 1, 8625: 1, 8624: 1, 8623: 1, 8622: 1, 8621: 1, 8620: 1, 8619: 1, 8618: 1, 8617: 1, 8616: 1, 8615: 1, 8614: 1, 8613: 1, 8612: 1, 8611: 1, 8610: 1, 8609: 1, 8608: 1, 8607: 1, 8606: 1, 8605: 1, 8604: 1, 8603: 1, 8602: 1, 8601: 1, 8600: 1, 8599: 1, 8598: 1, 8597: 1, 8596: 1, 8595: 1, 8594: 1, 8593: 1, 8592: 1, 8591: 1, 8590: 1, 8589: 1, 8588: 1, 8587: 1, 8586: 1, 8585: 1, 8584: 1, 8583: 1, 8582: 1, 8581: 1, 8580: 1, 8579: 1, 8578: 1, 8577: 1, 8576: 1, 8575: 1, 8574: 1, 8573: 1, 8572: 1, 8571: 1, 8570: 1, 8569: 1, 8568: 1, 8567: 1, 8566: 1, 8565: 1, 8564: 1, 8563: 1, 8562: 1, 8561: 1, 8560: 1, 8559: 1, 8558: 1, 8557: 1, 8556:

1, 8555: 1, 8554: 1, 8553: 1, 8552: 1, 8551: 1, 8550: 1, 8549: 1, 8548: 1, 8547: 1, 8546: 1, 8545: 1, 8544: 1, 8543: 1, 8542: 1, 8541: 1, 8540: 1, 8539: 1, 8538: 1, 8537: 1, 8536: 1, 8535: 1, 8534: 1, 8533: 1, 8532: 1, 8531: 1, 8530: 1, 8529: 1, 8528: 1, 8527: 1, 8526: 1, 8525: 1, 8524: 1, 8523: 1, 8522: 1, 8521: 1, 8520: 1, 8519: 1, 8518: 1, 8517: 1, 8516: 1, 8515: 1, 8514: 1, 8513: 1, 8512: 1, 8511: 1, 8510: 1, 8509: 1, 8508: 1, 8507: 1, 8506: 1, 8505: 1, 8504: 1, 8503: 1, 8502: 1, 8501: 1, 8500: 1, 8499: 1, 8498: 1, 8497: 1, 8496: 1, 8495: 1, 8494: 1, 8493: 1, 8492: 1, 8491: 1, 8490: 1, 8489: 1, 8488: 1, 8487: 1, 8486: 1, 8485: 1, 8484: 1, 8483: 1, 8482: 1, 8481: 1, 8480: 1, 8479: 1, 8478: 1, 8477: 1, 8476: 1, 8475: 1, 8474: 1, 8473: 1, 8472: 1, 8471: 1, 8470: 1, 8469: 1, 8468: 1, 8467: 1, 8466: 1, 8465: 1, 8464: 1, 8463: 1, 8462: 1, 8461: 1, 8460: 1, 8459: 1, 8458: 1, 8457: 1, 8456: 1, 8455: 1, 8454: 1, 8453: 1, 8452: 1, 8451: 1, 8450: 1, 8449: 1, 8448: 1, 8447: 1, 8446: 1, 8445: 1, 8444: 1, 8443: 1, 8442: 1, 8441: 1, 8440: 1, 8439: 1, 8438: 1, 8437: 1, 8436: 1, 8435: 1, 8434: 1, 8433: 1, 8432: 1, 8431: 1, 8430: 1, 8429: 1, 8428: 1, 8427: 1, 8426: 1, 8425: 1, 8424: 1, 8423: 1, 8422: 1, 8421: 1, 8420: 1, 8419: 1, 8418: 1, 8417: 1, 8416: 1, 8415: 1, 8414: 1, 8413: 1, 8412: 1, 8411: 1, 8410: 1, 8409: 1, 8408: 1, 8407: 1, 8406: 1, 8405: 1, 8404: 1, 8403: 1, 8402: 1, 8401: 1, 8400: 1, 8399: 1, 8398: 1, 8397: 1, 8396: 1, 8395: 1, 8394: 1, 8393: 1, 8392: 1, 8391: 1, 8390: 1, 8389: 1, 8388: 1, 8387: 1, 8386: 1, 8385: 1, 8384: 1, 8383: 1, 8382: 1, 8381: 1, 8380: 1, 8379: 1, 8378: 1, 8377: 1, 8376: 1, 8375: 1, 8374: 1, 8373: 1, 8372: 1, 8371: 1, 8370: 1, 8369: 1, 8368: 1, 8367: 1, 8366: 1, 8365: 1, 8364: 1, 8363: 1, 8362: 1, 8361: 1, 8360: 1, 8359: 1, 8358: 1, 8357: 1, 8356: 1, 8355: 1, 8354: 1, 8353: 1, 8352: 1, 8351: 1, 8350: 1, 8349: 1, 8348: 1, 8347: 1, 8346: 1, 8345: 1, 8344: 1, 8343: 1, 8342: 1, 8341: 1, 8340: 1, 8339: 1, 8338: 1, 8337: 1, 8336: 1, 8335: 1, 8334: 1, 8333: 1, 8332: 1, 8331: 1, 8330: 1, 8329: 1, 8328: 1, 8327: 1, 8326: 1, 8325: 1, 8324: 1, 8323: 1, 8322: 1, 8321: 1, 8320: 1, 8319: 1, 8318: 1, 8317: 1, 8316: 1, 8315: 1, 8314: 1, 8313: 1, 8312: 1, 8311: 1, 8310: 1, 8309: 1, 8308: 1, 8307: 1, 8306: 1, 8305: 1, 8304: 1, 8303: 1, 8302: 1, 8301: 1, 8300: 1, 8299: 1, 8298: 1, 8297: 1, 8296: 1, 8295: 1, 8294: 1, 8293: 1, 8292: 1, 8291: 1, 8290: 1, 8289: 1, 8288: 1, 8287: 1, 8286: 1, 8285: 1, 8284: 1, 8283: 1, 8282: 1, 8281: 1, 8280: 1, 8279: 1, 8278: 1, 8277: 1, 8276: 1, 8275: 1, 8274: 1, 8273: 1, 8272: 1, 8271: 1, 8270: 1, 8269: 1, 8268: 1, 8267: 1, 8266: 1, 8265: 1, 8264: 1, 8263: 1, 8262: 1, 8261: 1, 8260: 1, 8259: 1, 8258: 1, 8257: 1, 8256: 1, 8255: 1, 8254: 1, 8253: 1, 8252: 1,

8251: 1, 8250: 1, 8249: 1, 8248: 1, 8247: 1, 8246: 1, 8245: 1, 8244:
1, 8243: 1, 8242: 1, 8241: 1, 8240: 1, 8239: 1, 8238: 1, 8237: 1, 823
6: 1, 8235: 1, 8234: 1, 8233: 1, 8232: 1, 8231: 1, 8230: 1, 8229: 1, 82
28: 1, 8227: 1, 8226: 1, 8225: 1, 8224: 1, 8223: 1, 8222: 1, 8221: 1, 8
220: 1, 8219: 1, 8218: 1, 8217: 1, 8216: 1, 8215: 1, 8214: 1, 8213: 1,
8212: 1, 8211: 1, 8210: 1, 8209: 1, 8208: 1, 8207: 1, 8206: 1, 8205:
1, 8204: 1, 8203: 1, 8202: 1, 8201: 1, 8200: 1, 8199: 1, 8198: 1, 819
7: 1, 8196: 1, 8195: 1, 8194: 1, 8193: 1, 8192: 1, 8191: 1, 8190: 1, 81
89: 1, 8188: 1, 8187: 1, 8186: 1, 8185: 1, 8184: 1, 8183: 1, 8182: 1, 8
181: 1, 8180: 1, 8179: 1, 8178: 1, 8177: 1, 8176: 1, 8175: 1, 8174: 1,
8173: 1, 8172: 1, 8171: 1, 8170: 1, 8169: 1, 8168: 1, 8167: 1, 8166:
1, 8165: 1, 8164: 1, 8163: 1, 8162: 1, 8161: 1, 8160: 1, 8159: 1, 815
8: 1, 8157: 1, 8156: 1, 8155: 1, 8154: 1, 8153: 1, 8152: 1, 8151: 1, 81
50: 1, 8149: 1, 8148: 1, 8147: 1, 8146: 1, 8145: 1, 8144: 1, 8143: 1, 8
142: 1, 8141: 1, 8140: 1, 8139: 1, 8138: 1, 8137: 1, 8136: 1, 8135: 1,
8134: 1, 8133: 1, 8132: 1, 8131: 1, 8130: 1, 8129: 1, 8128: 1, 8127:
1, 8126: 1, 8125: 1, 8124: 1, 8123: 1, 8122: 1, 8121: 1, 8120: 1, 811
9: 1, 8118: 1, 8117: 1, 8116: 1, 8115: 1, 8114: 1, 8113: 1, 8112: 1, 81
11: 1, 8110: 1, 8109: 1, 8108: 1, 8107: 1, 8106: 1, 8105: 1, 8104: 1, 8
103: 1, 8102: 1, 8101: 1, 8100: 1, 8099: 1, 8098: 1, 8097: 1, 8096: 1,
8095: 1, 8094: 1, 8093: 1, 8092: 1, 8091: 1, 8090: 1, 8089: 1, 8088:
1, 8087: 1, 8086: 1, 8085: 1, 8084: 1, 8083: 1, 8082: 1, 8081: 1, 808
0: 1, 8079: 1, 8078: 1, 8077: 1, 8076: 1, 8075: 1, 8074: 1, 8073: 1, 80
72: 1, 8071: 1, 8070: 1, 8069: 1, 8068: 1, 8067: 1, 8066: 1, 8065: 1, 8
064: 1, 8063: 1, 8062: 1, 8061: 1, 8060: 1, 8059: 1, 8058: 1, 8057: 1,
8056: 1, 8055: 1, 8054: 1, 8053: 1, 8052: 1, 8051: 1, 8050: 1, 8049:
1, 8048: 1, 8047: 1, 8046: 1, 8045: 1, 8044: 1, 8043: 1, 8042: 1, 804
1: 1, 8040: 1, 8039: 1, 8038: 1, 8037: 1, 8036: 1, 8035: 1, 8034: 1, 80
33: 1, 8032: 1, 8031: 1, 8030: 1, 8029: 1, 8028: 1, 8027: 1, 8026: 1, 8
025: 1, 8024: 1, 8023: 1, 8022: 1, 8021: 1, 8020: 1, 8019: 1, 8018: 1,
8017: 1, 8016: 1, 8015: 1, 8014: 1, 8013: 1, 8012: 1, 8011: 1, 8010:
1, 8009: 1, 8008: 1, 8007: 1, 8006: 1, 8005: 1, 8004: 1, 8003: 1, 800
2: 1, 8001: 1, 8000: 1, 7999: 1, 7998: 1, 7997: 1, 7996: 1, 7995: 1, 79
94: 1, 7993: 1, 7992: 1, 7991: 1, 7990: 1, 7989: 1, 7988: 1, 7987: 1, 7
986: 1, 7985: 1, 7984: 1, 7983: 1, 7982: 1, 7981: 1, 7980: 1, 7979: 1,
7978: 1, 7977: 1, 7976: 1, 7975: 1, 7974: 1, 7973: 1, 7972: 1, 7971:
1, 7970: 1, 7969: 1, 7968: 1, 7967: 1, 7966: 1, 7965: 1, 7964: 1, 796
3: 1, 7962: 1, 7961: 1, 7960: 1, 7959: 1, 7958: 1, 7957: 1, 7956: 1, 79
55: 1, 7954: 1, 7953: 1, 7952: 1, 7951: 1, 7950: 1, 7949: 1, 7948: 1, 7

Create PDF in your applications with the Pdfcrowd [HTML to PDF API](#)

43: 1, 7642: 1, 7641: 1, 7640: 1, 7639: 1, 7638: 1, 7637: 1, 7636: 1, 7
635: 1, 7634: 1, 7633: 1, 7632: 1, 7631: 1, 7630: 1, 7629: 1, 7628: 1,
7627: 1, 7626: 1, 7625: 1, 7624: 1, 7623: 1, 7622: 1, 7621: 1, 7620:
1, 7619: 1, 7618: 1, 7617: 1, 7616: 1, 7615: 1, 7614: 1, 7613: 1, 761
2: 1, 7611: 1, 7610: 1, 7609: 1, 7608: 1, 7607: 1, 7606: 1, 7605: 1, 76
04: 1, 7603: 1, 7602: 1, 7601: 1, 7600: 1, 7599: 1, 7598: 1, 7597: 1, 7
596: 1, 7595: 1, 7594: 1, 7593: 1, 7592: 1, 7591: 1, 7590: 1, 7589: 1,
7588: 1, 7587: 1, 7586: 1, 7585: 1, 7584: 1, 7583: 1, 7582: 1, 7581:
1, 7580: 1, 7579: 1, 7578: 1, 7577: 1, 7576: 1, 7575: 1, 7574: 1, 757
3: 1, 7572: 1, 7571: 1, 7570: 1, 7569: 1, 7568: 1, 7567: 1, 7566: 1, 75
65: 1, 7564: 1, 7563: 1, 7562: 1, 7561: 1, 7560: 1, 7559: 1, 7558: 1, 7
557: 1, 7556: 1, 7555: 1, 7554: 1, 7553: 1, 7552: 1, 7551: 1, 7550: 1,
7549: 1, 7548: 1, 7547: 1, 7546: 1, 7545: 1, 7544: 1, 7543: 1, 7542:
1, 7541: 1, 7540: 1, 7539: 1, 7538: 1, 7537: 1, 7536: 1, 7535: 1, 753
4: 1, 7533: 1, 7532: 1, 7531: 1, 7530: 1, 7529: 1, 7528: 1, 7527: 1, 75
26: 1, 7525: 1, 7524: 1, 7523: 1, 7522: 1, 7521: 1, 7520: 1, 7519: 1, 7
518: 1, 7517: 1, 7516: 1, 7515: 1, 7514: 1, 7513: 1, 7512: 1, 7511: 1,
7510: 1, 7509: 1, 7508: 1, 7507: 1, 7506: 1, 7505: 1, 7504: 1, 7503:
1, 7502: 1, 7501: 1, 7500: 1, 7499: 1, 7498: 1, 7497: 1, 7496: 1, 749
5: 1, 7494: 1, 7493: 1, 7492: 1, 7491: 1, 7490: 1, 7489: 1, 7488: 1, 74
87: 1, 7486: 1, 7485: 1, 7484: 1, 7483: 1, 7482: 1, 7481: 1, 7480: 1, 7
479: 1, 7478: 1, 7477: 1, 7476: 1, 7475: 1, 7474: 1, 7473: 1, 7472: 1,
7471: 1, 7470: 1, 7469: 1, 7468: 1, 7467: 1, 7466: 1, 7465: 1, 7464:
1, 7463: 1, 7462: 1, 7461: 1, 7460: 1, 7459: 1, 7458: 1, 7457: 1, 745
6: 1, 7455: 1, 7454: 1, 7453: 1, 7452: 1, 7451: 1, 7450: 1, 7449: 1, 74
48: 1, 7447: 1, 7446: 1, 7445: 1, 7444: 1, 7443: 1, 7442: 1, 7441: 1, 7
440: 1, 7439: 1, 7438: 1, 7437: 1, 7436: 1, 7435: 1, 7434: 1, 7433: 1,
7432: 1, 7431: 1, 7430: 1, 7429: 1, 7428: 1, 7427: 1, 7426: 1, 7425:
1, 7424: 1, 7423: 1, 7422: 1, 7421: 1, 7420: 1, 7419: 1, 7418: 1, 741
7: 1, 7416: 1, 7415: 1, 7414: 1, 7413: 1, 7412: 1, 7411: 1, 7410: 1, 74
09: 1, 7408: 1, 7407: 1, 7406: 1, 7405: 1, 7404: 1, 7403: 1, 7402: 1, 7
401: 1, 7400: 1, 7399: 1, 7398: 1, 7397: 1, 7396: 1, 7395: 1, 7394: 1,
7393: 1, 7392: 1, 7391: 1, 7390: 1, 7389: 1, 7388: 1, 7387: 1, 7386:
1, 7385: 1, 7384: 1, 7383: 1, 7382: 1, 7381: 1, 7380: 1, 7379: 1, 737
8: 1, 7377: 1, 7376: 1, 7375: 1, 7374: 1, 7373: 1, 7372: 1, 7371: 1, 73
70: 1, 7369: 1, 7368: 1, 7367: 1, 7366: 1, 7365: 1, 7364: 1, 7363: 1, 7
362: 1, 7361: 1, 7360: 1, 7359: 1, 7358: 1, 7357: 1, 7356: 1, 7355: 1,
7354: 1, 7353: 1, 7352: 1, 7351: 1, 7350: 1, 7349: 1, 7348: 1, 7347:
1, 7346: 1, 7345: 1, 7344: 1, 7343: 1, 7342: 1, 7341: 1, 7340: 1, 733

9: 1, 7338: 1, 7337: 1, 7336: 1, 7335: 1, 7334: 1, 7333: 1, 7332: 1, 7331: 1, 7330: 1, 7329: 1, 7328: 1, 7327: 1, 7326: 1, 7325: 1, 7324: 1, 7323: 1, 7322: 1, 7321: 1, 7320: 1, 7319: 1, 7318: 1, 7317: 1, 7316: 1, 7315: 1, 7314: 1, 7313: 1, 7312: 1, 7311: 1, 7310: 1, 7309: 1, 7308: 1, 7307: 1, 7306: 1, 7305: 1, 7304: 1, 7303: 1, 7302: 1, 7301: 1, 7300: 1, 7299: 1, 7298: 1, 7297: 1, 7296: 1, 7295: 1, 7294: 1, 7293: 1, 7292: 1, 7291: 1, 7290: 1, 7289: 1, 7288: 1, 7287: 1, 7286: 1, 7285: 1, 7284: 1, 7283: 1, 7282: 1, 7281: 1, 7280: 1, 7279: 1, 7278: 1, 7277: 1, 7276: 1, 7275: 1, 7274: 1, 7273: 1, 7272: 1, 7271: 1, 7270: 1, 7269: 1, 7268: 1, 7267: 1, 7266: 1, 7265: 1, 7264: 1, 7263: 1, 7262: 1, 7261: 1, 7260: 1, 7259: 1, 7258: 1, 7257: 1, 7256: 1, 7255: 1, 7254: 1, 7253: 1, 7252: 1, 7251: 1, 7250: 1, 7249: 1, 7248: 1, 7247: 1, 7246: 1, 7245: 1, 7244: 1, 7243: 1, 7242: 1, 7241: 1, 7240: 1, 7239: 1, 7238: 1, 7237: 1, 7236: 1, 7235: 1, 7234: 1, 7233: 1, 7232: 1, 7231: 1, 7230: 1, 7229: 1, 7228: 1, 7227: 1, 7226: 1, 7225: 1, 7224: 1, 7223: 1, 7222: 1, 7221: 1, 7220: 1, 7219: 1, 7218: 1, 7217: 1, 7216: 1, 7215: 1, 7214: 1, 7213: 1, 7212: 1, 7211: 1, 7210: 1, 7209: 1, 7208: 1, 7207: 1, 7206: 1, 7205: 1, 7204: 1, 7203: 1, 7202: 1, 7201: 1, 7200: 1, 7199: 1, 7198: 1, 7197: 1, 7196: 1, 7195: 1, 7194: 1, 7193: 1, 7192: 1, 7191: 1, 7190: 1, 7189: 1, 7188: 1, 7187: 1, 7186: 1, 7185: 1, 7184: 1, 7183: 1, 7182: 1, 7181: 1, 7180: 1, 7179: 1, 7178: 1, 7177: 1, 7176: 1, 7175: 1, 7174: 1, 7173: 1, 7172: 1, 7171: 1, 7170: 1, 7169: 1, 7168: 1, 7167: 1, 7166: 1, 7165: 1, 7164: 1, 7163: 1, 7162: 1, 7161: 1, 7160: 1, 7159: 1, 7158: 1, 7157: 1, 7156: 1, 7155: 1, 7154: 1, 7153: 1, 7152: 1, 7151: 1, 7150: 1, 7149: 1, 7148: 1, 7147: 1, 7146: 1, 7145: 1, 7144: 1, 7143: 1, 7142: 1, 7141: 1, 7140: 1, 7139: 1, 7138: 1, 7137: 1, 7136: 1, 7135: 1, 7134: 1, 7133: 1, 7132: 1, 7131: 1, 7130: 1, 7129: 1, 7128: 1, 7127: 1, 7126: 1, 7125: 1, 7124: 1, 7123: 1, 7122: 1, 7121: 1, 7120: 1, 7119: 1, 7118: 1, 7117: 1, 7116: 1, 7115: 1, 7114: 1, 7113: 1, 7112: 1, 7111: 1, 7110: 1, 7109: 1, 7108: 1, 7107: 1, 7106: 1, 7105: 1, 7104: 1, 7103: 1, 7102: 1, 7101: 1, 7100: 1, 7099: 1, 7098: 1, 7097: 1, 7096: 1, 7095: 1, 7094: 1, 7093: 1, 7092: 1, 7091: 1, 7090: 1, 7089: 1, 7088: 1, 7087: 1, 7086: 1, 7085: 1, 7084: 1, 7083: 1, 7082: 1, 7081: 1, 7080: 1, 7079: 1, 7078: 1, 7077: 1, 7076: 1, 7075: 1, 7074: 1, 7073: 1, 7072: 1, 7071: 1, 7070: 1, 7069: 1, 7068: 1, 7067: 1, 7066: 1, 7065: 1, 7064: 1, 7063: 1, 7062: 1, 7061: 1, 7060: 1, 7059: 1, 7058: 1, 7057: 1, 7056: 1, 7055: 1, 7054: 1, 7053: 1, 7052: 1, 7051: 1, 7050: 1, 7049: 1, 7048: 1, 7047: 1, 7046: 1, 7045: 1, 7044: 1, 7043: 1, 7042: 1, 7041: 1, 7040: 1, 7039: 1, 7038: 1, 7037: 1, 7036: 1, 7035:

1, 7034: 1, 7033: 1, 7032: 1, 7031: 1, 7030: 1, 7029: 1, 7028: 1, 7027: 1, 7026: 1, 7025: 1, 7024: 1, 7023: 1, 7022: 1, 7021: 1, 7020: 1, 7019: 1, 7018: 1, 7017: 1, 7016: 1, 7015: 1, 7014: 1, 7013: 1, 7012: 1, 7011: 1, 7010: 1, 7009: 1, 7008: 1, 7007: 1, 7006: 1, 7005: 1, 7004: 1, 7003: 1, 7002: 1, 7001: 1, 7000: 1, 6999: 1, 6998: 1, 6997: 1, 6996: 1, 6995: 1, 6994: 1, 6993: 1, 6992: 1, 6991: 1, 6990: 1, 6989: 1, 6988: 1, 6987: 1, 6986: 1, 6985: 1, 6984: 1, 6983: 1, 6982: 1, 6981: 1, 6980: 1, 6979: 1, 6978: 1, 6977: 1, 6976: 1, 6975: 1, 6974: 1, 6973: 1, 6972: 1, 6971: 1, 6970: 1, 6969: 1, 6968: 1, 6967: 1, 6966: 1, 6965: 1, 6964: 1, 6963: 1, 6962: 1, 6961: 1, 6960: 1, 6959: 1, 6958: 1, 6957: 1, 6956: 1, 6955: 1, 6954: 1, 6953: 1, 6952: 1, 6951: 1, 6950: 1, 6949: 1, 6948: 1, 6947: 1, 6946: 1, 6945: 1, 6944: 1, 6943: 1, 6942: 1, 6941: 1, 6940: 1, 6939: 1, 6938: 1, 6937: 1, 6936: 1, 6935: 1, 6934: 1, 6933: 1, 6932: 1, 6931: 1, 6930: 1, 6929: 1, 6928: 1, 6927: 1, 6926: 1, 6925: 1, 6924: 1, 6923: 1, 6922: 1, 6921: 1, 6920: 1, 6919: 1, 6918: 1, 6917: 1, 6916: 1, 6915: 1, 6914: 1, 6913: 1, 6912: 1, 6911: 1, 6910: 1, 6909: 1, 6908: 1, 6907: 1, 6906: 1, 6905: 1, 6904: 1, 6903: 1, 6902: 1, 6901: 1, 6900: 1, 6899: 1, 6898: 1, 6897: 1, 6896: 1, 6895: 1, 6894: 1, 6893: 1, 6892: 1, 6891: 1, 6890: 1, 6889: 1, 6888: 1, 6887: 1, 6886: 1, 6885: 1, 6884: 1, 6883: 1, 6882: 1, 6881: 1, 6880: 1, 6879: 1, 6878: 1, 6877: 1, 6876: 1, 6875: 1, 6874: 1, 6873: 1, 6872: 1, 6871: 1, 6870: 1, 6869: 1, 6868: 1, 6867: 1, 6866: 1, 6865: 1, 6864: 1, 6863: 1, 6862: 1, 6861: 1, 6860: 1, 6859: 1, 6858: 1, 6857: 1, 6856: 1, 6855: 1, 6854: 1, 6853: 1, 6852: 1, 6851: 1, 6850: 1, 6849: 1, 6848: 1, 6847: 1, 6846: 1, 6845: 1, 6844: 1, 6843: 1, 6842: 1, 6841: 1, 6840: 1, 6839: 1, 6838: 1, 6837: 1, 6836: 1, 6835: 1, 6834: 1, 6833: 1, 6832: 1, 6831: 1, 6830: 1, 6829: 1, 6828: 1, 6827: 1, 6826: 1, 6825: 1, 6824: 1, 6823: 1, 6822: 1, 6821: 1, 6820: 1, 6819: 1, 6818: 1, 6817: 1, 6816: 1, 6815: 1, 6814: 1, 6813: 1, 6812: 1, 6811: 1, 6810: 1, 6809: 1, 6808: 1, 6807: 1, 6806: 1, 6805: 1, 6804: 1, 6803: 1, 6802: 1, 6801: 1, 6800: 1, 6799: 1, 6798: 1, 6797: 1, 6796: 1, 6795: 1, 6794: 1, 6793: 1, 6792: 1, 6791: 1, 6790: 1, 6789: 1, 6788: 1, 6787: 1, 6786: 1, 6785: 1, 6784: 1, 6783: 1, 6782: 1, 6781: 1, 6780: 1, 6779: 1, 6778: 1, 6777: 1, 6776: 1, 6775: 1, 6774: 1, 6773: 1, 6772: 1, 6771: 1, 6770: 1, 6769: 1, 6768: 1, 6767: 1, 6766: 1, 6765: 1, 6764: 1, 6763: 1, 6762: 1, 6761: 1, 6760: 1, 6759: 1, 6758: 1, 6757: 1, 6756: 1, 6755: 1, 6754: 1, 6753: 1, 6752: 1, 6751: 1, 6750: 1, 6749: 1, 6748: 1, 6747: 1, 6746: 1, 6745: 1, 6744: 1, 6743: 1, 6742: 1, 6741: 1, 6740: 1, 6739: 1, 6738: 1, 6737: 1, 6736: 1, 6735: 1, 6734: 1, 6733: 1, 6732: 1, 6731: 1,

6730: 1, 6729: 1, 6728: 1, 6727: 1, 6726: 1, 6725: 1, 6724: 1, 6723:
1, 6722: 1, 6721: 1, 6720: 1, 6719: 1, 6718: 1, 6717: 1, 6716: 1, 671
5: 1, 6714: 1, 6713: 1, 6712: 1, 6711: 1, 6710: 1, 6709: 1, 6708: 1, 67
07: 1, 6706: 1, 6705: 1, 6704: 1, 6703: 1, 6702: 1, 6701: 1, 6700: 1, 6
699: 1, 6698: 1, 6697: 1, 6696: 1, 6695: 1, 6694: 1, 6693: 1, 6692: 1,
6691: 1, 6690: 1, 6689: 1, 6688: 1, 6687: 1, 6686: 1, 6685: 1, 6684:
1, 6683: 1, 6682: 1, 6681: 1, 6680: 1, 6679: 1, 6678: 1, 6677: 1, 667
6: 1, 6675: 1, 6674: 1, 6673: 1, 6672: 1, 6671: 1, 6670: 1, 6669: 1, 66
68: 1, 6667: 1, 6666: 1, 6665: 1, 6664: 1, 6663: 1, 6662: 1, 6661: 1, 6
660: 1, 6659: 1, 6658: 1, 6657: 1, 6656: 1, 6655: 1, 6654: 1, 6653: 1,
6652: 1, 6651: 1, 6650: 1, 6649: 1, 6648: 1, 6647: 1, 6646: 1, 6645:
1, 6644: 1, 6643: 1, 6642: 1, 6641: 1, 6640: 1, 6639: 1, 6638: 1, 663
7: 1, 6636: 1, 6635: 1, 6634: 1, 6633: 1, 6632: 1, 6631: 1, 6630: 1, 66
29: 1, 6628: 1, 6627: 1, 6626: 1, 6625: 1, 6624: 1, 6623: 1, 6622: 1, 6
621: 1, 6620: 1, 6619: 1, 6618: 1, 6617: 1, 6616: 1, 6615: 1, 6614: 1,
6613: 1, 6612: 1, 6611: 1, 6610: 1, 6609: 1, 6608: 1, 6607: 1, 6606:
1, 6605: 1, 6604: 1, 6603: 1, 6602: 1, 6601: 1, 6600: 1, 6599: 1, 659
8: 1, 6597: 1, 6596: 1, 6595: 1, 6594: 1, 6593: 1, 6592: 1, 6591: 1, 65
90: 1, 6589: 1, 6588: 1, 6587: 1, 6586: 1, 6585: 1, 6584: 1, 6583: 1, 6
582: 1, 6581: 1, 6580: 1, 6579: 1, 6578: 1, 6577: 1, 6576: 1, 6575: 1,
6574: 1, 6573: 1, 6572: 1, 6571: 1, 6570: 1, 6569: 1, 6568: 1, 6567:
1, 6566: 1, 6565: 1, 6564: 1, 6563: 1, 6562: 1, 6561: 1, 6560: 1, 655
9: 1, 6558: 1, 6557: 1, 6556: 1, 6555: 1, 6554: 1, 6553: 1, 6552: 1, 65
51: 1, 6550: 1, 6549: 1, 6548: 1, 6547: 1, 6546: 1, 6545: 1, 6544: 1, 6
543: 1, 6542: 1, 6541: 1, 6540: 1, 6539: 1, 6538: 1, 6537: 1, 6536: 1,
6535: 1, 6534: 1, 6533: 1, 6532: 1, 6531: 1, 6530: 1, 6529: 1, 6528:
1, 6527: 1, 6526: 1, 6525: 1, 6524: 1, 6523: 1, 6522: 1, 6521: 1, 652
0: 1, 6519: 1, 6518: 1, 6517: 1, 6516: 1, 6515: 1, 6514: 1, 6513: 1, 65
12: 1, 6511: 1, 6510: 1, 6509: 1, 6508: 1, 6507: 1, 6506: 1, 6505: 1, 6
504: 1, 6503: 1, 6502: 1, 6501: 1, 6500: 1, 6499: 1, 6498: 1, 6497: 1,
6496: 1, 6495: 1, 6494: 1, 6493: 1, 6492: 1, 6491: 1, 6490: 1, 6489:
1, 6488: 1, 6487: 1, 6486: 1, 6485: 1, 6484: 1, 6483: 1, 6482: 1, 648
1: 1, 6480: 1, 6479: 1, 6478: 1, 6477: 1, 6476: 1, 6475: 1, 6474: 1, 64
73: 1, 6472: 1, 6471: 1, 6470: 1, 6469: 1, 6468: 1, 6467: 1, 6466: 1, 6
465: 1, 6464: 1, 6463: 1, 6462: 1, 6461: 1, 6460: 1, 6459: 1, 6458: 1,
6457: 1, 6456: 1, 6455: 1, 6454: 1, 6453: 1, 6452: 1, 6451: 1, 6450:
1, 6449: 1, 6448: 1, 6447: 1, 6446: 1, 6445: 1, 6444: 1, 6443: 1, 644
2: 1, 6441: 1, 6440: 1, 6439: 1, 6438: 1, 6437: 1, 6436: 1, 6435: 1, 64
34: 1, 6433: 1, 6432: 1, 6431: 1, 6430: 1, 6429: 1, 6428: 1, 6427: 1, 6

Create PDF in your applications with the Pdfcrowd [HTML to PDF API](#)

Create PDF in your applications with the Pdfcrowd [HTML to PDF API](#)

8: 1, 5817: 1, 5816: 1, 5815: 1, 5814: 1, 5813: 1, 5812: 1, 5811: 1, 5810: 1, 5809: 1, 5808: 1, 5807: 1, 5806: 1, 5805: 1, 5804: 1, 5803: 1, 5802: 1, 5801: 1, 5800: 1, 5799: 1, 5798: 1, 5797: 1, 5796: 1, 5795: 1, 5794: 1, 5793: 1, 5792: 1, 5791: 1, 5790: 1, 5789: 1, 5788: 1, 5787: 1, 5786: 1, 5785: 1, 5784: 1, 5783: 1, 5782: 1, 5781: 1, 5780: 1, 5779: 1, 5778: 1, 5777: 1, 5776: 1, 5775: 1, 5774: 1, 5773: 1, 5772: 1, 5771: 1, 5770: 1, 5769: 1, 5768: 1, 5767: 1, 5766: 1, 5765: 1, 5764: 1, 5763: 1, 5762: 1, 5761: 1, 5760: 1, 5759: 1, 5758: 1, 5757: 1, 5756: 1, 5755: 1, 5754: 1, 5753: 1, 5752: 1, 5751: 1, 5750: 1, 5749: 1, 5748: 1, 5747: 1, 5746: 1, 5745: 1, 5744: 1, 5743: 1, 5742: 1, 5741: 1, 5740: 1, 5739: 1, 5738: 1, 5737: 1, 5736: 1, 5735: 1, 5734: 1, 5733: 1, 5732: 1, 5731: 1, 5730: 1, 5729: 1, 5728: 1, 5727: 1, 5726: 1, 5725: 1, 5724: 1, 5723: 1, 5722: 1, 5721: 1, 5720: 1, 5719: 1, 5718: 1, 5717: 1, 5716: 1, 5715: 1, 5714: 1, 5713: 1, 5712: 1, 5711: 1, 5710: 1, 5709: 1, 5708: 1, 5707: 1, 5706: 1, 5705: 1, 5704: 1, 5703: 1, 5702: 1, 5701: 1, 5700: 1, 5699: 1, 5698: 1, 5697: 1, 5696: 1, 5695: 1, 5694: 1, 5693: 1, 5692: 1, 5691: 1, 5690: 1, 5689: 1, 5688: 1, 5687: 1, 5686: 1, 5685: 1, 5684: 1, 5683: 1, 5682: 1, 5681: 1, 5680: 1, 5679: 1, 5678: 1, 5677: 1, 5676: 1, 5675: 1, 5674: 1, 5673: 1, 5672: 1, 5671: 1, 5670: 1, 5669: 1, 5668: 1, 5667: 1, 5666: 1, 5665: 1, 5664: 1, 5663: 1, 5662: 1, 5661: 1, 5660: 1, 5659: 1, 5658: 1, 5657: 1, 5656: 1, 5655: 1, 5654: 1, 5653: 1, 5652: 1, 5651: 1, 5650: 1, 5649: 1, 5648: 1, 5647: 1, 5646: 1, 5645: 1, 5644: 1, 5643: 1, 5642: 1, 5641: 1, 5640: 1, 5639: 1, 5638: 1, 5637: 1, 5636: 1, 5635: 1, 5634: 1, 5633: 1, 5632: 1, 5631: 1, 5630: 1, 5629: 1, 5628: 1, 5627: 1, 5626: 1, 5625: 1, 5624: 1, 5623: 1, 5622: 1, 5621: 1, 5620: 1, 5619: 1, 5618: 1, 5617: 1, 5616: 1, 5615: 1, 5614: 1, 5613: 1, 5612: 1, 5611: 1, 5610: 1, 5609: 1, 5608: 1, 5607: 1, 5606: 1, 5605: 1, 5604: 1, 5603: 1, 5602: 1, 5601: 1, 5600: 1, 5599: 1, 5598: 1, 5597: 1, 5596: 1, 5595: 1, 5594: 1, 5593: 1, 5592: 1, 5591: 1, 5590: 1, 5589: 1, 5588: 1, 5587: 1, 5586: 1, 5585: 1, 5584: 1, 5583: 1, 5582: 1, 5581: 1, 5580: 1, 5579: 1, 5578: 1, 5577: 1, 5576: 1, 5575: 1, 5574: 1, 5573: 1, 5572: 1, 5571: 1, 5570: 1, 5569: 1, 5568: 1, 5567: 1, 5566: 1, 5565: 1, 5564: 1, 5563: 1, 5562: 1, 5561: 1, 5560: 1, 5559: 1, 5558: 1, 5557: 1, 5556: 1, 5555: 1, 5554: 1, 5553: 1, 5552: 1, 5551: 1, 5550: 1, 5549: 1, 5548: 1, 5547: 1, 5546: 1, 5545: 1, 5544: 1, 5543: 1, 5542: 1, 5541: 1, 5540: 1, 5539: 1, 5538: 1, 5537: 1, 5536: 1, 5535: 1, 5534: 1, 5533: 1, 5532: 1, 5531: 1, 5530: 1, 5529: 1, 5528: 1, 5527: 1, 5526: 1, 5525: 1, 5524: 1, 5523: 1, 5522: 1, 5521: 1, 5520: 1, 5519: 1, 5518: 1, 5517: 1, 5516: 1, 5515: 1, 5514:

1, 5513: 1, 5512: 1, 5511: 1, 5510: 1, 5509: 1, 5508: 1, 5507: 1, 5506: 1, 5505: 1, 5504: 1, 5503: 1, 5502: 1, 5501: 1, 5500: 1, 5499: 1, 5498: 1, 5497: 1, 5496: 1, 5495: 1, 5494: 1, 5493: 1, 5492: 1, 5491: 1, 5490: 1, 5489: 1, 5488: 1, 5487: 1, 5486: 1, 5485: 1, 5484: 1, 5483: 1, 5482: 1, 5481: 1, 5480: 1, 5479: 1, 5478: 1, 5477: 1, 5476: 1, 5475: 1, 5474: 1, 5473: 1, 5472: 1, 5471: 1, 5470: 1, 5469: 1, 5468: 1, 5467: 1, 5466: 1, 5465: 1, 5464: 1, 5463: 1, 5462: 1, 5461: 1, 5460: 1, 5459: 1, 5458: 1, 5457: 1, 5456: 1, 5455: 1, 5454: 1, 5453: 1, 5452: 1, 5451: 1, 5450: 1, 5449: 1, 5448: 1, 5447: 1, 5446: 1, 5445: 1, 5444: 1, 5443: 1, 5442: 1, 5441: 1, 5440: 1, 5439: 1, 5438: 1, 5437: 1, 5436: 1, 5435: 1, 5434: 1, 5433: 1, 5432: 1, 5431: 1, 5430: 1, 5429: 1, 5428: 1, 5427: 1, 5426: 1, 5425: 1, 5424: 1, 5423: 1, 5422: 1, 5421: 1, 5420: 1, 5419: 1, 5418: 1, 5417: 1, 5416: 1, 5415: 1, 5414: 1, 5413: 1, 5412: 1, 5411: 1, 5410: 1, 5409: 1, 5408: 1, 5407: 1, 5406: 1, 5405: 1, 5404: 1, 5403: 1, 5402: 1, 5401: 1, 5400: 1, 5399: 1, 5398: 1, 5397: 1, 5396: 1, 5395: 1, 5394: 1, 5393: 1, 5392: 1, 5391: 1, 5390: 1, 5389: 1, 5388: 1, 5387: 1, 5386: 1, 5385: 1, 5384: 1, 5383: 1, 5382: 1, 5381: 1, 5380: 1, 5379: 1, 5378: 1, 5377: 1, 5376: 1, 5375: 1, 5374: 1, 5373: 1, 5372: 1, 5371: 1, 5370: 1, 5369: 1, 5368: 1, 5367: 1, 5366: 1, 5365: 1, 5364: 1, 5363: 1, 5362: 1, 5361: 1, 5360: 1, 5359: 1, 5358: 1, 5357: 1, 5356: 1, 5355: 1, 5354: 1, 5353: 1, 5352: 1, 5351: 1, 5350: 1, 5349: 1, 5348: 1, 5347: 1, 5346: 1, 5345: 1, 5344: 1, 5343: 1, 5342: 1, 5341: 1, 5340: 1, 5339: 1, 5338: 1, 5337: 1, 5336: 1, 5335: 1, 5334: 1, 5333: 1, 5332: 1, 5331: 1, 5330: 1, 5329: 1, 5328: 1, 5327: 1, 5326: 1, 5325: 1, 5324: 1, 5323: 1, 5322: 1, 5321: 1, 5320: 1, 5319: 1, 5318: 1, 5317: 1, 5316: 1, 5315: 1, 5314: 1, 5313: 1, 5312: 1, 5311: 1, 5310: 1, 5309: 1, 5308: 1, 5307: 1, 5306: 1, 5305: 1, 5304: 1, 5303: 1, 5302: 1, 5301: 1, 5300: 1, 5299: 1, 5298: 1, 5297: 1, 5296: 1, 5295: 1, 5294: 1, 5293: 1, 5292: 1, 5291: 1, 5290: 1, 5289: 1, 5288: 1, 5287: 1, 5286: 1, 5285: 1, 5284: 1, 5283: 1, 5282: 1, 5281: 1, 5280: 1, 5279: 1, 5278: 1, 5277: 1, 5276: 1, 5275: 1, 5274: 1, 5273: 1, 5272: 1, 5271: 1, 5270: 1, 5269: 1, 5268: 1, 5267: 1, 5266: 1, 5265: 1, 5264: 1, 5263: 1, 5262: 1, 5261: 1, 5260: 1, 5259: 1, 5258: 1, 5257: 1, 5256: 1, 5255: 1, 5254: 1, 5253: 1, 5252: 1, 5251: 1, 5250: 1, 5249: 1, 5248: 1, 5247: 1, 5246: 1, 5245: 1, 5244: 1, 5243: 1, 5242: 1, 5241: 1, 5240: 1, 5239: 1, 5238: 1, 5237: 1, 5236: 1, 5235: 1, 5234: 1, 5233: 1, 5232: 1, 5231: 1, 5230: 1, 5229: 1, 5228: 1, 5227: 1, 5226: 1, 5225: 1, 5224: 1, 5223: 1, 5222: 1, 5221: 1, 5220: 1, 5219: 1, 5218: 1, 5217: 1, 5216: 1, 5215: 1, 5214: 1, 5213: 1, 5212: 1, 5211: 1, 5210: 1,

5209: 1, 5208: 1, 5207: 1, 5206: 1, 5205: 1, 5204: 1, 5203: 1, 5202:
1, 5201: 1, 5200: 1, 5199: 1, 5198: 1, 5197: 1, 5196: 1, 5195: 1, 519
4: 1, 5193: 1, 5192: 1, 5191: 1, 5190: 1, 5189: 1, 5188: 1, 5187: 1, 51
86: 1, 5185: 1, 5184: 1, 5183: 1, 5182: 1, 5181: 1, 5180: 1, 5179: 1, 5
178: 1, 5177: 1, 5176: 1, 5175: 1, 5174: 1, 5173: 1, 5172: 1, 5171: 1,
5170: 1, 5169: 1, 5168: 1, 5167: 1, 5166: 1, 5165: 1, 5164: 1, 5163:
1, 5162: 1, 5161: 1, 5160: 1, 5159: 1, 5158: 1, 5157: 1, 5156: 1, 515
5: 1, 5154: 1, 5153: 1, 5152: 1, 5151: 1, 5150: 1, 5149: 1, 5148: 1, 51
47: 1, 5146: 1, 5145: 1, 5144: 1, 5143: 1, 5142: 1, 5141: 1, 5140: 1, 5
139: 1, 5138: 1, 5137: 1, 5136: 1, 5135: 1, 5134: 1, 5133: 1, 5132: 1,
5131: 1, 5130: 1, 5129: 1, 5128: 1, 5127: 1, 5126: 1, 5125: 1, 5124:
1, 5123: 1, 5122: 1, 5121: 1, 5120: 1, 5119: 1, 5118: 1, 5117: 1, 511
6: 1, 5115: 1, 5114: 1, 5113: 1, 5112: 1, 5111: 1, 5110: 1, 5109: 1, 51
08: 1, 5107: 1, 5106: 1, 5105: 1, 5104: 1, 5103: 1, 5102: 1, 5101: 1, 5
100: 1, 5099: 1, 5098: 1, 5097: 1, 5096: 1, 5095: 1, 5094: 1, 5093: 1,
5092: 1, 5091: 1, 5090: 1, 5089: 1, 5088: 1, 5087: 1, 5086: 1, 5085:
1, 5084: 1, 5083: 1, 5082: 1, 5081: 1, 5080: 1, 5079: 1, 5078: 1, 507
7: 1, 5076: 1, 5075: 1, 5074: 1, 5073: 1, 5072: 1, 5071: 1, 5070: 1, 50
69: 1, 5068: 1, 5067: 1, 5066: 1, 5065: 1, 5064: 1, 5063: 1, 5062: 1, 5
061: 1, 5060: 1, 5059: 1, 5058: 1, 5057: 1, 5056: 1, 5055: 1, 5054: 1,
5053: 1, 5052: 1, 5051: 1, 5050: 1, 5049: 1, 5048: 1, 5047: 1, 5046:
1, 5045: 1, 5044: 1, 5043: 1, 5042: 1, 5041: 1, 5040: 1, 5039: 1, 503
8: 1, 5037: 1, 5036: 1, 5035: 1, 5034: 1, 5033: 1, 5032: 1, 5031: 1, 50
30: 1, 5029: 1, 5028: 1, 5027: 1, 5026: 1, 5025: 1, 5024: 1, 5023: 1, 5
022: 1, 5021: 1, 5020: 1, 5019: 1, 5018: 1, 5017: 1, 5016: 1, 5015: 1,
5014: 1, 5013: 1, 5012: 1, 5011: 1, 5010: 1, 5009: 1, 5008: 1, 5007:
1, 5006: 1, 5005: 1, 5004: 1, 5003: 1, 5002: 1, 5001: 1, 5000: 1, 499
9: 1, 4998: 1, 4997: 1, 4996: 1, 4995: 1, 4994: 1, 4993: 1, 4992: 1, 49
91: 1, 4990: 1, 4989: 1, 4988: 1, 4987: 1, 4986: 1, 4985: 1, 4984: 1, 4
983: 1, 4982: 1, 4981: 1, 4980: 1, 4979: 1, 4978: 1, 4977: 1, 4976: 1,
4975: 1, 4974: 1, 4973: 1, 4972: 1, 4971: 1, 4970: 1, 4969: 1, 4968:
1, 4967: 1, 4966: 1, 4965: 1, 4964: 1, 4963: 1, 4962: 1, 4961: 1, 496
0: 1, 4959: 1, 4958: 1, 4957: 1, 4956: 1, 4955: 1, 4954: 1, 4953: 1, 49
52: 1, 4951: 1, 4950: 1, 4949: 1, 4948: 1, 4947: 1, 4946: 1, 4945: 1, 4
944: 1, 4943: 1, 4942: 1, 4941: 1, 4940: 1, 4939: 1, 4938: 1, 4937: 1,
4936: 1, 4935: 1, 4934: 1, 4933: 1, 4932: 1, 4931: 1, 4930: 1, 4929:
1, 4928: 1, 4927: 1, 4926: 1, 4925: 1, 4924: 1, 4923: 1, 4922: 1, 492
1: 1, 4920: 1, 4919: 1, 4918: 1, 4917: 1, 4916: 1, 4915: 1, 4914: 1, 49
13: 1, 4912: 1, 4911: 1, 4910: 1, 4909: 1, 4908: 1, 4907: 1, 4906: 1, 4

Create PDF in your applications with the Pdfcrowd [HTML to PDF API](#)

Create PDF in your applications with the Pdfcrowd [HTML to PDF API](#)

7: 1, 4296: 1, 4295: 1, 4294: 1, 4293: 1, 4292: 1, 4291: 1, 4290: 1, 4289: 1, 4288: 1, 4287: 1, 4286: 1, 4285: 1, 4284: 1, 4283: 1, 4282: 1, 4281: 1, 4280: 1, 4279: 1, 4278: 1, 4277: 1, 4276: 1, 4275: 1, 4274: 1, 4273: 1, 4272: 1, 4271: 1, 4270: 1, 4269: 1, 4268: 1, 4267: 1, 4266: 1, 4265: 1, 4264: 1, 4263: 1, 4262: 1, 4261: 1, 4260: 1, 4259: 1, 4258: 1, 4257: 1, 4256: 1, 4255: 1, 4254: 1, 4253: 1, 4252: 1, 4251: 1, 4250: 1, 4249: 1, 4248: 1, 4247: 1, 4246: 1, 4245: 1, 4244: 1, 4243: 1, 4242: 1, 4241: 1, 4240: 1, 4239: 1, 4238: 1, 4237: 1, 4236: 1, 4235: 1, 4234: 1, 4233: 1, 4232: 1, 4231: 1, 4230: 1, 4229: 1, 4228: 1, 4227: 1, 4226: 1, 4225: 1, 4224: 1, 4223: 1, 4222: 1, 4221: 1, 4220: 1, 4219: 1, 4218: 1, 4217: 1, 4216: 1, 4215: 1, 4214: 1, 4213: 1, 4212: 1, 4211: 1, 4210: 1, 4209: 1, 4208: 1, 4207: 1, 4206: 1, 4205: 1, 4204: 1, 4203: 1, 4202: 1, 4201: 1, 4200: 1, 4199: 1, 4198: 1, 4197: 1, 4196: 1, 4195: 1, 4194: 1, 4193: 1, 4192: 1, 4191: 1, 4190: 1, 4189: 1, 4188: 1, 4187: 1, 4186: 1, 4185: 1, 4184: 1, 4183: 1, 4182: 1, 4181: 1, 4180: 1, 4179: 1, 4178: 1, 4177: 1, 4176: 1, 4175: 1, 4174: 1, 4173: 1, 4172: 1, 4171: 1, 4170: 1, 4169: 1, 4168: 1, 4167: 1, 4166: 1, 4165: 1, 4164: 1, 4163: 1, 4162: 1, 4161: 1, 4160: 1, 4159: 1, 4158: 1, 4157: 1, 4156: 1, 4155: 1, 4154: 1, 4153: 1, 4152: 1, 4151: 1, 4150: 1, 4149: 1, 4148: 1, 4147: 1, 4146: 1, 4145: 1, 4144: 1, 4143: 1, 4142: 1, 4141: 1, 4140: 1, 4139: 1, 4138: 1, 4137: 1, 4136: 1, 4135: 1, 4134: 1, 4133: 1, 4132: 1, 4131: 1, 4130: 1, 4129: 1, 4128: 1, 4127: 1, 4126: 1, 4125: 1, 4124: 1, 4123: 1, 4122: 1, 4121: 1, 4120: 1, 4119: 1, 4118: 1, 4117: 1, 4116: 1, 4115: 1, 4114: 1, 4113: 1, 4112: 1, 4111: 1, 4110: 1, 4109: 1, 4108: 1, 4107: 1, 4106: 1, 4105: 1, 4104: 1, 4103: 1, 4102: 1, 4101: 1, 4100: 1, 4099: 1, 4098: 1, 4097: 1, 4096: 1, 4095: 1, 4094: 1, 4093: 1, 4092: 1, 4091: 1, 4090: 1, 4089: 1, 4088: 1, 4087: 1, 4086: 1, 4085: 1, 4084: 1, 4083: 1, 4082: 1, 4081: 1, 4080: 1, 4079: 1, 4078: 1, 4077: 1, 4076: 1, 4075: 1, 4074: 1, 4073: 1, 4072: 1, 4071: 1, 4070: 1, 4069: 1, 4068: 1, 4067: 1, 4066: 1, 4065: 1, 4064: 1, 4063: 1, 4062: 1, 4061: 1, 4060: 1, 4059: 1, 4058: 1, 4057: 1, 4056: 1, 4055: 1, 4054: 1, 4053: 1, 4052: 1, 4051: 1, 4050: 1, 4049: 1, 4048: 1, 4047: 1, 4046: 1, 4045: 1, 4044: 1, 4043: 1, 4042: 1, 4041: 1, 4040: 1, 4039: 1, 4038: 1, 4037: 1, 4036: 1, 4035: 1, 4034: 1, 4033: 1, 4032: 1, 4031: 1, 4030: 1, 4029: 1, 4028: 1, 4027: 1, 4026: 1, 4025: 1, 4024: 1, 4023: 1, 4022: 1, 4021: 1, 4020: 1, 4019: 1, 4018: 1, 4017: 1, 4016: 1, 4015: 1, 4014: 1, 4013: 1, 4012: 1, 4011: 1, 4010: 1, 4009: 1, 4008: 1, 4007: 1, 4006: 1, 4005: 1, 4004: 1, 4003: 1, 4002: 1, 4001: 1, 4000: 1, 3999: 1, 3998: 1, 3997: 1, 3996: 1, 3995: 1, 3994: 1, 3993:

1, 3992: 1, 3991: 1, 3990: 1, 3989: 1, 3988: 1, 3987: 1, 3986: 1, 3985: 1, 3984: 1, 3983: 1, 3982: 1, 3981: 1, 3980: 1, 3979: 1, 3978: 1, 3977: 1, 3976: 1, 3975: 1, 3974: 1, 3973: 1, 3972: 1, 3971: 1, 3970: 1, 3969: 1, 3968: 1, 3967: 1, 3966: 1, 3965: 1, 3964: 1, 3963: 1, 3962: 1, 3961: 1, 3960: 1, 3959: 1, 3958: 1, 3957: 1, 3956: 1, 3955: 1, 3954: 1, 3953: 1, 3952: 1, 3951: 1, 3950: 1, 3949: 1, 3948: 1, 3947: 1, 3946: 1, 3945: 1, 3944: 1, 3943: 1, 3942: 1, 3941: 1, 3940: 1, 3939: 1, 3938: 1, 3937: 1, 3936: 1, 3935: 1, 3934: 1, 3933: 1, 3932: 1, 3931: 1, 3930: 1, 3929: 1, 3928: 1, 3927: 1, 3926: 1, 3925: 1, 3924: 1, 3923: 1, 3922: 1, 3921: 1, 3920: 1, 3919: 1, 3918: 1, 3917: 1, 3916: 1, 3915: 1, 3914: 1, 3913: 1, 3912: 1, 3911: 1, 3910: 1, 3909: 1, 3908: 1, 3907: 1, 3906: 1, 3905: 1, 3904: 1, 3903: 1, 3902: 1, 3901: 1, 3900: 1, 3899: 1, 3898: 1, 3897: 1, 3896: 1, 3895: 1, 3894: 1, 3893: 1, 3892: 1, 3891: 1, 3890: 1, 3889: 1, 3888: 1, 3887: 1, 3886: 1, 3885: 1, 3884: 1, 3883: 1, 3882: 1, 3881: 1, 3880: 1, 3879: 1, 3878: 1, 3877: 1, 3876: 1, 3875: 1, 3874: 1, 3873: 1, 3872: 1, 3871: 1, 3870: 1, 3869: 1, 3868: 1, 3867: 1, 3866: 1, 3865: 1, 3864: 1, 3863: 1, 3862: 1, 3861: 1, 3860: 1, 3859: 1, 3858: 1, 3857: 1, 3856: 1, 3855: 1, 3854: 1, 3853: 1, 3852: 1, 3851: 1, 3850: 1, 3849: 1, 3848: 1, 3847: 1, 3846: 1, 3845: 1, 3844: 1, 3843: 1, 3842: 1, 3841: 1, 3840: 1, 3839: 1, 3838: 1, 3837: 1, 3836: 1, 3835: 1, 3834: 1, 3833: 1, 3832: 1, 3831: 1, 3830: 1, 3829: 1, 3828: 1, 3827: 1, 3826: 1, 3825: 1, 3824: 1, 3823: 1, 3822: 1, 3821: 1, 3820: 1, 3819: 1, 3818: 1, 3817: 1, 3816: 1, 3815: 1, 3814: 1, 3813: 1, 3812: 1, 3811: 1, 3810: 1, 3809: 1, 3808: 1, 3807: 1, 3806: 1, 3805: 1, 3804: 1, 3803: 1, 3802: 1, 3801: 1, 3800: 1, 3799: 1, 3798: 1, 3797: 1, 3796: 1, 3795: 1, 3794: 1, 3793: 1, 3792: 1, 3791: 1, 3790: 1, 3789: 1, 3788: 1, 3787: 1, 3786: 1, 3785: 1, 3784: 1, 3783: 1, 3782: 1, 3781: 1, 3780: 1, 3779: 1, 3778: 1, 3777: 1, 3776: 1, 3775: 1, 3774: 1, 3773: 1, 3772: 1, 3771: 1, 3770: 1, 3769: 1, 3768: 1, 3767: 1, 3766: 1, 3765: 1, 3764: 1, 3763: 1, 3762: 1, 3761: 1, 3760: 1, 3759: 1, 3758: 1, 3757: 1, 3756: 1, 3755: 1, 3754: 1, 3753: 1, 3752: 1, 3751: 1, 3750: 1, 3749: 1, 3748: 1, 3747: 1, 3746: 1, 3745: 1, 3744: 1, 3743: 1, 3742: 1, 3741: 1, 3740: 1, 3739: 1, 3738: 1, 3737: 1, 3736: 1, 3735: 1, 3734: 1, 3733: 1, 3732: 1, 3731: 1, 3730: 1, 3729: 1, 3728: 1, 3727: 1, 3726: 1, 3725: 1, 3724: 1, 3723: 1, 3722: 1, 3721: 1, 3720: 1, 3719: 1, 3718: 1, 3717: 1, 3716: 1, 3715: 1, 3714: 1, 3713: 1, 3712: 1, 3711: 1, 3710: 1, 3709: 1, 3708: 1, 3707: 1, 3706: 1, 3705: 1, 3704: 1, 3703: 1, 3702: 1, 3701: 1, 3700: 1, 3699: 1, 3698: 1, 3697: 1, 3696: 1, 3695: 1, 3694: 1, 3693: 1, 3692: 1, 3691: 1, 3690: 1, 3689: 1,

3688: 1, 3687: 1, 3686: 1, 3685: 1, 3684: 1, 3683: 1, 3682: 1, 3681:
1, 3680: 1, 3679: 1, 3678: 1, 3677: 1, 3676: 1, 3675: 1, 3674: 1, 367
3: 1, 3672: 1, 3671: 1, 3670: 1, 3669: 1, 3668: 1, 3667: 1, 3666: 1, 36
65: 1, 3664: 1, 3663: 1, 3662: 1, 3661: 1, 3660: 1, 3659: 1, 3658: 1, 3
657: 1, 3656: 1, 3655: 1, 3654: 1, 3653: 1, 3652: 1, 3651: 1, 3650: 1,
3649: 1, 3648: 1, 3647: 1, 3646: 1, 3645: 1, 3644: 1, 3643: 1, 3642:
1, 3641: 1, 3640: 1, 3639: 1, 3638: 1, 3637: 1, 3636: 1, 3635: 1, 363
4: 1, 3633: 1, 3632: 1, 3631: 1, 3630: 1, 3629: 1, 3628: 1, 3627: 1, 36
26: 1, 3625: 1, 3624: 1, 3623: 1, 3622: 1, 3621: 1, 3620: 1, 3619: 1, 3
618: 1, 3617: 1, 3616: 1, 3615: 1, 3614: 1, 3613: 1, 3612: 1, 3611: 1,
3610: 1, 3609: 1, 3608: 1, 3607: 1, 3606: 1, 3605: 1, 3604: 1, 3603:
1, 3602: 1, 3601: 1, 3600: 1, 3599: 1, 3598: 1, 3597: 1, 3596: 1, 359
5: 1, 3594: 1, 3593: 1, 3592: 1, 3591: 1, 3590: 1, 3589: 1, 3588: 1, 35
87: 1, 3586: 1, 3585: 1, 3584: 1, 3583: 1, 3582: 1, 3581: 1, 3580: 1, 3
579: 1, 3578: 1, 3577: 1, 3576: 1, 3575: 1, 3574: 1, 3573: 1, 3572: 1,
3571: 1, 3570: 1, 3569: 1, 3568: 1, 3567: 1, 3566: 1, 3565: 1, 3564:
1, 3563: 1, 3562: 1, 3561: 1, 3560: 1, 3559: 1, 3558: 1, 3557: 1, 355
6: 1, 3555: 1, 3554: 1, 3553: 1, 3552: 1, 3551: 1, 3550: 1, 3549: 1, 35
48: 1, 3547: 1, 3546: 1, 3545: 1, 3544: 1, 3543: 1, 3542: 1, 3541: 1, 3
540: 1, 3539: 1, 3538: 1, 3537: 1, 3536: 1, 3535: 1, 3534: 1, 3533: 1,
3532: 1, 3531: 1, 3530: 1, 3529: 1, 3528: 1, 3527: 1, 3526: 1, 3525:
1, 3524: 1, 3523: 1, 3522: 1, 3521: 1, 3520: 1, 3519: 1, 3518: 1, 351
7: 1, 3516: 1, 3515: 1, 3514: 1, 3513: 1, 3512: 1, 3511: 1, 3510: 1, 35
09: 1, 3508: 1, 3507: 1, 3506: 1, 3505: 1, 3504: 1, 3503: 1, 3502: 1, 3
501: 1, 3500: 1, 3499: 1, 3498: 1, 3497: 1, 3496: 1, 3495: 1, 3494: 1,
3493: 1, 3492: 1, 3491: 1, 3490: 1, 3489: 1, 3488: 1, 3487: 1, 3486:
1, 3485: 1, 3484: 1, 3483: 1, 3482: 1, 3481: 1, 3480: 1, 3479: 1, 347
8: 1, 3477: 1, 3476: 1, 3475: 1, 3474: 1, 3473: 1, 3472: 1, 3471: 1, 34
70: 1, 3469: 1, 3468: 1, 3467: 1, 3466: 1, 3465: 1, 3464: 1, 3463: 1, 3
462: 1, 3461: 1, 3460: 1, 3459: 1, 3458: 1, 3457: 1, 3456: 1, 3455: 1,
3454: 1, 3453: 1, 3452: 1, 3451: 1, 3450: 1, 3449: 1, 3448: 1, 3447:
1, 3446: 1, 3445: 1, 3444: 1, 3443: 1, 3442: 1, 3441: 1, 3440: 1, 343
9: 1, 3438: 1, 3437: 1, 3436: 1, 3435: 1, 3434: 1, 3433: 1, 3432: 1, 34
31: 1, 3430: 1, 3429: 1, 3428: 1, 3427: 1, 3426: 1, 3425: 1, 3424: 1, 3
423: 1, 3422: 1, 3421: 1, 3420: 1, 3419: 1, 3418: 1, 3417: 1, 3416: 1,
3415: 1, 3414: 1, 3413: 1, 3412: 1, 3411: 1, 3410: 1, 3409: 1, 3408:
1, 3407: 1, 3406: 1, 3405: 1, 3404: 1, 3403: 1, 3402: 1, 3401: 1, 340
0: 1, 3399: 1, 3398: 1, 3397: 1, 3396: 1, 3395: 1, 3394: 1, 3393: 1, 33
92: 1, 3391: 1, 3390: 1, 3389: 1, 3388: 1, 3387: 1, 3386: 1, 3385: 1, 3

384: 1, 3383: 1, 3382: 1, 3381: 1, 3380: 1, 3379: 1, 3378: 1, 3377: 1,
3376: 1, 3375: 1, 3374: 1, 3373: 1, 3372: 1, 3371: 1, 3370: 1, 3369:
1, 3368: 1, 3367: 1, 3366: 1, 3365: 1, 3364: 1, 3363: 1, 3362: 1, 336
1: 1, 3360: 1, 3359: 1, 3358: 1, 3357: 1, 3356: 1, 3355: 1, 3354: 1, 33
53: 1, 3352: 1, 3351: 1, 3350: 1, 3349: 1, 3348: 1, 3347: 1, 3346: 1, 3
345: 1, 3344: 1, 3343: 1, 3342: 1, 3341: 1, 3340: 1, 3339: 1, 3338: 1,
3337: 1, 3336: 1, 3335: 1, 3334: 1, 3333: 1, 3332: 1, 3331: 1, 3330:
1, 3329: 1, 3328: 1, 3327: 1, 3326: 1, 3325: 1, 3324: 1, 3323: 1, 332
2: 1, 3321: 1, 3320: 1, 3319: 1, 3318: 1, 3317: 1, 3316: 1, 3315: 1, 33
14: 1, 3313: 1, 3312: 1, 3311: 1, 3310: 1, 3309: 1, 3308: 1, 3307: 1, 3
306: 1, 3305: 1, 3304: 1, 3303: 1, 3302: 1, 3301: 1, 3300: 1, 3299: 1,
3298: 1, 3297: 1, 3296: 1, 3295: 1, 3294: 1, 3293: 1, 3292: 1, 3291:
1, 3290: 1, 3289: 1, 3288: 1, 3287: 1, 3286: 1, 3285: 1, 3284: 1, 328
3: 1, 3282: 1, 3281: 1, 3280: 1, 3279: 1, 3278: 1, 3277: 1, 3276: 1, 32
75: 1, 3274: 1, 3273: 1, 3272: 1, 3271: 1, 3270: 1, 3269: 1, 3268: 1, 3
267: 1, 3266: 1, 3265: 1, 3264: 1, 3263: 1, 3262: 1, 3261: 1, 3260: 1,
3259: 1, 3258: 1, 3257: 1, 3256: 1, 3255: 1, 3254: 1, 3253: 1, 3252:
1, 3251: 1, 3250: 1, 3249: 1, 3248: 1, 3247: 1, 3246: 1, 3245: 1, 324
4: 1, 3243: 1, 3242: 1, 3241: 1, 3240: 1, 3239: 1, 3238: 1, 3237: 1, 32
36: 1, 3235: 1, 3234: 1, 3233: 1, 3232: 1, 3231: 1, 3230: 1, 3229: 1, 3
228: 1, 3227: 1, 3226: 1, 3225: 1, 3224: 1, 3223: 1, 3222: 1, 3221: 1,
3220: 1, 3219: 1, 3218: 1, 3217: 1, 3216: 1, 3215: 1, 3214: 1, 3213:
1, 3212: 1, 3211: 1, 3210: 1, 3209: 1, 3208: 1, 3207: 1, 3206: 1, 320
5: 1, 3204: 1, 3203: 1, 3202: 1, 3201: 1, 3200: 1, 3199: 1, 3198: 1, 31
97: 1, 3196: 1, 3195: 1, 3194: 1, 3193: 1, 3192: 1, 3191: 1, 3190: 1, 3
189: 1, 3188: 1, 3187: 1, 3186: 1, 3185: 1, 3184: 1, 3183: 1, 3182: 1,
3181: 1, 3180: 1, 3179: 1, 3178: 1, 3177: 1, 3176: 1, 3175: 1, 3174:
1, 3173: 1, 3172: 1, 3171: 1, 3170: 1, 3169: 1, 3168: 1, 3167: 1, 316
6: 1, 3165: 1, 3164: 1, 3163: 1, 3162: 1, 3161: 1, 3160: 1, 3159: 1, 31
58: 1, 3157: 1, 3156: 1, 3155: 1, 3154: 1, 3153: 1, 3152: 1, 3151: 1, 3
150: 1, 3149: 1, 3148: 1, 3147: 1, 3146: 1, 3145: 1, 3144: 1, 3143: 1,
3142: 1, 3141: 1, 3140: 1, 3139: 1, 3138: 1, 3137: 1, 3136: 1, 3135:
1, 3134: 1, 3133: 1, 3132: 1, 3131: 1, 3130: 1, 3129: 1, 3128: 1, 312
7: 1, 3126: 1, 3125: 1, 3124: 1, 3123: 1, 3122: 1, 3121: 1, 3120: 1, 31
19: 1, 3118: 1, 3117: 1, 3116: 1, 3115: 1, 3114: 1, 3113: 1, 3112: 1, 3
111: 1, 3110: 1, 3109: 1, 3108: 1, 3107: 1, 3106: 1, 3105: 1, 3104: 1,
3103: 1, 3102: 1, 3101: 1, 3100: 1, 3099: 1, 3098: 1, 3097: 1, 3096:
1, 3095: 1, 3094: 1, 3093: 1, 3092: 1, 3091: 1, 3090: 1, 3089: 1, 308
8: 1, 3087: 1, 3086: 1, 3085: 1, 3084: 1, 3083: 1, 3082: 1, 3081: 1, 30

Create PDF in your applications with the Pdfcrowd [HTML to PDF API](#)

6: 1, 2775: 1, 2774: 1, 2773: 1, 2772: 1, 2771: 1, 2770: 1, 2769: 1, 2768: 1, 2767: 1, 2766: 1, 2765: 1, 2764: 1, 2763: 1, 2762: 1, 2761: 1, 2760: 1, 2759: 1, 2758: 1, 2757: 1, 2756: 1, 2755: 1, 2754: 1, 2753: 1, 2752: 1, 2751: 1, 2750: 1, 2749: 1, 2748: 1, 2747: 1, 2746: 1, 2745: 1, 2744: 1, 2743: 1, 2742: 1, 2741: 1, 2740: 1, 2739: 1, 2738: 1, 2737: 1, 2736: 1, 2735: 1, 2734: 1, 2733: 1, 2732: 1, 2731: 1, 2730: 1, 2729: 1, 2728: 1, 2727: 1, 2726: 1, 2725: 1, 2724: 1, 2723: 1, 2722: 1, 2721: 1, 2720: 1, 2719: 1, 2718: 1, 2717: 1, 2716: 1, 2715: 1, 2714: 1, 2713: 1, 2712: 1, 2711: 1, 2710: 1, 2709: 1, 2708: 1, 2707: 1, 2706: 1, 2705: 1, 2704: 1, 2703: 1, 2702: 1, 2701: 1, 2700: 1, 2699: 1, 2698: 1, 2697: 1, 2696: 1, 2695: 1, 2694: 1, 2693: 1, 2692: 1, 2691: 1, 2690: 1, 2689: 1, 2688: 1, 2687: 1, 2686: 1, 2685: 1, 2684: 1, 2683: 1, 2682: 1, 2681: 1, 2680: 1, 2679: 1, 2678: 1, 2677: 1, 2676: 1, 2675: 1, 2674: 1, 2673: 1, 2672: 1, 2671: 1, 2670: 1, 2669: 1, 2668: 1, 2667: 1, 2666: 1, 2665: 1, 2664: 1, 2663: 1, 2662: 1, 2661: 1, 2660: 1, 2659: 1, 2658: 1, 2657: 1, 2656: 1, 2655: 1, 2654: 1, 2653: 1, 2652: 1, 2651: 1, 2650: 1, 2649: 1, 2648: 1, 2647: 1, 2646: 1, 2645: 1, 2644: 1, 2643: 1, 2642: 1, 2641: 1, 2640: 1, 2639: 1, 2638: 1, 2637: 1, 2636: 1, 2635: 1, 2634: 1, 2633: 1, 2632: 1, 2631: 1, 2630: 1, 2629: 1, 2628: 1, 2627: 1, 2626: 1, 2625: 1, 2624: 1, 2623: 1, 2622: 1, 2621: 1, 2620: 1, 2619: 1, 2618: 1, 2617: 1, 2616: 1, 2615: 1, 2614: 1, 2613: 1, 2612: 1, 2611: 1, 2610: 1, 2609: 1, 2608: 1, 2607: 1, 2606: 1, 2605: 1, 2604: 1, 2603: 1, 2602: 1, 2601: 1, 2600: 1, 2599: 1, 2598: 1, 2597: 1, 2596: 1, 2595: 1, 2594: 1, 2593: 1, 2592: 1, 2591: 1, 2590: 1, 2589: 1, 2588: 1, 2587: 1, 2586: 1, 2585: 1, 2584: 1, 2583: 1, 2582: 1, 2581: 1, 2580: 1, 2579: 1, 2578: 1, 2577: 1, 2576: 1, 2575: 1, 2574: 1, 2573: 1, 2572: 1, 2571: 1, 2570: 1, 2569: 1, 2568: 1, 2567: 1, 2566: 1, 2565: 1, 2564: 1, 2563: 1, 2562: 1, 2561: 1, 2560: 1, 2559: 1, 2558: 1, 2557: 1, 2556: 1, 2555: 1, 2554: 1, 2553: 1, 2552: 1, 2551: 1, 2550: 1, 2549: 1, 2548: 1, 2547: 1, 2546: 1, 2545: 1, 2544: 1, 2543: 1, 2542: 1, 2541: 1, 2540: 1, 2539: 1, 2538: 1, 2537: 1, 2536: 1, 2535: 1, 2534: 1, 2533: 1, 2532: 1, 2531: 1, 2530: 1, 2529: 1, 2528: 1, 2527: 1, 2526: 1, 2525: 1, 2524: 1, 2523: 1, 2522: 1, 2521: 1, 2520: 1, 2519: 1, 2518: 1, 2517: 1, 2516: 1, 2515: 1, 2514: 1, 2513: 1, 2512: 1, 2511: 1, 2510: 1, 2509: 1, 2508: 1, 2507: 1, 2506: 1, 2505: 1, 2504: 1, 2503: 1, 2502: 1, 2501: 1, 2500: 1, 2499: 1, 2498: 1, 2497: 1, 2496: 1, 2495: 1, 2494: 1, 2493: 1, 2492: 1, 2491: 1, 2490: 1, 2489: 1, 2488: 1, 2487: 1, 2486: 1, 2485: 1, 2484: 1, 2483: 1, 2482: 1, 2481: 1, 2480: 1, 2479: 1, 2478: 1, 2477: 1, 2476: 1, 2475: 1, 2474: 1, 2473: 1, 2472:

1, 2471: 1, 2470: 1, 2469: 1, 2468: 1, 2467: 1, 2466: 1, 2465: 1, 2464: 1, 2463: 1, 2462: 1, 2461: 1, 2460: 1, 2459: 1, 2458: 1, 2457: 1, 2456: 1, 2455: 1, 2454: 1, 2453: 1, 2452: 1, 2451: 1, 2450: 1, 2449: 1, 2448: 1, 2447: 1, 2446: 1, 2445: 1, 2444: 1, 2443: 1, 2442: 1, 2441: 1, 2440: 1, 2439: 1, 2438: 1, 2437: 1, 2436: 1, 2435: 1, 2434: 1, 2433: 1, 2432: 1, 2431: 1, 2430: 1, 2429: 1, 2428: 1, 2427: 1, 2426: 1, 2425: 1, 2424: 1, 2423: 1, 2422: 1, 2421: 1, 2420: 1, 2419: 1, 2418: 1, 2417: 1, 2416: 1, 2415: 1, 2414: 1, 2413: 1, 2412: 1, 2411: 1, 2410: 1, 2409: 1, 2408: 1, 2407: 1, 2406: 1, 2405: 1, 2404: 1, 2403: 1, 2402: 1, 2401: 1, 2400: 1, 2399: 1, 2398: 1, 2397: 1, 2396: 1, 2395: 1, 2394: 1, 2393: 1, 2392: 1, 2391: 1, 2390: 1, 2389: 1, 2388: 1, 2387: 1, 2386: 1, 2385: 1, 2384: 1, 2383: 1, 2382: 1, 2381: 1, 2380: 1, 2379: 1, 2378: 1, 2377: 1, 2376: 1, 2375: 1, 2374: 1, 2373: 1, 2372: 1, 2371: 1, 2370: 1, 2369: 1, 2368: 1, 2367: 1, 2366: 1, 2365: 1, 2364: 1, 2363: 1, 2362: 1, 2361: 1, 2360: 1, 2359: 1, 2358: 1, 2357: 1, 2356: 1, 2355: 1, 2354: 1, 2353: 1, 2352: 1, 2351: 1, 2350: 1, 2349: 1, 2348: 1, 2347: 1, 2346: 1, 2345: 1, 2344: 1, 2343: 1, 2342: 1, 2341: 1, 2340: 1, 2339: 1, 2338: 1, 2337: 1, 2336: 1, 2335: 1, 2334: 1, 2333: 1, 2332: 1, 2331: 1, 2330: 1, 2329: 1, 2328: 1, 2327: 1, 2326: 1, 2325: 1, 2324: 1, 2323: 1, 2322: 1, 2321: 1, 2320: 1, 2319: 1, 2318: 1, 2317: 1, 2316: 1, 2315: 1, 2314: 1, 2313: 1, 2312: 1, 2311: 1, 2310: 1, 2309: 1, 2308: 1, 2307: 1, 2306: 1, 2305: 1, 2304: 1, 2303: 1, 2302: 1, 2301: 1, 2300: 1, 2299: 1, 2298: 1, 2297: 1, 2296: 1, 2295: 1, 2294: 1, 2293: 1, 2292: 1, 2291: 1, 2290: 1, 2289: 1, 2288: 1, 2287: 1, 2286: 1, 2285: 1, 2284: 1, 2283: 1, 2282: 1, 2281: 1, 2280: 1, 2279: 1, 2278: 1, 2277: 1, 2276: 1, 2275: 1, 2274: 1, 2273: 1, 2272: 1, 2271: 1, 2270: 1, 2269: 1, 2268: 1, 2267: 1, 2266: 1, 2265: 1, 2264: 1, 2263: 1, 2262: 1, 2261: 1, 2260: 1, 2259: 1, 2258: 1, 2257: 1, 2256: 1, 2255: 1, 2254: 1, 2253: 1, 2252: 1, 2251: 1, 2250: 1, 2249: 1, 2248: 1, 2247: 1, 2246: 1, 2245: 1, 2244: 1, 2243: 1, 2242: 1, 2241: 1, 2240: 1, 2239: 1, 2238: 1, 2237: 1, 2236: 1, 2235: 1, 2234: 1, 2233: 1, 2232: 1, 2231: 1, 2230: 1, 2229: 1, 2228: 1, 2227: 1, 2226: 1, 2225: 1, 2224: 1, 2223: 1, 2222: 1, 2221: 1, 2220: 1, 2219: 1, 2218: 1, 2217: 1, 2216: 1, 2215: 1, 2214: 1, 2213: 1, 2212: 1, 2211: 1, 2210: 1, 2209: 1, 2208: 1, 2207: 1, 2206: 1, 2205: 1, 2204: 1, 2203: 1, 2202: 1, 2201: 1, 2200: 1, 2199: 1, 2198: 1, 2197: 1, 2196: 1, 2195: 1, 2194: 1, 2193: 1, 2192: 1, 2191: 1, 2190: 1, 2189: 1, 2188: 1, 2187: 1, 2186: 1, 2185: 1, 2184: 1, 2183: 1, 2182: 1, 2181: 1, 2180: 1, 2179: 1, 2178: 1, 2177: 1, 2176: 1, 2175: 1, 2174: 1, 2173: 1, 2172: 1, 2171: 1, 2170: 1, 2169: 1, 2168: 1,

2167: 1, 2166: 1, 2165: 1, 2164: 1, 2163: 1, 2162: 1, 2161: 1, 2160: 1, 2159: 1, 2158: 1, 2157: 1, 2156: 1, 2155: 1, 2154: 1, 2153: 1, 2152: 1, 2151: 1, 2150: 1, 2149: 1, 2148: 1, 2147: 1, 2146: 1, 2145: 1, 2144: 1, 2143: 1, 2142: 1, 2141: 1, 2140: 1, 2139: 1, 2138: 1, 2137: 1, 2136: 1, 2135: 1, 2134: 1, 2133: 1, 2132: 1, 2131: 1, 2130: 1, 2129: 1, 2128: 1, 2127: 1, 2126: 1, 2125: 1, 2124: 1, 2123: 1, 2122: 1, 2121: 1, 2120: 1, 2119: 1, 2118: 1, 2117: 1, 2116: 1, 2115: 1, 2114: 1, 2113: 1, 2112: 1, 2111: 1, 2110: 1, 2109: 1, 2108: 1, 2107: 1, 2106: 1, 2105: 1, 2104: 1, 2103: 1, 2102: 1, 2101: 1, 2100: 1, 2099: 1, 2098: 1, 2097: 1, 2096: 1, 2095: 1, 2094: 1, 2093: 1, 2092: 1, 2091: 1, 2090: 1, 2089: 1, 2088: 1, 2087: 1, 2086: 1, 2085: 1, 2084: 1, 2083: 1, 2082: 1, 2081: 1, 2080: 1, 2079: 1, 2078: 1, 2077: 1, 2076: 1, 2075: 1, 2074: 1, 2073: 1, 2072: 1, 2071: 1, 2070: 1, 2069: 1, 2068: 1, 2067: 1, 2066: 1, 2065: 1, 2064: 1, 2063: 1, 2062: 1, 2061: 1, 2060: 1, 2059: 1, 2058: 1, 2057: 1, 2056: 1, 2055: 1, 2054: 1, 2053: 1, 2052: 1, 2051: 1, 2050: 1, 2049: 1, 2048: 1, 2047: 1, 2046: 1, 2045: 1, 2044: 1, 2043: 1, 2042: 1, 2041: 1, 2040: 1, 2039: 1, 2038: 1, 2037: 1, 2036: 1, 2035: 1, 2034: 1, 2033: 1, 2032: 1, 2031: 1, 2030: 1, 2029: 1, 2028: 1, 2027: 1, 2026: 1, 2025: 1, 2024: 1, 2023: 1, 2022: 1, 2021: 1, 2020: 1, 2019: 1, 2018: 1, 2017: 1, 2016: 1, 2015: 1, 2014: 1, 2013: 1, 2012: 1, 2011: 1, 2010: 1, 2009: 1, 2008: 1, 2007: 1, 2006: 1, 2005: 1, 2004: 1, 2003: 1, 2002: 1, 2001: 1, 2000: 1, 1999: 1, 1998: 1, 1997: 1, 1996: 1, 1995: 1, 1994: 1, 1993: 1, 1992: 1, 1991: 1, 1990: 1, 1989: 1, 1988: 1, 1987: 1, 1986: 1, 1985: 1, 1984: 1, 1983: 1, 1982: 1, 1981: 1, 1980: 1, 1979: 1, 1978: 1, 1977: 1, 1976: 1, 1975: 1, 1974: 1, 1973: 1, 1972: 1, 1971: 1, 1970: 1, 1969: 1, 1968: 1, 1967: 1, 1966: 1, 1965: 1, 1964: 1, 1963: 1, 1962: 1, 1961: 1, 1960: 1, 1959: 1, 1958: 1, 1957: 1, 1956: 1, 1955: 1, 1954: 1, 1953: 1, 1952: 1, 1951: 1, 1950: 1, 1949: 1, 1948: 1, 1947: 1, 1946: 1, 1945: 1, 1944: 1, 1943: 1, 1942: 1, 1941: 1, 1940: 1, 1939: 1, 1938: 1, 1937: 1, 1936: 1, 1935: 1, 1934: 1, 1933: 1, 1932: 1, 1931: 1, 1930: 1, 1929: 1, 1928: 1, 1927: 1, 1926: 1, 1925: 1, 1924: 1, 1923: 1, 1922: 1, 1921: 1, 1920: 1, 1919: 1, 1918: 1, 1917: 1, 1916: 1, 1915: 1, 1914: 1, 1913: 1, 1912: 1, 1911: 1, 1910: 1, 1909: 1, 1908: 1, 1907: 1, 1906: 1, 1905: 1, 1904: 1, 1903: 1, 1902: 1, 1901: 1, 1900: 1, 1899: 1, 1898: 1, 1897: 1, 1896: 1, 1895: 1, 1894: 1, 1893: 1, 1892: 1, 1891: 1, 1890: 1, 1889: 1, 1888: 1, 1887: 1, 1886: 1, 1885: 1, 1884: 1, 1883: 1, 1882: 1, 1881: 1, 1880: 1, 1879: 1, 1878: 1, 1877: 1, 1876: 1, 1875: 1, 1874: 1, 1873: 1, 1872: 1, 1871: 1, 1870: 1, 1869: 1, 1868: 1, 1867: 1, 1866: 1, 1865: 1, 1864: 1, 1

Create PDF in your applications with the Pdfcrowd [HTML to PDF API](#)

59: 1, 1558: 1, 1557: 1, 1556: 1, 1555: 1, 1554: 1, 1553: 1, 1552: 1, 1
551: 1, 1550: 1, 1549: 1, 1548: 1, 1547: 1, 1546: 1, 1545: 1, 1544: 1,
1543: 1, 1542: 1, 1541: 1, 1540: 1, 1539: 1, 1538: 1, 1537: 1, 1536:
1, 1535: 1, 1534: 1, 1533: 1, 1532: 1, 1531: 1, 1530: 1, 1529: 1, 152
8: 1, 1527: 1, 1526: 1, 1525: 1, 1524: 1, 1523: 1, 1522: 1, 1521: 1, 15
20: 1, 1519: 1, 1518: 1, 1517: 1, 1516: 1, 1515: 1, 1514: 1, 1513: 1, 1
512: 1, 1511: 1, 1510: 1, 1509: 1, 1508: 1, 1507: 1, 1506: 1, 1505: 1,
1504: 1, 1503: 1, 1502: 1, 1501: 1, 1500: 1, 1499: 1, 1498: 1, 1497:
1, 1496: 1, 1495: 1, 1494: 1, 1493: 1, 1492: 1, 1491: 1, 1490: 1, 148
9: 1, 1488: 1, 1487: 1, 1486: 1, 1485: 1, 1484: 1, 1483: 1, 1482: 1, 14
81: 1, 1480: 1, 1479: 1, 1478: 1, 1477: 1, 1476: 1, 1475: 1, 1474: 1, 1
473: 1, 1472: 1, 1471: 1, 1470: 1, 1469: 1, 1468: 1, 1467: 1, 1466: 1,
1465: 1, 1464: 1, 1463: 1, 1462: 1, 1461: 1, 1460: 1, 1459: 1, 1458:
1, 1457: 1, 1456: 1, 1455: 1, 1454: 1, 1453: 1, 1452: 1, 1451: 1, 145
0: 1, 1449: 1, 1448: 1, 1447: 1, 1446: 1, 1445: 1, 1444: 1, 1443: 1, 14
42: 1, 1441: 1, 1440: 1, 1439: 1, 1438: 1, 1437: 1, 1436: 1, 1435: 1, 1
434: 1, 1433: 1, 1432: 1, 1431: 1, 1430: 1, 1429: 1, 1428: 1, 1427: 1,
1426: 1, 1425: 1, 1424: 1, 1423: 1, 1422: 1, 1421: 1, 1420: 1, 1419:
1, 1418: 1, 1417: 1, 1416: 1, 1415: 1, 1414: 1, 1413: 1, 1412: 1, 141
1: 1, 1410: 1, 1409: 1, 1408: 1, 1407: 1, 1406: 1, 1405: 1, 1404: 1, 14
03: 1, 1402: 1, 1401: 1, 1400: 1, 1399: 1, 1398: 1, 1397: 1, 1396: 1, 1
395: 1, 1394: 1, 1393: 1, 1392: 1, 1391: 1, 1390: 1, 1389: 1, 1388: 1,
1387: 1, 1386: 1, 1385: 1, 1384: 1, 1383: 1, 1382: 1, 1381: 1, 1380:
1, 1379: 1, 1378: 1, 1377: 1, 1376: 1, 1375: 1, 1374: 1, 1373: 1, 137
2: 1, 1371: 1, 1370: 1, 1369: 1, 1368: 1, 1367: 1, 1366: 1, 1365: 1, 13
64: 1, 1363: 1, 1362: 1, 1361: 1, 1360: 1, 1359: 1, 1358: 1, 1357: 1, 1
356: 1, 1355: 1, 1354: 1, 1353: 1, 1352: 1, 1351: 1, 1350: 1, 1349: 1,
1348: 1, 1347: 1, 1346: 1, 1345: 1, 1344: 1, 1343: 1, 1342: 1, 1341:
1, 1340: 1, 1339: 1, 1338: 1, 1337: 1, 1336: 1, 1335: 1, 1334: 1, 133
3: 1, 1332: 1, 1331: 1, 1330: 1, 1329: 1, 1328: 1, 1327: 1, 1326: 1, 13
25: 1, 1324: 1, 1323: 1, 1322: 1, 1321: 1, 1320: 1, 1319: 1, 1318: 1, 1
317: 1, 1316: 1, 1315: 1, 1314: 1, 1313: 1, 1312: 1, 1311: 1, 1310: 1,
1309: 1, 1308: 1, 1307: 1, 1306: 1, 1305: 1, 1304: 1, 1303: 1, 1302:
1, 1301: 1, 1300: 1, 1299: 1, 1298: 1, 1297: 1, 1296: 1, 1295: 1, 129
4: 1, 1293: 1, 1292: 1, 1291: 1, 1290: 1, 1289: 1, 1288: 1, 1287: 1, 12
86: 1, 1285: 1, 1284: 1, 1283: 1, 1282: 1, 1281: 1, 1280: 1, 1279: 1, 1
278: 1, 1277: 1, 1276: 1, 1275: 1, 1274: 1, 1273: 1, 1272: 1, 1271: 1,
1270: 1, 1269: 1, 1268: 1, 1267: 1, 1266: 1, 1265: 1, 1264: 1, 1263:
1, 1262: 1, 1261: 1, 1260: 1, 1259: 1, 1258: 1, 1257: 1, 1256: 1, 125

5: 1, 1254: 1, 1253: 1, 1252: 1, 1251: 1, 1250: 1, 1249: 1, 1248: 1, 1247: 1, 1246: 1, 1245: 1, 1244: 1, 1243: 1, 1242: 1, 1241: 1, 1240: 1, 1239: 1, 1238: 1, 1237: 1, 1236: 1, 1235: 1, 1234: 1, 1233: 1, 1232: 1, 1231: 1, 1230: 1, 1229: 1, 1228: 1, 1227: 1, 1226: 1, 1225: 1, 1224: 1, 1223: 1, 1222: 1, 1221: 1, 1220: 1, 1219: 1, 1218: 1, 1217: 1, 1216: 1, 1215: 1, 1214: 1, 1213: 1, 1212: 1, 1211: 1, 1210: 1, 1209: 1, 1208: 1, 1207: 1, 1206: 1, 1205: 1, 1204: 1, 1203: 1, 1202: 1, 1201: 1, 1200: 1, 1199: 1, 1198: 1, 1197: 1, 1196: 1, 1195: 1, 1194: 1, 1193: 1, 1192: 1, 1191: 1, 1190: 1, 1189: 1, 1188: 1, 1187: 1, 1186: 1, 1185: 1, 1184: 1, 1183: 1, 1182: 1, 1181: 1, 1180: 1, 1179: 1, 1178: 1, 1177: 1, 1176: 1, 1175: 1, 1174: 1, 1173: 1, 1172: 1, 1171: 1, 1170: 1, 1169: 1, 1168: 1, 1167: 1, 1166: 1, 1165: 1, 1164: 1, 1163: 1, 1162: 1, 1161: 1, 1160: 1, 1159: 1, 1158: 1, 1157: 1, 1156: 1, 1155: 1, 1154: 1, 1153: 1, 1152: 1, 1151: 1, 1150: 1, 1149: 1, 1148: 1, 1147: 1, 1146: 1, 1145: 1, 1144: 1, 1143: 1, 1142: 1, 1141: 1, 1140: 1, 1139: 1, 1138: 1, 1137: 1, 1136: 1, 1135: 1, 1134: 1, 1133: 1, 1132: 1, 1131: 1, 1130: 1, 1129: 1, 1128: 1, 1127: 1, 1126: 1, 1125: 1, 1124: 1, 1123: 1, 1122: 1, 1121: 1, 1120: 1, 1119: 1, 1118: 1, 1117: 1, 1116: 1, 1115: 1, 1114: 1, 1113: 1, 1112: 1, 1111: 1, 1110: 1, 1109: 1, 1108: 1, 1107: 1, 1106: 1, 1105: 1, 1104: 1, 1103: 1, 1102: 1, 1101: 1, 1100: 1, 1099: 1, 1098: 1, 1097: 1, 1096: 1, 1095: 1, 1094: 1, 1093: 1, 1092: 1, 1091: 1, 1090: 1, 1089: 1, 1088: 1, 1087: 1, 1086: 1, 1085: 1, 1084: 1, 1083: 1, 1082: 1, 1081: 1, 1080: 1, 1079: 1, 1078: 1, 1077: 1, 1076: 1, 1075: 1, 1074: 1, 1073: 1, 1072: 1, 1071: 1, 1070: 1, 1069: 1, 1068: 1, 1067: 1, 1066: 1, 1065: 1, 1064: 1, 1063: 1, 1062: 1, 1061: 1, 1060: 1, 1059: 1, 1058: 1, 1057: 1, 1056: 1, 1055: 1, 1054: 1, 1053: 1, 1052: 1, 1051: 1, 1050: 1, 1049: 1, 1048: 1, 1047: 1, 1046: 1, 1045: 1, 1044: 1, 1043: 1, 1042: 1, 1041: 1, 1040: 1, 1039: 1, 1038: 1, 1037: 1, 1036: 1, 1035: 1, 1034: 1, 1033: 1, 1032: 1, 1031: 1, 1030: 1, 1029: 1, 1028: 1, 1027: 1, 1026: 1, 1025: 1, 1024: 1, 1023: 1, 1022: 1, 1021: 1, 1020: 1, 1019: 1, 1018: 1, 1017: 1, 1016: 1, 1015: 1, 1014: 1, 1013: 1, 1012: 1, 1011: 1, 1010: 1, 1009: 1, 1008: 1, 1007: 1, 1006: 1, 1005: 1, 1004: 1, 1003: 1, 1002: 1, 1001: 1, 1000: 1, 999: 1, 998: 1, 997: 1, 996: 1, 995: 1, 994: 1, 993: 1, 992: 1, 991: 1, 990: 1, 989: 1, 988: 1, 987: 1, 986: 1, 985: 1, 984: 1, 983: 1, 982: 1, 981: 1, 980: 1, 979: 1, 978: 1, 977: 1, 976: 1, 975: 1, 974: 1, 973: 1, 972: 1, 971: 1, 970: 1, 969: 1, 968: 1, 967: 1, 966: 1, 965: 1, 964: 1, 963: 1, 962: 1, 961: 1, 960: 1, 959: 1, 958: 1, 957: 1, 956: 1, 955: 1, 954: 1, 953: 1, 952: 1, 951: 1, 950: 1, 949: 1, 948: 1, 947: 1, 946: 1, 945: 1,

944: 1, 943: 1, 942: 1, 941: 1, 940: 1, 939: 1, 938: 1, 937: 1, 936:
1, 935: 1, 934: 1, 933: 1, 932: 1, 931: 1, 930: 1, 929: 1, 928: 1, 92
7: 1, 926: 1, 925: 1, 924: 1, 923: 1, 922: 1, 921: 1, 920: 1, 919: 1, 9
18: 1, 917: 1, 916: 1, 915: 1, 914: 1, 913: 1, 912: 1, 911: 1, 910: 1,
909: 1, 908: 1, 907: 1, 906: 1, 905: 1, 904: 1, 903: 1, 902: 1, 901:
1, 900: 1, 899: 1, 898: 1, 897: 1, 896: 1, 895: 1, 894: 1, 893: 1, 89
2: 1, 891: 1, 890: 1, 889: 1, 888: 1, 887: 1, 886: 1, 885: 1, 884: 1, 8
83: 1, 882: 1, 881: 1, 880: 1, 879: 1, 878: 1, 877: 1, 876: 1, 875: 1,
874: 1, 873: 1, 872: 1, 871: 1, 870: 1, 869: 1, 868: 1, 867: 1, 866:
1, 865: 1, 864: 1, 863: 1, 862: 1, 861: 1, 860: 1, 859: 1, 858: 1, 85
7: 1, 856: 1, 855: 1, 854: 1, 853: 1, 852: 1, 851: 1, 850: 1, 849: 1, 8
48: 1, 847: 1, 846: 1, 845: 1, 844: 1, 843: 1, 842: 1, 841: 1, 840: 1,
839: 1, 838: 1, 837: 1, 836: 1, 835: 1, 834: 1, 833: 1, 832: 1, 831:
1, 830: 1, 829: 1, 828: 1, 827: 1, 826: 1, 825: 1, 824: 1, 823: 1, 82
2: 1, 821: 1, 820: 1, 819: 1, 818: 1, 817: 1, 816: 1, 815: 1, 814: 1, 8
13: 1, 812: 1, 811: 1, 810: 1, 809: 1, 808: 1, 807: 1, 806: 1, 805: 1,
804: 1, 803: 1, 802: 1, 801: 1, 800: 1, 799: 1, 798: 1, 797: 1, 796:
1, 795: 1, 794: 1, 793: 1, 792: 1, 791: 1, 790: 1, 789: 1, 788: 1, 78
7: 1, 786: 1, 785: 1, 784: 1, 783: 1, 782: 1, 781: 1, 780: 1, 779: 1, 7
78: 1, 777: 1, 776: 1, 775: 1, 774: 1, 773: 1, 772: 1, 771: 1, 770: 1,
769: 1, 768: 1, 767: 1, 766: 1, 765: 1, 764: 1, 763: 1, 762: 1, 761:
1, 760: 1, 759: 1, 758: 1, 757: 1, 756: 1, 755: 1, 754: 1, 753: 1, 75
2: 1, 751: 1, 750: 1, 749: 1, 748: 1, 747: 1, 746: 1, 745: 1, 744: 1, 7
43: 1, 742: 1, 741: 1, 740: 1, 739: 1, 738: 1, 737: 1, 736: 1, 735: 1,
734: 1, 733: 1, 732: 1, 731: 1, 730: 1, 729: 1, 728: 1, 727: 1, 726:
1, 725: 1, 724: 1, 723: 1, 722: 1, 721: 1, 720: 1, 719: 1, 718: 1, 71
7: 1, 716: 1, 715: 1, 714: 1, 713: 1, 712: 1, 711: 1, 710: 1, 709: 1, 7
08: 1, 707: 1, 706: 1, 705: 1, 704: 1, 703: 1, 702: 1, 701: 1, 700: 1,
699: 1, 698: 1, 697: 1, 696: 1, 695: 1, 694: 1, 693: 1, 692: 1, 691:
1, 690: 1, 689: 1, 688: 1, 687: 1, 686: 1, 685: 1, 684: 1, 683: 1, 68
2: 1, 681: 1, 680: 1, 679: 1, 678: 1, 677: 1, 676: 1, 675: 1, 674: 1, 6
73: 1, 672: 1, 671: 1, 670: 1, 669: 1, 668: 1, 667: 1, 666: 1, 665: 1,
664: 1, 663: 1, 662: 1, 661: 1, 660: 1, 659: 1, 658: 1, 657: 1, 656:
1, 655: 1, 654: 1, 653: 1, 652: 1, 651: 1, 650: 1, 649: 1, 648: 1, 64
7: 1, 646: 1, 645: 1, 644: 1, 643: 1, 642: 1, 641: 1, 640: 1, 639: 1, 6
38: 1, 637: 1, 636: 1, 635: 1, 634: 1, 633: 1, 632: 1, 631: 1, 630: 1,
629: 1, 628: 1, 627: 1, 626: 1, 625: 1, 624: 1, 623: 1, 622: 1, 621:
1, 620: 1, 619: 1, 618: 1, 617: 1, 616: 1, 615: 1, 614: 1, 613: 1, 61
2: 1, 611: 1, 610: 1, 609: 1, 608: 1, 607: 1, 606: 1, 605: 1, 604: 1, 6

03: 1, 602: 1, 601: 1, 600: 1, 599: 1, 598: 1, 597: 1, 596: 1, 595: 1,
594: 1, 593: 1, 592: 1, 591: 1, 590: 1, 589: 1, 588: 1, 587: 1, 586:
1, 585: 1, 584: 1, 583: 1, 582: 1, 581: 1, 580: 1, 579: 1, 578: 1, 57
7: 1, 576: 1, 575: 1, 574: 1, 573: 1, 572: 1, 571: 1, 570: 1, 569: 1, 5
68: 1, 567: 1, 566: 1, 565: 1, 564: 1, 563: 1, 562: 1, 561: 1, 560: 1,
559: 1, 558: 1, 557: 1, 556: 1, 555: 1, 554: 1, 553: 1, 552: 1, 551:
1, 550: 1, 549: 1, 548: 1, 547: 1, 546: 1, 545: 1, 544: 1, 543: 1, 54
2: 1, 541: 1, 540: 1, 539: 1, 538: 1, 537: 1, 536: 1, 535: 1, 534: 1, 5
33: 1, 532: 1, 531: 1, 530: 1, 529: 1, 528: 1, 527: 1, 526: 1, 525: 1,
524: 1, 523: 1, 522: 1, 521: 1, 520: 1, 519: 1, 518: 1, 517: 1, 516:
1, 515: 1, 514: 1, 513: 1, 512: 1, 511: 1, 510: 1, 509: 1, 508: 1, 50
7: 1, 506: 1, 505: 1, 504: 1, 503: 1, 502: 1, 501: 1, 500: 1, 499: 1, 4
98: 1, 497: 1, 496: 1, 495: 1, 494: 1, 493: 1, 492: 1, 491: 1, 490: 1,
489: 1, 488: 1, 487: 1, 486: 1, 485: 1, 484: 1, 483: 1, 482: 1, 481:
1, 480: 1, 479: 1, 478: 1, 477: 1, 476: 1, 475: 1, 474: 1, 473: 1, 47
2: 1, 471: 1, 470: 1, 469: 1, 468: 1, 467: 1, 466: 1, 465: 1, 464: 1, 4
63: 1, 462: 1, 461: 1, 460: 1, 459: 1, 458: 1, 457: 1, 456: 1, 455: 1,
454: 1, 453: 1, 452: 1, 451: 1, 450: 1, 449: 1, 448: 1, 447: 1, 446:
1, 445: 1, 444: 1, 443: 1, 442: 1, 441: 1, 440: 1, 439: 1, 438: 1, 43
7: 1, 436: 1, 435: 1, 434: 1, 433: 1, 432: 1, 431: 1, 430: 1, 429: 1, 4
28: 1, 427: 1, 426: 1, 425: 1, 424: 1, 423: 1, 422: 1, 421: 1, 420: 1,
419: 1, 418: 1, 417: 1, 416: 1, 415: 1, 414: 1, 413: 1, 412: 1, 411:
1, 410: 1, 409: 1, 408: 1, 407: 1, 406: 1, 405: 1, 404: 1, 403: 1, 40
2: 1, 401: 1, 400: 1, 399: 1, 398: 1, 397: 1, 396: 1, 395: 1, 394: 1, 3
93: 1, 392: 1, 391: 1, 390: 1, 389: 1, 388: 1, 387: 1, 386: 1, 385: 1,
384: 1, 383: 1, 382: 1, 381: 1, 380: 1, 379: 1, 378: 1, 377: 1, 376:
1, 375: 1, 374: 1, 373: 1, 372: 1, 371: 1, 370: 1, 369: 1, 368: 1, 36
7: 1, 366: 1, 365: 1, 364: 1, 363: 1, 362: 1, 361: 1, 360: 1, 359: 1, 3
58: 1, 357: 1, 356: 1, 355: 1, 354: 1, 353: 1, 352: 1, 351: 1, 350: 1,
349: 1, 348: 1, 347: 1, 346: 1, 345: 1, 344: 1, 343: 1, 342: 1, 341:
1, 340: 1, 339: 1, 338: 1, 337: 1, 336: 1, 335: 1, 334: 1, 333: 1, 33
2: 1, 331: 1, 330: 1, 329: 1, 328: 1, 327: 1, 326: 1, 325: 1, 324: 1, 3
23: 1, 322: 1, 321: 1, 320: 1, 319: 1, 318: 1, 317: 1, 316: 1, 315: 1,
314: 1, 313: 1, 312: 1, 311: 1, 310: 1, 309: 1, 308: 1, 307: 1, 306:
1, 305: 1, 304: 1, 303: 1, 302: 1, 301: 1, 300: 1, 299: 1, 298: 1, 29
7: 1, 296: 1, 295: 1, 294: 1, 293: 1, 292: 1, 291: 1, 290: 1, 289: 1, 2
88: 1, 287: 1, 286: 1, 285: 1, 284: 1, 283: 1, 282: 1, 281: 1, 280: 1,
279: 1, 278: 1, 277: 1, 276: 1, 275: 1, 274: 1, 273: 1, 272: 1, 271:
1, 270: 1, 269: 1, 268: 1, 267: 1, 266: 1, 265: 1, 264: 1, 263: 1, 26

```

2: 1, 261: 1, 260: 1, 259: 1, 258: 1, 257: 1, 256: 1, 255: 1, 254: 1, 2
53: 1, 252: 1, 251: 1, 250: 1, 249: 1, 248: 1, 247: 1, 246: 1, 245: 1,
244: 1, 243: 1, 242: 1, 241: 1, 240: 1, 239: 1, 238: 1, 237: 1, 236:
1, 235: 1, 234: 1, 233: 1, 232: 1, 231: 1, 230: 1, 229: 1, 228: 1, 22
7: 1, 226: 1, 225: 1, 224: 1, 223: 1, 222: 1, 221: 1, 220: 1, 219: 1, 2
18: 1, 217: 1, 216: 1, 215: 1, 214: 1, 213: 1, 212: 1, 211: 1, 210: 1,
209: 1, 208: 1, 207: 1, 206: 1, 205: 1, 204: 1, 203: 1, 202: 1, 201:
1, 200: 1, 199: 1, 198: 1, 197: 1, 196: 1, 195: 1, 194: 1, 193: 1, 19
2: 1, 191: 1, 190: 1, 189: 1, 188: 1, 187: 1, 186: 1, 185: 1, 184: 1, 1
83: 1, 182: 1, 181: 1, 180: 1, 179: 1, 178: 1, 177: 1, 176: 1, 175: 1,
174: 1, 173: 1, 172: 1, 171: 1, 170: 1, 169: 1, 168: 1, 167: 1, 166:
1, 165: 1, 164: 1, 163: 1, 162: 1, 161: 1, 160: 1, 159: 1, 158: 1, 15
7: 1, 156: 1, 155: 1, 154: 1, 153: 1, 152: 1, 151: 1, 150: 1, 149: 1, 1
48: 1, 147: 1, 146: 1, 145: 1, 144: 1, 143: 1, 142: 1, 141: 1, 140: 1,
139: 1, 138: 1, 137: 1, 136: 1, 135: 1, 134: 1, 133: 1, 132: 1, 131:
1, 130: 1, 129: 1, 128: 1, 127: 1, 126: 1, 125: 1, 124: 1, 123: 1, 12
2: 1, 121: 1, 120: 1, 119: 1, 118: 1, 117: 1, 116: 1, 115: 1, 114: 1, 1
13: 1, 112: 1, 111: 1, 110: 1, 109: 1, 108: 1, 107: 1, 106: 1, 105: 1,
104: 1, 103: 1, 102: 1, 101: 1, 100: 1, 99: 1, 98: 1, 97: 1, 96: 1, 9
5: 1, 94: 1, 93: 1, 92: 1, 91: 1, 90: 1, 89: 1, 88: 1, 87: 1, 86: 1, 8
5: 1, 84: 1, 83: 1, 82: 1, 81: 1, 80: 1, 79: 1, 78: 1, 77: 1, 76: 1, 7
5: 1, 74: 1, 73: 1, 72: 1, 71: 1, 70: 1, 69: 1, 68: 1, 67: 1, 66: 1, 6
5: 1, 64: 1, 63: 1, 62: 1, 61: 1, 60: 1, 59: 1, 58: 1, 57: 1, 56: 1, 5
5: 1, 54: 1, 53: 1, 52: 1, 51: 1, 50: 1, 49: 1, 48: 1, 47: 1, 46: 1, 4
5: 1, 44: 1, 43: 1, 42: 1, 41: 1, 40: 1, 39: 1, 38: 1, 37: 1, 36: 1, 3
5: 1, 34: 1, 33: 1, 32: 1, 31: 1, 30: 1, 29: 1, 28: 1, 27: 1, 26: 1, 2
5: 1, 24: 1, 23: 1, 22: 1, 21: 1, 20: 1, 19: 1, 18: 1, 17: 1, 16: 1, 1
5: 1, 14: 1, 13: 1, 12: 1, 11: 1, 10: 1, 9: 1, 8: 1, 7: 1, 6: 1, 5: 1,
4: 1, 3: 1, 2: 1, 1: 1, 0: 1})

```

```

In [52]: # Train a Logistic regression+Calibration model using text features whi
cha re on-hot encoded
alpha = [10 ** x for x in range(-5, 1)]

# read more about SGDClassifier() at http://scikit-learn.org/stable/mod
ules/generated/sklearn.linear_model.SGDClassifier.html
# -----
# default parameters
# SGDClassifier(loss='hinge', penalty='l2', alpha=0.0001, l1_ratio=0.1

```



```

5, fit_intercept=True, max_iter=None, tol=None,
# shuffle=True, verbose=0, epsilon=0.1, n_jobs=1, random_state=None, le
arning_rate='optimal', eta0=0.0, power_t=0.5,
# class_weight=None, warm_start=False, average=False, n_iter=None)

# some of methods
# fit(X, y[, coef_init, intercept_init, ...])    Fit linear model with S
tochastic Gradient Descent.
# predict(X)    Predict class labels for samples in X.

#-----
# video link:
#-----

cv_log_error_array=[]
for i in alpha:
    clf = SGDClassifier(alpha=i, penalty='l2', loss='log', random_state
=42)
    clf.fit(train_text_feature_onehotCoding, y_train)

    sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
    sig_clf.fit(train_text_feature_onehotCoding, y_train)
    predict_y = sig_clf.predict_proba(cv_text_feature_onehotCoding)
    cv_log_error_array.append(log_loss(y_cv, predict_y, labels=clf.clas
ses_, eps=1e-15))
    print('For values of alpha = ', i, "The log loss is:", log_loss(y_cv
, predict_y, labels=clf.classes_, eps=1e-15))

fig, ax = plt.subplots()
ax.plot(alpha, cv_log_error_array, c='g')
for i, txt in enumerate(np.round(cv_log_error_array,3)):
    ax.annotate((alpha[i],np.round(txt,3)), (alpha[i],cv_log_error_arra
y[i]))
plt.grid()
plt.title("Cross Validation Error for each alpha")
plt.xlabel("Alpha i's")
plt.ylabel("Error measure")
plt.show()

```

```

best_alpha = np.argmin(cv_log_error_array)
clf = SGDClassifier(alpha=alpha[best_alpha], penalty='l2', loss='log',
random_state=42)
clf.fit(train_text_feature_onehotCoding, y_train)
sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
sig_clf.fit(train_text_feature_onehotCoding, y_train)

#summarizing text data
text_best_alpha = best_alpha

predict_y = sig_clf.predict_proba(train_text_feature_onehotCoding)
text_train_log_loss = log_loss(y_train, predict_y, labels=clf.classes_,
eps=1e-15)
print('For values of best alpha = ', alpha[best_alpha], "The train log
loss is:",log_loss(y_train, predict_y, labels=clf.classes_, eps=1e-15
))

predict_y = sig_clf.predict_proba(cv_text_feature_onehotCoding)
text_cv_log_loss = log_loss(y_cv, predict_y, labels=clf.classes_, eps=1
e-15)
print('For values of best alpha = ', alpha[best_alpha], "The cross vali
dation log loss is:",log_loss(y_cv, predict_y, labels=clf.classes_, eps
=1e-15))

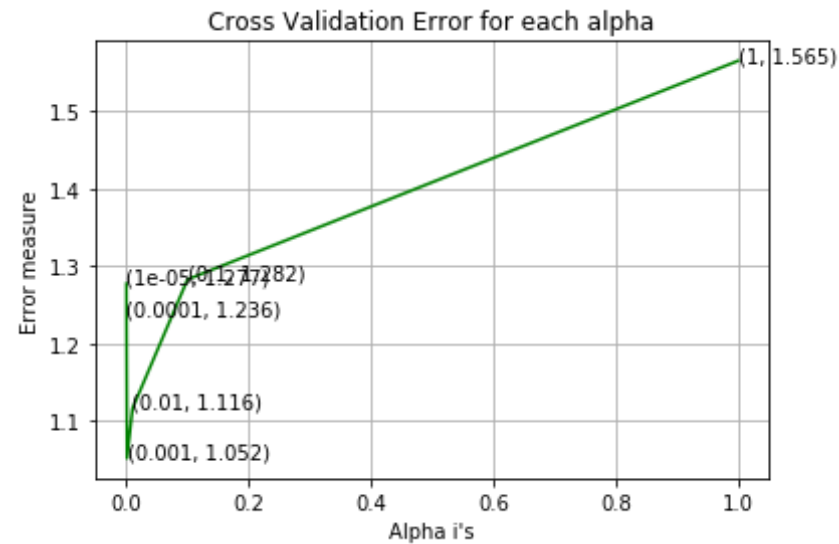
predict_y = sig_clf.predict_proba(test_text_feature_onehotCoding)
text_test_log_loss = log_loss(y_test, predict_y, labels=clf.classes_,
eps=1e-15)
print('For values of best alpha = ', alpha[best_alpha], "The test log l
oss is:",log_loss(y_test, predict_y, labels=clf.classes_, eps=1e-15))

```

```

For values of alpha = 1e-05 The log loss is: 1.2769522995023777
For values of alpha = 0.0001 The log loss is: 1.2362798742894758
For values of alpha = 0.001 The log loss is: 1.0515673752791947
For values of alpha = 0.01 The log loss is: 1.115924025235313
For values of alpha = 0.1 The log loss is: 1.282460987940429
For values of alpha = 1 The log loss is: 1.5645882577699846

```



For values of best alpha = 0.001 The train log loss is: 0.6808062934422892

For values of best alpha = 0.001 The cross validation log loss is: 1.0515673752791947

For values of best alpha = 0.001 The test log loss is: 1.1663402223432042

Q. Is the Text feature stable across all the data sets (Test, Train, Cross validation)?

Ans. Yes, it seems like!

```
In [53]: def get_intersec_text(df):
df_text_vec = TfidfVectorizer(min_df=5)
df_text_fea = df_text_vec.fit_transform(df['TEXT'])
df_text_features = df_text_vec.get_feature_names()

df_text_fea_counts = df_text_fea.sum(axis=0).A1
df_text_fea_dict = dict(zip(list(df_text_features), df_text_fea_counts))
len1 = len(set(df_text_features))
```

```
len2 = len(set(tfidf_train_text_onehotencoding) & set(df_text_features))
return len1, len2
```

```
In [54]: len1, len2 = get_intersec_text(test_df)
print(np.round((len2/len1)*100, 3), "% of word of test data appeared in train data")
len1, len2 = get_intersec_text(cv_df)
print(np.round((len2/len1)*100, 3), "% of word of Cross Validation appeared in train data")

text_stability = (np.round((len2/len1)*100, 3) + np.round((len2/len1)*100, 3))/2
```

99.334 % of word of test data appeared in train data

99.613 % of word of Cross Validation appeared in train data

4. Machine Learning Models

```
In [55]: #Data preparation for ML models.

#Misc. functions for ML models

def predict_and_plot_confusion_matrix(train_x, train_y, test_x, test_y, clf):
    clf.fit(train_x, train_y)
    sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
    sig_clf.fit(train_x, train_y)
    pred_y = sig_clf.predict(test_x)

    # for calculating log_loss we will provide the array of probabilities belongs to each class
    print("Log loss :", log_loss(test_y, sig_clf.predict_proba(test_x)))
    # calculating the number of data points that are misclassified
    mis_classified = np.count_nonzero((pred_y - test_y))/test_y.shape[0]
    print("Number of mis-classified points :", mis_classified)
```

```
plot_confusion_matrix(test_y, pred_y)
return mis_classified
```

```
In [56]: def report_log_loss(train_x, train_y, test_x, test_y, clf):
        clf.fit(train_x, train_y)
        sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
        sig_clf.fit(train_x, train_y)
        sig_clf_probs = sig_clf.predict_proba(test_x)
        return log_loss(test_y, sig_clf_probs, eps=1e-15)
```

```
In [57]: # this function will be used just for naive bayes
        # for the given indices, we will print the name of the features
        # and we will check whether the feature present in the test point text
        or not
        def get_impfeature_names(indices, text, gene, var, no_features):
            gene_count_vec = CountVectorizer()
            var_count_vec = CountVectorizer()
            text_count_vec = TfidfVectorizer(min_df=5)

            gene_vec = gene_count_vec.fit(train_df['Gene'])
            var_vec = var_count_vec.fit(train_df['Variation'])
            text_vec = text_count_vec.fit(train_df['TEXT'])

            fea1_len = len(gene_vec.get_feature_names())
            fea2_len = len(var_count_vec.get_feature_names())

            word_present = 0
            for i,v in enumerate(indices):
                if (v < fea1_len):
                    word = gene_vec.get_feature_names()[v]
                    yes_no = True if word == gene else False
                    if yes_no:
                        word_present += 1
                        print(i, "Gene feature [{}] present in test data point
[{}]" .format(word,yes_no))
                elif (v < fea1_len+fea2_len):
                    word = var_vec.get_feature_names()[v-(fea1_len)]
                    yes_no = True if word == var else False
```

```

        if yes_no:
            word_present += 1
            print(i, "variation feature [{}]"
                  "present in test data point [{}]"
                  .format(word, yes_no))
        else:
            word = text_vec.get_feature_names()[v-(fea1_len+fea2_len)]
            yes_no = True if word in text.split() else False
            if yes_no:
                word_present += 1
                print(i, "Text feature [{}]"
                      "present in test data point [{}]"
                      .format(word, yes_no))

    print("Out of the top ", no_features, " features ", word_present, " are present in query point")

```

Stacking the three types of features

```

In [58]: # merging gene, variance and text features

# building train, test and cross validation data sets
# a = [[1, 2],
#       [3, 4]]
# b = [[4, 5],
#       [6, 7]]
# hstack(a, b) = [[1, 2, 4, 5],
#                 [ 3, 4, 6, 7]]

train_gene_var_onehotCoding = hstack((train_gene_feature_onehotCoding,
train_variation_feature_onehotCoding))
test_gene_var_onehotCoding = hstack((test_gene_feature_onehotCoding,
test_variation_feature_onehotCoding))
cv_gene_var_onehotCoding = hstack((cv_gene_feature_onehotCoding,
cv_variation_feature_onehotCoding))

train_x_onehotCoding = hstack((train_gene_var_onehotCoding, train_text_
feature_onehotCoding)).tocsr()

```

```

train_y = np.array(list(train_df['Class']))

test_x_onehotCoding = hstack((test_gene_var_onehotCoding, test_text_feature_onehotCoding)).tocsr()
test_y = np.array(list(test_df['Class']))

cv_x_onehotCoding = hstack((cv_gene_var_onehotCoding, cv_text_feature_onehotCoding)).tocsr()
cv_y = np.array(list(cv_df['Class']))

train_gene_var_responseCoding = np.hstack((train_gene_feature_responseCoding, train_variation_feature_responseCoding))
test_gene_var_responseCoding = np.hstack((test_gene_feature_responseCoding, test_variation_feature_responseCoding))
cv_gene_var_responseCoding = np.hstack((cv_gene_feature_responseCoding, cv_variation_feature_responseCoding))

train_x_responseCoding = np.hstack((train_gene_var_responseCoding, train_text_feature_responseCoding))
test_x_responseCoding = np.hstack((test_gene_var_responseCoding, test_text_feature_responseCoding))
cv_x_responseCoding = np.hstack((cv_gene_var_responseCoding, cv_text_feature_responseCoding))

```

```

In [59]: print("One hot encoding features :")
print("(number of data points * number of features) in train data = ", train_x_onehotCoding.shape)
print("(number of data points * number of features) in test data = ", test_x_onehotCoding.shape)
print("(number of data points * number of features) in cross validation data = ", cv_x_onehotCoding.shape)

```

```

One hot encoding features :
(number of data points * number of features) in train data = (2124, 40956)
(number of data points * number of features) in test data = (665, 40956)

```

```
(number of data points * number of features) in cross validation data =  
(532, 40956)
```

```
In [60]: print(" Response encoding features :")  
print("(number of data points * number of features) in train data = ",  
train_x_responseCoding.shape)  
print("(number of data points * number of features) in test data = ", t  
est_x_responseCoding.shape)  
print("(number of data points * number of features) in cross validation  
data =", cv_x_responseCoding.shape)
```

```
Response encoding features :  
(number of data points * number of features) in train data = (2124, 2  
7)  
(number of data points * number of features) in test data = (665, 27)  
(number of data points * number of features) in cross validation data =  
(532, 27)
```

4.1. Base Line Model

4.1.1. Naive Bayes

4.1.1.1. Hyper parameter tuning

```
In [61]: # find more about Multinomial Naive base function here http://scikit-learn.org/stable/modules/generated/sklearn.naive\_bayes.MultinomialNB.html  
# -----  
# default paramters  
# sklearn.naive_bayes.MultinomialNB(alpha=1.0, fit_prior=True, class_pr  
ior=None)  
  
# some of methods of MultinomialNB()  
# fit(X, y[, sample_weight])    Fit Naive Bayes classifier according to  
X, y  
# predict(X)    Perform classification on an array of test vectors X.
```



```

# predict_log_proba(X) Return log-probability estimates for the test vector X.
# -----
# video link: https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/naive-bayes-algorithm-1/
# -----

# find more about CalibratedClassifierCV here at http://scikit-learn.org/stable/modules/generated/sklearn.calibration.CalibratedClassifierCV.html
# -----
# default parameters
# sklearn.calibration.CalibratedClassifierCV(base_estimator=None, method='sigmoid', cv=3)
#
# some of the methods of CalibratedClassifierCV()
# fit(X, y[, sample_weight]) Fit the calibrated model
# get_params([deep]) Get parameters for this estimator.
# predict(X) Predict the target of new samples.
# predict_proba(X) Posterior probabilities of classification
# -----
# video link: https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/naive-bayes-algorithm-1/
# -----

alpha = [0.00001, 0.0001, 0.001, 0.1, 1, 10, 100, 1000]
cv_log_error_array = []
for i in alpha:
    print("for alpha =", i)
    clf = MultinomialNB(alpha=i)
    clf.fit(train_x_onehotCoding, train_y)
    sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
    sig_clf.fit(train_x_onehotCoding, train_y)
    sig_clf_probs = sig_clf.predict_proba(cv_x_onehotCoding)
    cv_log_error_array.append(log_loss(cv_y, sig_clf_probs, labels=clf.classes_, eps=1e-15))
    # to avoid rounding error while multiplying probabilities we use log

```

```

-probability estimates
    print("Log Loss :", log_loss(cv_y, sig_clf_probs))

fig, ax = plt.subplots()
ax.plot(np.log10(alpha), cv_log_error_array, c='g')
for i, txt in enumerate(np.round(cv_log_error_array, 3)):
    ax.annotate((alpha[i], str(txt)), (np.log10(alpha[i]), cv_log_error_a
rray[i]))
plt.grid()
plt.xticks(np.log10(alpha))
plt.title("Cross Validation Error for each alpha")
plt.xlabel("Alpha i's")
plt.ylabel("Error measure")
plt.show()

best_alpha = np.argmin(cv_log_error_array)
clf = MultinomialNB(alpha=alpha[best_alpha])
clf.fit(train_x_onehotCoding, train_y)
sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
sig_clf.fit(train_x_onehotCoding, train_y)

# summarizing data
nb_best_alpha = alpha[best_alpha]
nb_encoding = "One hot"

predict_y = sig_clf.predict_proba(train_x_onehotCoding)
nb_train_log_loss = log_loss(y_train, predict_y, labels=clf.classes_, e
ps=1e-15)
print('For values of best alpha = ', alpha[best_alpha], "The train log
loss is:", log_loss(y_train, predict_y, labels=clf.classes_, eps=1e-15
))

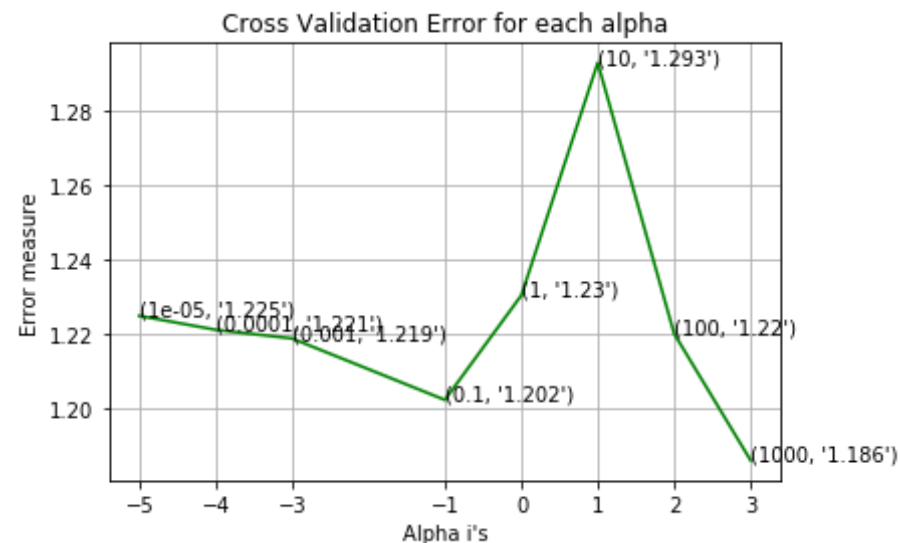
predict_y = sig_clf.predict_proba(cv_x_onehotCoding)
nb_cv_log_loss = log_loss(y_cv, predict_y, labels=clf.classes_, eps=1e-
15)
print('For values of best alpha = ', alpha[best_alpha], "The cross vali
dation log loss is:", log_loss(y_cv, predict_y, labels=clf.classes_, eps

```

```
=1e-15))

predict_y = sig_clf.predict_proba(test_x_onehotCoding)
nb_test_log_loss = log_loss(y_test, predict_y, labels=clf.classes_, eps
=1e-15)
print('For values of best alpha = ', alpha[best_alpha], "The test log l
oss is:", log_loss(y_test, predict_y, labels=clf.classes_, eps=1e-15))

for alpha = 1e-05
Log Loss : 1.2248916439718198
for alpha = 0.0001
Log Loss : 1.22114957784026
for alpha = 0.001
Log Loss : 1.2188172758242328
for alpha = 0.1
Log Loss : 1.2023318145337416
for alpha = 1
Log Loss : 1.2303754191239364
for alpha = 10
Log Loss : 1.2928036107510528
for alpha = 100
Log Loss : 1.2200890369209003
for alpha = 1000
Log Loss : 1.1861092473639456
```



For values of best alpha = 1000 The train log loss is: 0.9935963631522705

For values of best alpha = 1000 The cross validation log loss is: 1.1861092473639456

For values of best alpha = 1000 The test log loss is: 1.2187926224279089

4.1.1.2. Testing the model with best hyper paramters

```
In [62]: # find more about Multinomial Naive base function here http://scikit-learn.org/stable/modules/generated/sklearn.naive\_bayes.MultinomialNB.html
# -----
# default paramters
# sklearn.naive_bayes.MultinomialNB(alpha=1.0, fit_prior=True, class_prior=None)

# some of methods of MultinomialNB()
# fit(X, y[, sample_weight])    Fit Naive Bayes classifier according to X, y
# predict(X)    Perform classification on an array of test vectors X.
# predict_log_proba(X) Return log-probability estimates for the test v
```

```

ector X.
# -----
# video link: https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/naive-bayes-algorithm-1/
# -----

# find more about CalibratedClassifierCV here at http://scikit-learn.org/stable/modules/generated/sklearn.calibration.CalibratedClassifierCV.html
# -----
# default paramters
# sklearn.calibration.CalibratedClassifierCV(base_estimator=None, method='sigmoid', cv=3)
#
# some of the methods of CalibratedClassifierCV()
# fit(X, y[, sample_weight])    Fit the calibrated model
# get_params([deep])           Get parameters for this estimator.
# predict(X)                    Predict the target of new samples.
# predict_proba(X)              Posterior probabilities of classification
# -----

clf = MultinomialNB(alpha=alpha[best_alpha])
clf.fit(train_x_onehotCoding, train_y)
sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
sig_clf.fit(train_x_onehotCoding, train_y)
sig_clf_probs = sig_clf.predict_proba(cv_x_onehotCoding)
# to avoid rounding error while multiplying probabilities we use log-probability estimates
print("Log Loss :", log_loss(cv_y, sig_clf_probs))

nb_misclassified = np.count_nonzero((sig_clf.predict(cv_x_onehotCoding) - cv_y))/cv_y.shape[0]
print("Number of misclassified point :", nb_misclassified)
plot_confusion_matrix(cv_y, sig_clf.predict(cv_x_onehotCoding.toarray()))

```

```

Log Loss : 1.1861092473639456
Number of misclassified point : 0.3966165413533835

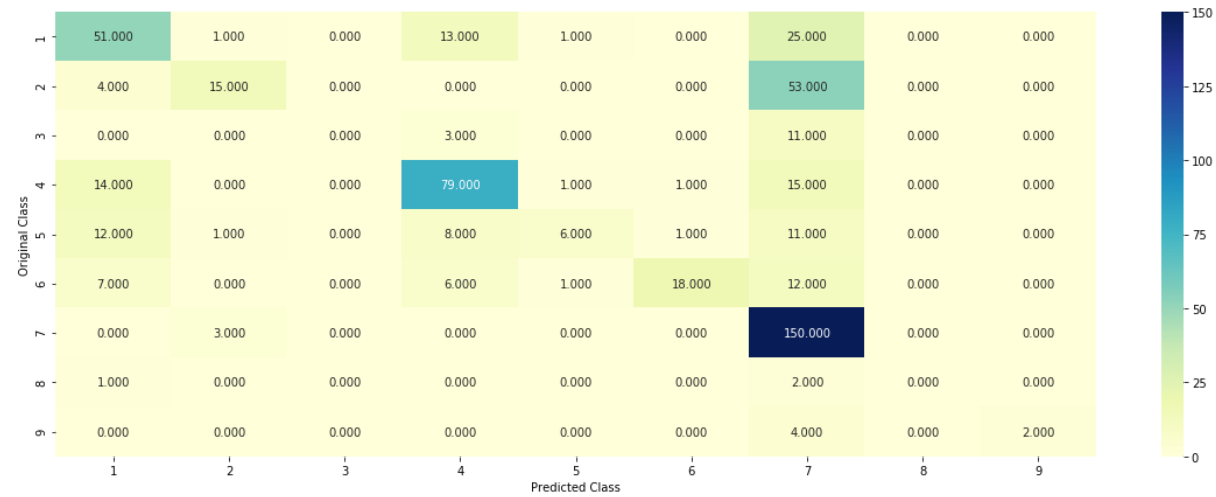
```

```

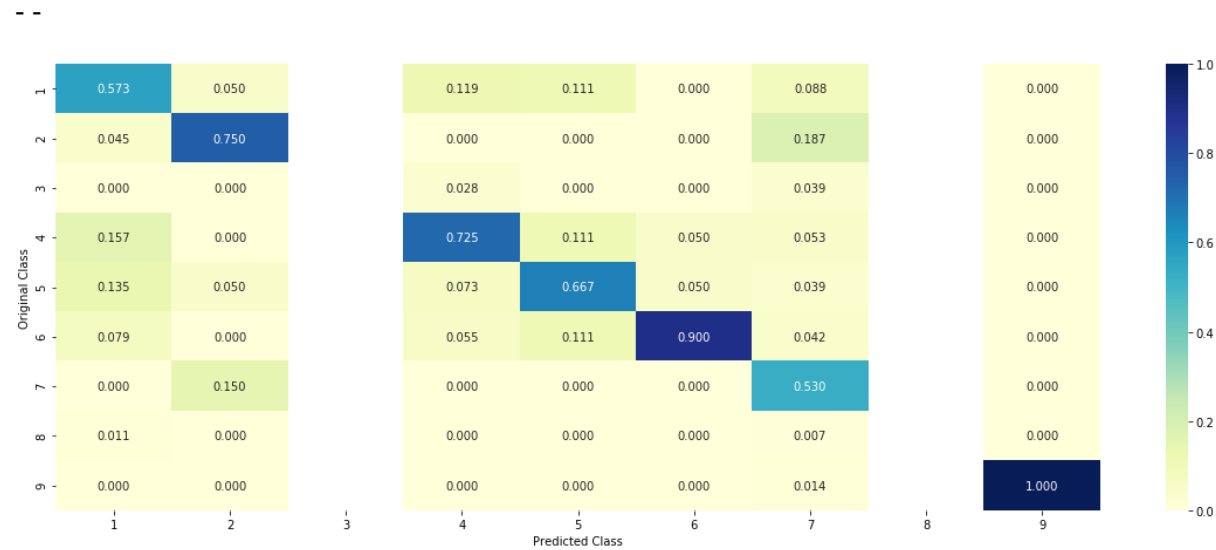
----- Confusion matrix -----

```

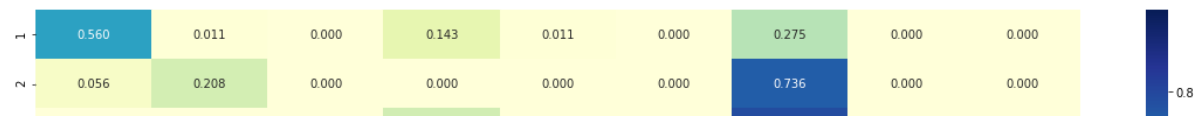
CONFUSION MATRIX

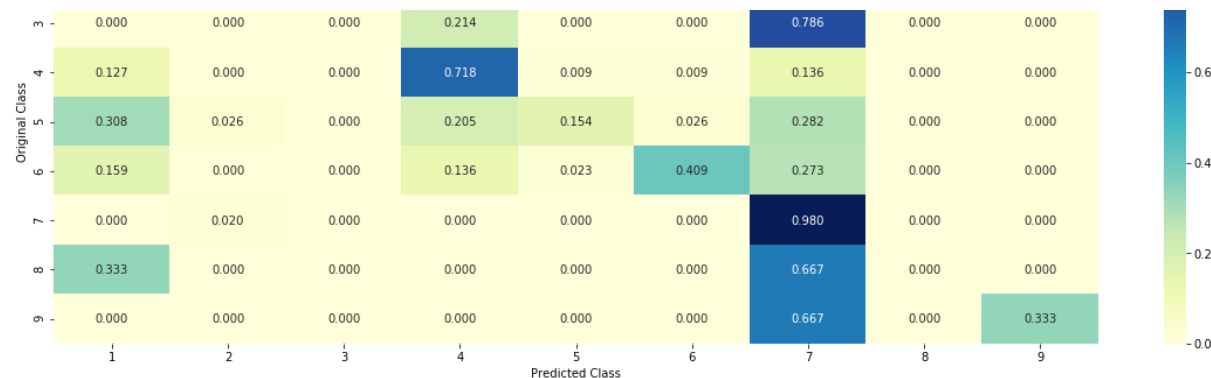


Precision matrix (Column Sum=1)



Recall matrix (Row sum=1)





4.1.1.3. Feature Importance, Correctly classified point

```
In [63]: test_point_index = 1
no_feature = 100
predicted_cls = sig_clf.predict(test_x_onehotCoding[test_point_index])
print("Predicted Class :", predicted_cls[0])
print("Predicted Class Probabilities:", np.round(sig_clf.predict_proba(
test_x_onehotCoding[test_point_index]),4))
print("Actual Class :", test_y[test_point_index])
indices = np.argsort(-clf.coef_)[predicted_cls-1][:,:no_feature]
print("-"*50)
get_impfeature_names(indices[0], test_df['TEXT'].iloc[test_point_index],
test_df['Gene'].iloc[test_point_index],test_df['Variation'].iloc[test_point_index], no_feature)
```

```
Predicted Class : 1
Predicted Class Probabilities: [[5.820e-01 2.000e-03 1.000e-04 2.486e-0
1 1.188e-01 2.630e-02 2.190e-02
3.000e-04 0.000e+00]]
```

```
Actual Class : 1
```

```
-----
11 Text feature [protein] present in test data point [True]
12 Text feature [type] present in test data point [True]
13 Text feature [functions] present in test data point [True]
15 Text feature [involved] present in test data point [True]
16 Text feature [one] present in test data point [True]
17 Text feature [dual] present in test data point [True]
```

```
17 Text feature [and] present in test data point [True]
18 Text feature [wild] present in test data point [True]
19 Text feature [two] present in test data point [True]
20 Text feature [therefore] present in test data point [True]
21 Text feature [also] present in test data point [True]
22 Text feature [results] present in test data point [True]
23 Text feature [region] present in test data point [True]
24 Text feature [loss] present in test data point [True]
25 Text feature [role] present in test data point [True]
26 Text feature [binding] present in test data point [True]
27 Text feature [control] present in test data point [True]
28 Text feature [human] present in test data point [True]
29 Text feature [function] present in test data point [True]
30 Text feature [reduced] present in test data point [True]
31 Text feature [specific] present in test data point [True]
32 Text feature [affect] present in test data point [True]
33 Text feature [shown] present in test data point [True]
34 Text feature [containing] present in test data point [True]
35 Text feature [using] present in test data point [True]
36 Text feature [expression] present in test data point [True]
37 Text feature [table] present in test data point [True]
38 Text feature [indicate] present in test data point [True]
39 Text feature [either] present in test data point [True]
40 Text feature [possible] present in test data point [True]
41 Text feature [determined] present in test data point [True]
42 Text feature [indicating] present in test data point [True]
43 Text feature [essential] present in test data point [True]
44 Text feature [however] present in test data point [True]
45 Text feature [present] present in test data point [True]
46 Text feature [within] present in test data point [True]
47 Text feature [gene] present in test data point [True]
48 Text feature [important] present in test data point [True]
49 Text feature [indicated] present in test data point [True]
50 Text feature [three] present in test data point [True]
51 Text feature [several] present in test data point [True]
52 Text feature [result] present in test data point [True]
53 Text feature [four] present in test data point [True]
54 Text feature [following] present in test data point [True]
55 Text feature [terminal] present in test data point [True]

58 Text feature [respectively] present in test data point [True]
```


58 Text feature [respectively] present in test data point [True]
60 Text feature [effect] present in test data point [True]
61 Text feature [similar] present in test data point [True]
62 Text feature [may] present in test data point [True]
63 Text feature [critical] present in test data point [True]
64 Text feature [analysis] present in test data point [True]
65 Text feature [addition] present in test data point [True]
66 Text feature [complex] present in test data point [True]
67 Text feature [including] present in test data point [True]
68 Text feature [corresponding] present in test data point [True]
69 Text feature [whether] present in test data point [True]
70 Text feature [well] present in test data point [True]
71 Text feature [mediated] present in test data point [True]
72 Text feature [different] present in test data point [True]
73 Text feature [form] present in test data point [True]
74 Text feature [contains] present in test data point [True]
75 Text feature [interacts] present in test data point [True]
76 Text feature [together] present in test data point [True]
77 Text feature [amino] present in test data point [True]
78 Text feature [observed] present in test data point [True]
79 Text feature [domains] present in test data point [True]
80 Text feature [structure] present in test data point [True]
81 Text feature [compared] present in test data point [True]
82 Text feature [suggest] present in test data point [True]
83 Text feature [previous] present in test data point [True]
84 Text feature [ability] present in test data point [True]
85 Text feature [interact] present in test data point [True]
86 Text feature [used] present in test data point [True]
87 Text feature [thus] present in test data point [True]
88 Text feature [previously] present in test data point [True]
89 Text feature [transcription] present in test data point [True]
90 Text feature [proteins] present in test data point [True]
91 Text feature [10] present in test data point [True]
92 Text feature [plays] present in test data point [True]
93 Text feature [conserved] present in test data point [True]
94 Text feature [significant] present in test data point [True]
95 Text feature [another] present in test data point [True]
96 Text feature [fig] present in test data point [True]
97 Text feature [associated] present in test data point [True]

98 Text feature [length] present in test data point [True]

```
98 Text feature [length] present in test data point [True]
99 Text feature [many] present in test data point [True]
Out of the top 100 features 85 are present in query point
```

4.1.1.4. Feature Importance, Incorrectly classified point

```
In [64]: test_point_index = 100
no_feature = 100
predicted_cls = sig_clf.predict(test_x_onehotCoding[test_point_index])
print("Predicted Class :", predicted_cls[0])
print("Predicted Class Probabilities:", np.round(sig_clf.predict_proba(
test_x_onehotCoding[test_point_index]),4))
print("Actual Class :", test_y[test_point_index])
indices = np.argsort(-clf.coef_)[predicted_cls-1][:,no_feature]
print("-"*50)
get_impfeature_names(indices[0], test_df['TEXT'].iloc[test_point_index
],test_df['Gene'].iloc[test_point_index],test_df['Variation'].iloc[test
_point_index], no_feature)
```

```
Predicted Class : 4
Predicted Class Probabilities: [[1.117e-01 6.160e-02 5.100e-03 5.689e-0
1 2.740e-02 3.050e-02 1.915e-01
3.000e-03 1.000e-04]]
```

```
Actual Class : 4
```

```
-----
9 Text feature [proteins] present in test data point [True]
10 Text feature [activity] present in test data point [True]
11 Text feature [protein] present in test data point [True]
12 Text feature [experiments] present in test data point [True]
16 Text feature [determined] present in test data point [True]
18 Text feature [described] present in test data point [True]
21 Text feature [whereas] present in test data point [True]
22 Text feature [loss] present in test data point [True]
23 Text feature [shown] present in test data point [True]
24 Text feature [function] present in test data point [True]
25 Text feature [indicated] present in test data point [True]
28 Text feature [tagged] present in test data point [True]
30 Text feature [results] present in test data point [True]
35 Text feature [levels] present in test data point [True]
```

```
36 Text feature [type] present in test data point [True]
37 Text feature [related] present in test data point [True]
38 Text feature [two] present in test data point [True]
39 Text feature [also] present in test data point [True]
40 Text feature [abrogate] present in test data point [True]
41 Text feature [missense] present in test data point [True]
45 Text feature [mutations] present in test data point [True]
47 Text feature [bind] present in test data point [True]
51 Text feature [containing] present in test data point [True]
52 Text feature [suppressor] present in test data point [True]
53 Text feature [either] present in test data point [True]
54 Text feature [transfected] present in test data point [True]
57 Text feature [30] present in test data point [True]
58 Text feature [although] present in test data point [True]
60 Text feature [wild] present in test data point [True]
63 Text feature [analyzed] present in test data point [True]
65 Text feature [may] present in test data point [True]
69 Text feature [expression] present in test data point [True]
70 Text feature [germline] present in test data point [True]
72 Text feature [performed] present in test data point [True]
73 Text feature [using] present in test data point [True]
75 Text feature [previously] present in test data point [True]
76 Text feature [functional] present in test data point [True]
77 Text feature [thus] present in test data point [True]
78 Text feature [see] present in test data point [True]
81 Text feature [result] present in test data point [True]
82 Text feature [suggest] present in test data point [True]
83 Text feature [associated] present in test data point [True]
84 Text feature [marked] present in test data point [True]
86 Text feature [lower] present in test data point [True]
94 Text feature [cells] present in test data point [True]
96 Text feature [affected] present in test data point [True]
97 Text feature [three] present in test data point [True]
Out of the top 100 features 47 are present in query point
```

4.2. K Nearest Neighbour Classification

4.2.1. Hyper parameter tuning

```
In [65]: # find more about KNeighborsClassifier() here http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
# -----
# default parameter
# KNeighborsClassifier(n_neighbors=5, weights='uniform', algorithm='auto', leaf_size=30, p=2,
# metric='minkowski', metric_params=None, n_jobs=1, **kwargs)

# methods of
# fit(X, y) : Fit the model using X as training data and y as target values
# predict(X): Predict the class labels for the provided data
# predict_proba(X): Return probability estimates for the test data X.
#-----
# video link: https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/k-nearest-neighbors-geometric-intuition-with-a-toy-example-1/
#-----

# find more about CalibratedClassifierCV here at http://scikit-learn.org/stable/modules/generated/sklearn.calibration.CalibratedClassifierCV.html
# -----
# default parameters
# sklearn.calibration.CalibratedClassifierCV(base_estimator=None, method='sigmoid', cv=3)
#
# some of the methods of CalibratedClassifierCV()
# fit(X, y[, sample_weight])    Fit the calibrated model
# get_params([deep])           Get parameters for this estimator.
# predict(X)                    Predict the target of new samples.
# predict_proba(X)              Posterior probabilities of classification
#-----
# video link:
#-----
```

```

alpha = [5, 11, 15, 21, 31, 41, 51, 99]
cv_log_error_array = []
for i in alpha:
    print("for alpha =", i)
    clf = KNeighborsClassifier(n_neighbors=i)
    clf.fit(train_x_responseCoding, train_y)
    sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
    sig_clf.fit(train_x_responseCoding, train_y)
    sig_clf_probs = sig_clf.predict_proba(cv_x_responseCoding)
    cv_log_error_array.append(log_loss(cv_y, sig_clf_probs, labels=clf.
classes_, eps=1e-15))
    # to avoid rounding error while multiplying probabilities we use log
-probability estimates
    print("Log Loss :", log_loss(cv_y, sig_clf_probs))

fig, ax = plt.subplots()
ax.plot(alpha, cv_log_error_array, c='g')
for i, txt in enumerate(np.round(cv_log_error_array, 3)):
    ax.annotate((alpha[i], str(txt)), (alpha[i], cv_log_error_array[i]))
plt.grid()
plt.title("Cross Validation Error for each alpha")
plt.xlabel("Alpha i's")
plt.ylabel("Error measure")
plt.show()

best_alpha = np.argmin(cv_log_error_array)
clf = KNeighborsClassifier(n_neighbors=alpha[best_alpha])
clf.fit(train_x_responseCoding, train_y)
sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
sig_clf.fit(train_x_responseCoding, train_y)

# summarizing data
knn_best_alpha = alpha[best_alpha]
knn_encoding = "Response"

predict_y = sig_clf.predict_proba(train_x_responseCoding)
knn_train_log_loss = log_loss(y_train, predict_y, labels=clf.classes_,

```

```

eps=1e-15)
print('For values of best alpha = ', alpha[best_alpha], "The train log
loss is:", log_loss(y_train, predict_y, labels=clf.classes_, eps=1e-15
))

predict_y = sig_clf.predict_proba(cv_x_responseCoding)
knn_cv_log_loss = log_loss(y_cv, predict_y, labels=clf.classes_, eps=1e
-15)
print('For values of best alpha = ', alpha[best_alpha], "The cross vali
dation log loss is:", log_loss(y_cv, predict_y, labels=clf.classes_, eps
=1e-15))

predict_y = sig_clf.predict_proba(test_x_responseCoding)
knn_test_log_loss = log_loss(y_test, predict_y, labels=clf.classes_, ep
s=1e-15)
print('For values of best alpha = ', alpha[best_alpha], "The test log l
oss is:", log_loss(y_test, predict_y, labels=clf.classes_, eps=1e-15))

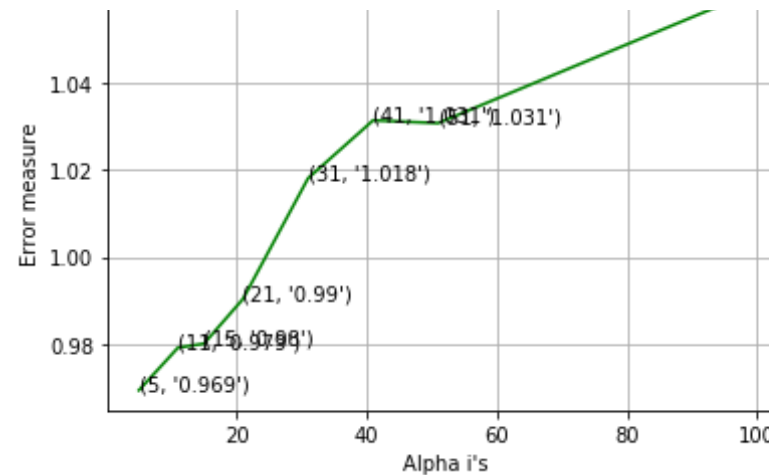
```

```

for alpha = 5
Log Loss : 0.9693899628571674
for alpha = 11
Log Loss : 0.9791744529156299
for alpha = 15
Log Loss : 0.9801005837273411
for alpha = 21
Log Loss : 0.9903768983087803
for alpha = 31
Log Loss : 1.018074523822651
for alpha = 41
Log Loss : 1.0313764700689592
for alpha = 51
Log Loss : 1.0307367494709267
for alpha = 99
Log Loss : 1.060657468275455

```





For values of best alpha = 5 The train log loss is: 0.49105605560947524
 For values of best alpha = 5 The cross validation log loss is: 0.9693899628571674
 For values of best alpha = 5 The test log loss is: 1.0571383119253346

4.2.2. Testing the model with best hyper paramters

In [66]:

```
# find more about KNeighborsClassifier() here http://scikit-learn.org/s
table/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
# -----
# default parameter
# KNeighborsClassifier(n_neighbors=5, weights='uniform', algorithm='aut
o', leaf_size=30, p=2,
# metric='minkowski', metric_params=None, n_jobs=1, **kwargs)

# methods of
# fit(X, y) : Fit the model using X as training data and y as target va
lues
# predict(X):Predict the class labels for the provided data
# predict_proba(X):Return probability estimates for the test data X.
#-----
# video link: https://www.appliedaicourse.com/course/applied-ai-course-
```

```
online/lessons/k-nearest-neighbors-geometric-intuition-with-a-toy-example-1/
#-----
clf = KNeighborsClassifier(n_neighbors=alpha[best_alpha])
knn_misclassified = predict_and_plot_confusion_matrix(train_x_responseC
oding, train_y, cv_x_responseCoding, cv_y, clf)
```

Log loss : 0.9693899628571674

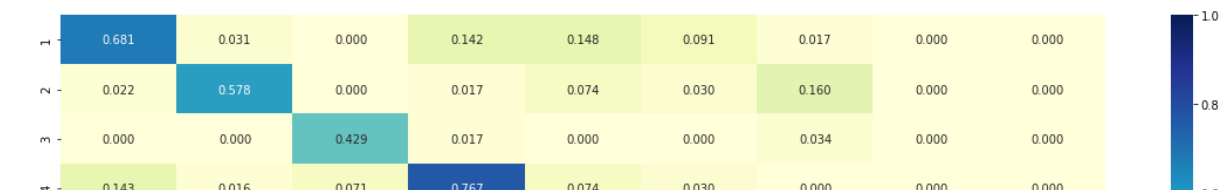
Number of mis-classified points : 0.31203007518796994

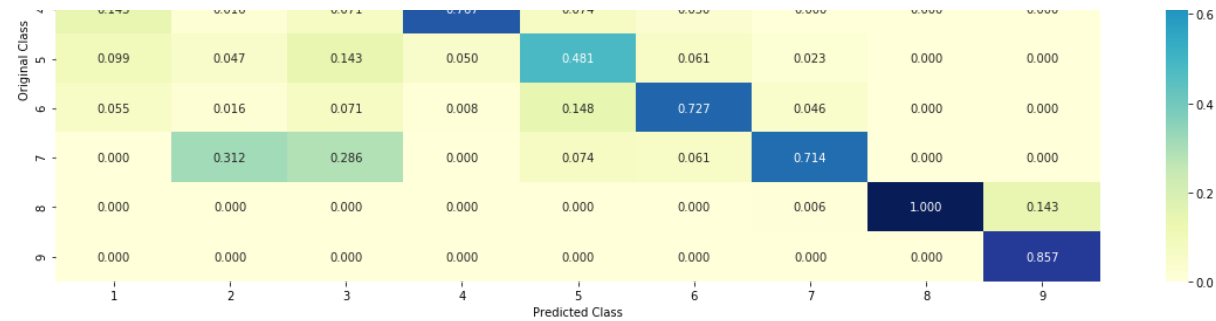
----- Confusion matrix -----



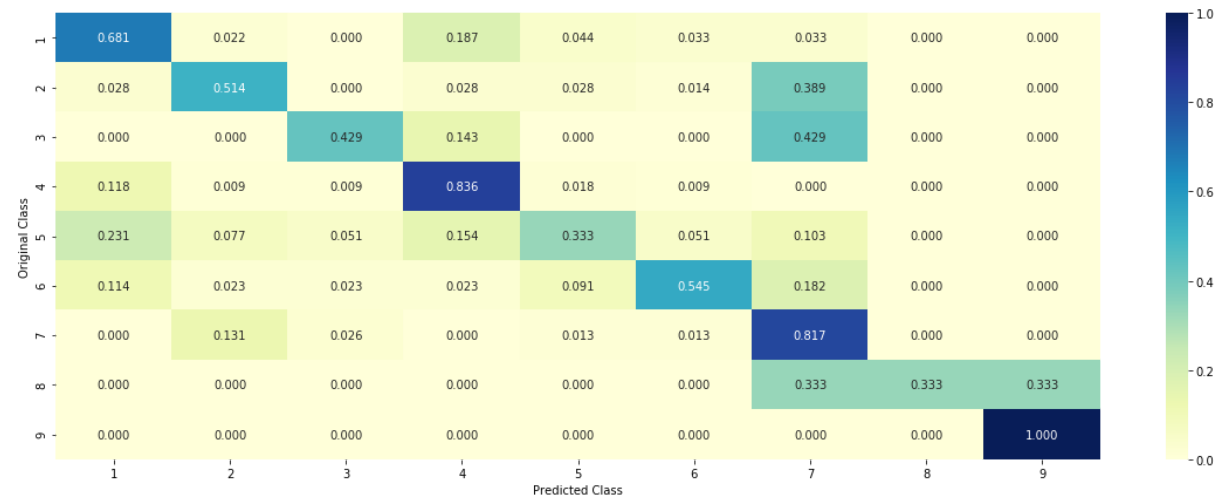
----- Precision matrix (Column Sum=1) -----

--





----- Recall matrix (Row sum=1) -----



4.2.3. Sample Query point -1

```
In [67]: clf = KNeighborsClassifier(n_neighbors=alpha[best_alpha])
clf.fit(train_x_responseCoding, train_y)
sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
sig_clf.fit(train_x_responseCoding, train_y)

test_point_index = 1
predicted_cls = sig_clf.predict(test_x_responseCoding[0].reshape(1,-1))
print("Predicted Class :", predicted_cls[0])
```

```
print("Actual Class :", test_y[test_point_index])
neighbors = clf.kneighbors(test_x_responseCoding[test_point_index].reshape(1, -1), alpha[best_alpha])
print("The ", alpha[best_alpha], " nearest neighbours of the test points belongs to classes", train_y[neighbors[1][0]])
print("Frequency of nearest points :", Counter(train_y[neighbors[1][0]]))
```

Predicted Class : 4
 Actual Class : 1
 The 5 nearest neighbours of the test points belongs to classes [1 6 5 6 1]
 Frequency of nearest points : Counter({1: 2, 6: 2, 5: 1})

4.2.4. Sample Query Point-2

```
In [68]: clf = KNeighborsClassifier(n_neighbors=alpha[best_alpha])
clf.fit(train_x_responseCoding, train_y)
sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
sig_clf.fit(train_x_responseCoding, train_y)

test_point_index = 100

predicted_cls = sig_clf.predict(test_x_responseCoding[test_point_index].reshape(1, -1))
print("Predicted Class :", predicted_cls[0])
print("Actual Class :", test_y[test_point_index])
neighbors = clf.kneighbors(test_x_responseCoding[test_point_index].reshape(1, -1), alpha[best_alpha])
print("the k value for knn is", alpha[best_alpha], "and the nearest neighbours of the test points belongs to classes", train_y[neighbors[1][0]])
print("Frequency of nearest points :", Counter(train_y[neighbors[1][0]]))
```

Predicted Class : 4
 Actual Class : 4
 the k value for knn is 5 and the nearest neighbours of the test points belongs to classes [4 4 4 4 1]
 Frequency of nearest points : Counter({4: 4, 1: 1})

4.3. Logistic Regression

4.3.1. With Class balancing

4.3.1.1. Hyper paramter tuning

```
In [69]: # read more about SGDClassifier() at http://scikit-learn.org/stable/modules/generated/sklearn.linear\_model.SGDClassifier.html
# -----
# default parameters
# SGDClassifier(loss='hinge', penalty='l2', alpha=0.0001, l1_ratio=0.15, fit_intercept=True, max_iter=None, tol=None,
# shuffle=True, verbose=0, epsilon=0.1, n_jobs=1, random_state=None, learning_rate='optimal', eta0=0.0, power_t=0.5,
# class_weight=None, warm_start=False, average=False, n_iter=None)

# some of methods
# fit(X, y[, coef_init, intercept_init, ...])      Fit linear model with Stochastic Gradient Descent.
# predict(X)      Predict class labels for samples in X.

#-----
# video link: https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/geometric-intuition-1/
#-----

# find more about CalibratedClassifierCV here at http://scikit-learn.org/stable/modules/generated/sklearn.calibration.CalibratedClassifierCV.html
# -----
# default paramters
# sklearn.calibration.CalibratedClassifierCV(base_estimator=None, method='sigmoid', cv=3)
#
```

```

# some of the methods of CalibratedClassifierCV()
# fit(X, y[, sample_weight])    Fit the calibrated model
# get_params([deep])           Get parameters for this estimator.
# predict(X)                    Predict the target of new samples.
# predict_proba(X)              Posterior probabilities of classification
#-----
# video link:
#-----

alpha = [10 ** x for x in range(-6, 3)]
cv_log_error_array = []
for i in alpha:
    print("for alpha =", i)
    clf = SGDClassifier(class_weight='balanced', alpha=i, penalty='l2',
loss='log', random_state=42)
    clf.fit(train_x_onehotCoding, train_y)
    sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
    sig_clf.fit(train_x_onehotCoding, train_y)
    sig_clf_probs = sig_clf.predict_proba(cv_x_onehotCoding)
    cv_log_error_array.append(log_loss(cv_y, sig_clf_probs, labels=clf.
classes_, eps=1e-15))
    # to avoid rounding error while multiplying probabilitites we use log
-probability estimates
    print("Log Loss :", log_loss(cv_y, sig_clf_probs))

fig, ax = plt.subplots()
ax.plot(alpha, cv_log_error_array, c='g')
for i, txt in enumerate(np.round(cv_log_error_array,3)):
    ax.annotate((alpha[i],str(txt)), (alpha[i],cv_log_error_array[i]))
plt.grid()
plt.title("Cross Validation Error for each alpha")
plt.xlabel("Alpha i's")
plt.ylabel("Error measure")
plt.show()

best_alpha = np.argmin(cv_log_error_array)
clf = SGDClassifier(class_weight='balanced', alpha=alpha[best_alpha], p
enalty='l2', loss='log', random_state=42)

```

```

clf.fit(train_x_onehotCoding, train_y)
sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
sig_clf.fit(train_x_onehotCoding, train_y)

lr_bal_best_alpha = alpha[best_alpha]
lr_bal_encoding = "One hot"

predict_y = sig_clf.predict_proba(train_x_onehotCoding)
lr_bal_train_log_loss = log_loss(y_train, predict_y, labels=clf.classes_,
_, eps=1e-15)
print('For values of best alpha = ', alpha[best_alpha], "The train log
loss is:", log_loss(y_train, predict_y, labels=clf.classes_, eps=1e-15
))

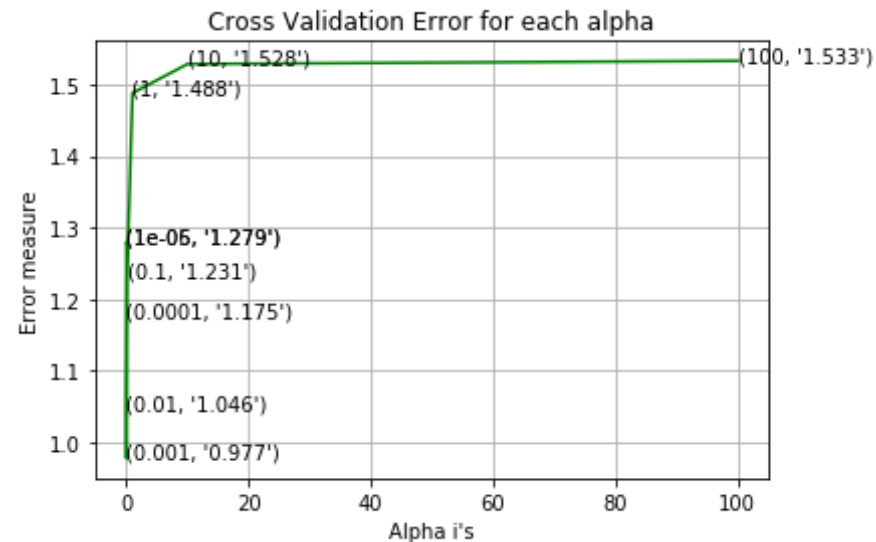
predict_y = sig_clf.predict_proba(cv_x_onehotCoding)
lr_bal_cv_log_loss = log_loss(y_cv, predict_y, labels=clf.classes_, eps
=1e-15)
print('For values of best alpha = ', alpha[best_alpha], "The cross vali
dation log loss is:", log_loss(y_cv, predict_y, labels=clf.classes_, eps
=1e-15))

predict_y = sig_clf.predict_proba(test_x_onehotCoding)
lr_bal_test_log_loss = log_loss(y_test, predict_y, labels=clf.classes_,
eps=1e-15)
print('For values of best alpha = ', alpha[best_alpha], "The test log l
oss is:", log_loss(y_test, predict_y, labels=clf.classes_, eps=1e-15))

for alpha = 1e-06
Log Loss : 1.2786797981678568
for alpha = 1e-05
Log Loss : 1.2791916824138063
for alpha = 0.0001
Log Loss : 1.1752371243044615
for alpha = 0.001
Log Loss : 0.9772975521957078
for alpha = 0.01
Log Loss : 1.0458255068863063
for alpha = 0.1

```

Log Loss : 1.231011479069885
 for alpha = 1
 Log Loss : 1.4878197060769986
 for alpha = 10
 Log Loss : 1.5284410707312621
 for alpha = 100
 Log Loss : 1.5329125960878465



For values of best alpha = 0.001 The train log loss is: 0.5797808831760713
 For values of best alpha = 0.001 The cross validation log loss is: 0.9772975521957078
 For values of best alpha = 0.001 The test log loss is: 1.098628782919

4.3.1.2. Testing the model with best hyper paramters

```
In [70]: # read more about SGDClassifier() at http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html
# -----
# default parameters
# SGDClassifier(loss='hinge', penalty='l2', alpha=0.0001, l1_ratio=0.1
```

```

5, fit_intercept=True, max_iter=None, tol=None,
# shuffle=True, verbose=0, epsilon=0.1, n_jobs=1, random_state=None, le
arning_rate='optimal', eta0=0.0, power_t=0.5,
# class_weight=None, warm_start=False, average=False, n_iter=None)

# some of methods
# fit(X, y[, coef_init, intercept_init, ...])    Fit linear model with S
tochastic Gradient Descent.
# predict(X)    Predict class labels for samples in X.

#-----
# video link: https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/geometric-intuition-1/
#-----
clf = SGDClassifier(class_weight='balanced', alpha=alpha[best_alpha], p
enalty='l2', loss='log', random_state=42)
lr_bal_misclassified = predict_and_plot_confusion_matrix(train_x_onehot
Coding, train_y, cv_x_onehotCoding, cv_y, clf)

```

Log loss : 0.9772975521957078

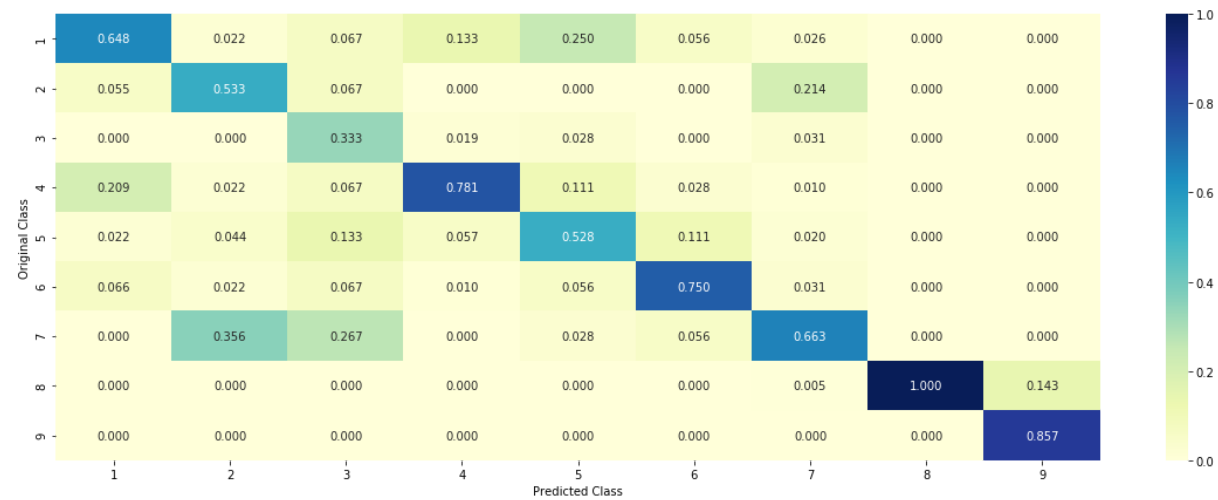
Number of mis-classified points : 0.33646616541353386

----- Confusion matrix -----

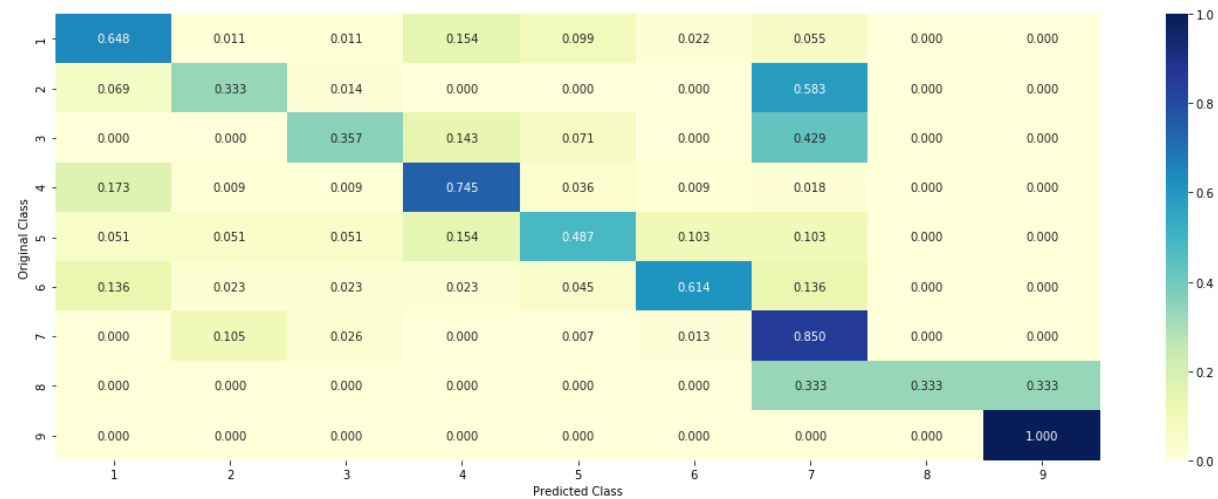


----- Precision matrix (Column Sum=1) -----

--



----- Recall matrix (Row sum=1) -----



4.3.1.3. Feature Importance

```
In [71]: def get_imp_feature_names(text, indices, removed_ind = []):
          word_present = 0
```



```

tabulte_list = []
increasingorder_ind = 0
for i in indices:
    if i < train_gene_feature_onehotCoding.shape[1]:
        tabulte_list.append([increasingorder_ind, "Gene", "Yes"])
    elif i < 18:
        tabulte_list.append([increasingorder_ind, "Variation", "Yes"
])

    if ((i > 17) & (i not in removed_ind)) :
        word = train_text_features[i]
        yes_no = True if word in text.split() else False
        if yes_no:
            word_present += 1
            tabulte_list.append([increasingorder_ind, train_text_features
[i], yes_no])
            increasingorder_ind += 1
        print(word_present, "most important features are present in our que
ry point")
        print("-"*50)
        print("The features that are most important of the ", predicted_cls[
0], " class:")
        print(tabulate(tabulte_list, headers=["Index", 'Feature name', 'Pre
sent or Not']))

```

4.3.1.3.1. Correctly Classified point

```

In [72]: # from tabulate import tabulate
clf = SGDClassifier(class_weight='balanced', alpha=alpha[best_alpha], p
enalty='l2', loss='log', random_state=42)
clf.fit(train_x_onehotCoding, train_y)
test_point_index = 1
no_feature = 500
predicted_cls = sig_clf.predict(test_x_onehotCoding[test_point_index])
print("Predicted Class :", predicted_cls[0])
print("Predicted Class Probabilities:", np.round(sig_clf.predict_proba(
test_x_onehotCoding[test_point_index]), 4))
print("Actual Class :", test_y[test_point_index])
indices = np.argsort(-clf.coef_)[predicted_cls-1][:, :no_feature]

```

```
print("-"*50)
get_impfeature_names(indices[0], test_df['TEXT'].iloc[test_point_index],
test_df['Gene'].iloc[test_point_index],test_df['Variation'].iloc[test_point_index], no_feature)
```

```
Predicted Class : 1
Predicted Class Probabilities: [[6.071e-01 1.050e-02 1.030e-02 8.640e-0
2 2.274e-01 5.360e-02 2.100e-03
2.100e-03 5.000e-04]]
Actual Class : 1
```

```
-----
172 Text feature [immobilized] present in test data point [True]
290 Text feature [mapping] present in test data point [True]
441 Text feature [gives] present in test data point [True]
450 Text feature [absolutely] present in test data point [True]
Out of the top 500 features 4 are present in query point
```

4.3.1.3.2. Incorrectly Classified point

```
In [73]: test_point_index = 100
no_feature = 500
predicted_cls = sig_clf.predict(test_x_onehotCoding[test_point_index])
print("Predicted Class :", predicted_cls[0])
print("Predicted Class Probabilities:", np.round(sig_clf.predict_proba(
test_x_onehotCoding[test_point_index]),4))
print("Actual Class :", test_y[test_point_index])
indices = np.argsort(-clf.coef_)[predicted_cls-1][:,:no_feature]
print("-"*50)
get_impfeature_names(indices[0], test_df['TEXT'].iloc[test_point_index],
test_df['Gene'].iloc[test_point_index],test_df['Variation'].iloc[test_point_index], no_feature)
```

```
Predicted Class : 4
Predicted Class Probabilities: [[1.470e-02 1.820e-02 5.100e-03 9.358e-0
1 9.400e-03 4.300e-03 1.030e-02
1.400e-03 8.000e-04]]
Actual Class : 4
```

```
-----
111 Text feature [suppressor] present in test data point [True]
```

```
111 Text feature [suppressor] present in test data point [True]
139 Text feature [edges] present in test data point [True]
150 Text feature [germline] present in test data point [True]
176 Text feature [tagged] present in test data point [True]
238 Text feature [nonsense] present in test data point [True]
308 Text feature [heterozygosity] present in test data point [True]
317 Text feature [tgfbrii] present in test data point [True]
366 Text feature [emsa] present in test data point [True]
438 Text feature [tgf] present in test data point [True]
Out of the top 500 features 9 are present in query point
```

4.3.2. Without Class balancing

4.3.2.1. Hyper paramter tuning

```
In [74]: # read more about SGDClassifier() at http://scikit-learn.org/stable/modules/generated/sklearn.linear\_model.SGDClassifier.html
# -----
# default parameters
# SGDClassifier(loss='hinge', penalty='l2', alpha=0.0001, l1_ratio=0.15, fit_intercept=True, max_iter=None, tol=None,
# shuffle=True, verbose=0, epsilon=0.1, n_jobs=1, random_state=None, learning_rate='optimal', eta0=0.0, power_t=0.5,
# class_weight=None, warm_start=False, average=False, n_iter=None)

# some of methods
# fit(X, y[, coef_init, intercept_init, ...])      Fit linear model with Stochastic Gradient Descent.
# predict(X)      Predict class labels for samples in X.

#-----
# video link: https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/geometric-intuition-1/
#-----
```

```

# find more about CalibratedClassifierCV here at http://scikit-learn.org/stable/modules/generated/sklearn.calibration.CalibratedClassifierCV.html
# -----
# default paramters
# sklearn.calibration.CalibratedClassifierCV(base_estimator=None, method='sigmoid', cv=3)
#
# some of the methods of CalibratedClassifierCV()
# fit(X, y[, sample_weight])    Fit the calibrated model
# get_params([deep])           Get parameters for this estimator.
# predict(X)                    Predict the target of new samples.
# predict_proba(X)              Posterior probabilities of classification
#-----
# video link:
#-----

alpha = [10 ** x for x in range(-6, 1)]
cv_log_error_array = []
for i in alpha:
    print("for alpha =", i)
    clf = SGDClassifier(alpha=i, penalty='l2', loss='log', random_state=42)
    clf.fit(train_x_onehotCoding, train_y)
    sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
    sig_clf.fit(train_x_onehotCoding, train_y)
    sig_clf_probs = sig_clf.predict_proba(cv_x_onehotCoding)
    cv_log_error_array.append(log_loss(cv_y, sig_clf_probs, labels=clf.classes_, eps=1e-15))
    print("Log Loss :", log_loss(cv_y, sig_clf_probs))

fig, ax = plt.subplots()
ax.plot(alpha, cv_log_error_array, c='g')
for i, txt in enumerate(np.round(cv_log_error_array, 3)):
    ax.annotate((alpha[i], str(txt)), (alpha[i], cv_log_error_array[i]))
plt.grid()
plt.title("Cross Validation Error for each alpha")
plt.xlabel("Alpha i's")
plt.ylabel("Error measure")

```

```

plt.show()

best_alpha = np.argmin(cv_log_error_array)
clf = SGDClassifier(alpha=alpha[best_alpha], penalty='l2', loss='log',
random_state=42)
clf.fit(train_x_onehotCoding, train_y)
sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
sig_clf.fit(train_x_onehotCoding, train_y)

# summarizing data
lr_best_alpha = alpha[best_alpha]
lr_encoding = "one hot"

predict_y = sig_clf.predict_proba(train_x_onehotCoding)
lr_train_log_loss = log_loss(y_train, predict_y, labels=clf.classes_, eps=1e-15)
print('For values of best alpha = ', alpha[best_alpha], "The train log
loss is:", log_loss(y_train, predict_y, labels=clf.classes_, eps=1e-15
))

predict_y = sig_clf.predict_proba(cv_x_onehotCoding)
lr_cv_log_loss = log_loss(y_cv, predict_y, labels=clf.classes_, eps=1e-15)
print('For values of best alpha = ', alpha[best_alpha], "The cross vali
dation log loss is:", log_loss(y_cv, predict_y, labels=clf.classes_, eps
=1e-15))

predict_y = sig_clf.predict_proba(test_x_onehotCoding)
lr_test_log_loss = log_loss(y_test, predict_y, labels=clf.classes_, eps
=1e-15)
print('For values of best alpha = ', alpha[best_alpha], "The test log l
oss is:", log_loss(y_test, predict_y, labels=clf.classes_, eps=1e-15))

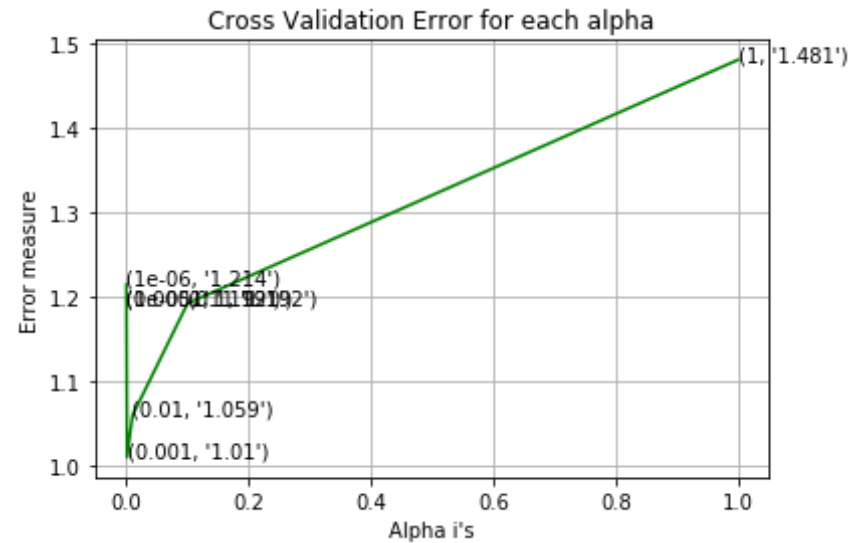
for alpha = 1e-06
Log Loss : 1.2142745781252415
for alpha = 1e-05
Log Loss : 1.1920588230997877

```

```

Log Loss : 1.192090230092077
for alpha = 0.0001
Log Loss : 1.1909043293818675
for alpha = 0.001
Log Loss : 1.009614118649886
for alpha = 0.01
Log Loss : 1.0592274136432778
for alpha = 0.1
Log Loss : 1.191909214215443
for alpha = 1
Log Loss : 1.4806178107122772

```



For values of best alpha = 0.001 The train log loss is: 0.5745933764493746
 For values of best alpha = 0.001 The cross validation log loss is: 1.009614118649886
 For values of best alpha = 0.001 The test log loss is: 1.1088806001836657

4.3.2.2. Testing model with best hyper parameters

In [75]: `# read more about SGDClassifier() at http://scikit-learn.org/stable/mod`

```

ules/generated/sklearn.linear_model.SGDClassifier.html
# -----
# default parameters
# SGDClassifier(loss='hinge', penalty='l2', alpha=0.0001, l1_ratio=0.1
5, fit_intercept=True, max_iter=None, tol=None,
# shuffle=True, verbose=0, epsilon=0.1, n_jobs=1, random_state=None, le
arning_rate='optimal', eta0=0.0, power_t=0.5,
# class_weight=None, warm_start=False, average=False, n_iter=None)

# some of methods
# fit(X, y[, coef_init, intercept_init, ...])      Fit linear model with S
tochastic Gradient Descent.
# predict(X)      Predict class labels for samples in X.

#-----
# video link:
#-----

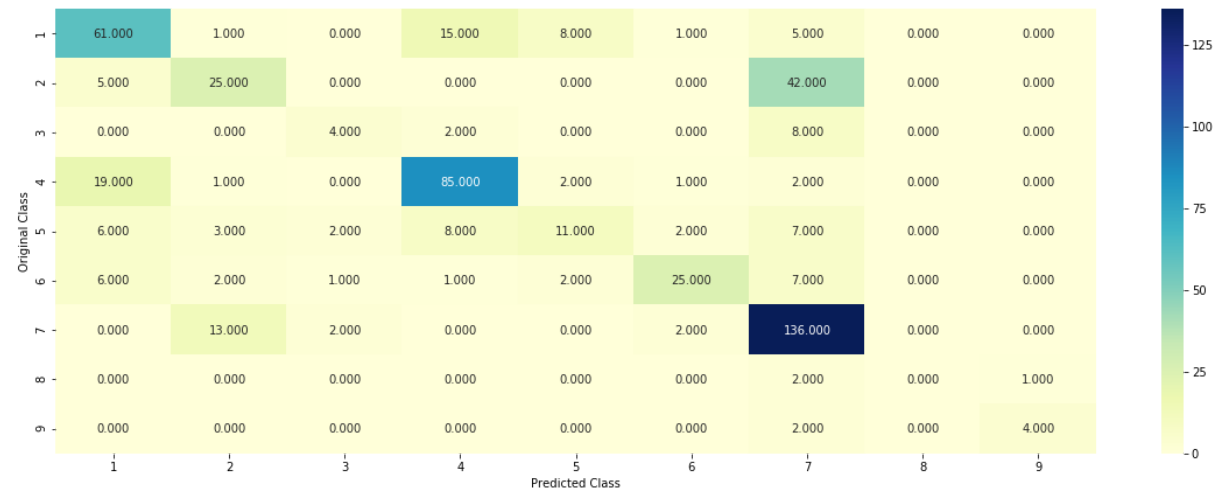
clf = SGDClassifier(alpha=alpha[best_alpha], penalty='l2', loss='log',
random_state=42)
lr_misclassified = predict_and_plot_confusion_matrix(train_x_onehotCodi
ng, train_y, cv_x_onehotCoding, cv_y, clf)

```

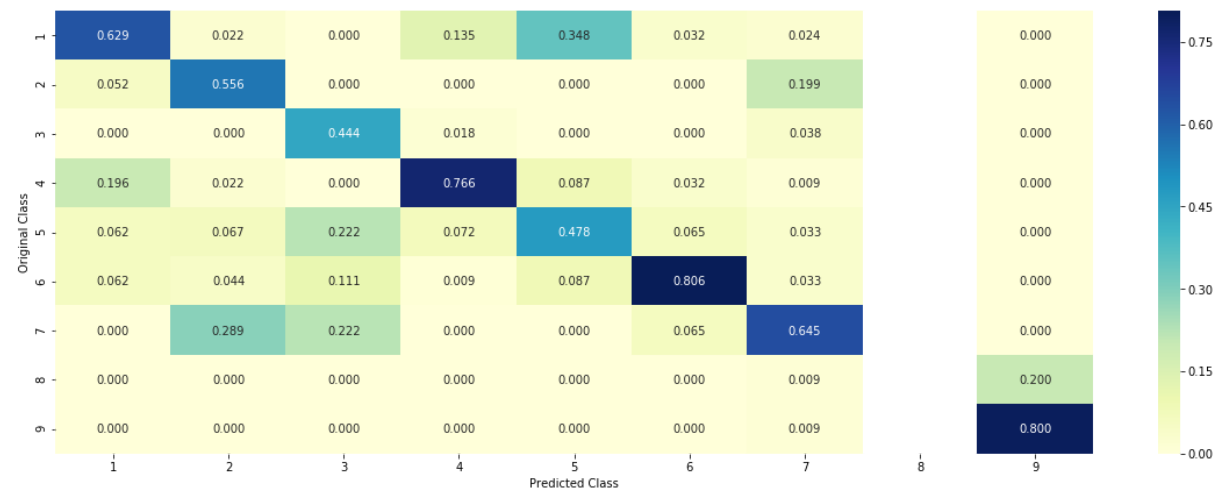
Log loss : 1.009614118649886

Number of mis-classified points : 0.34022556390977443

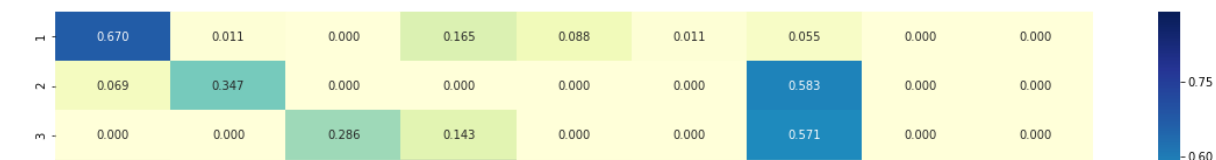
----- Confusion matrix -----

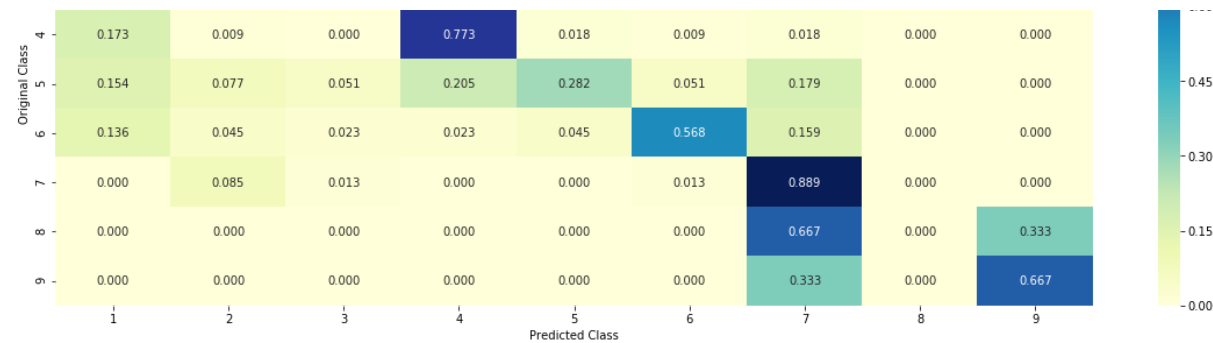


----- Precision matrix (Column Sum=1) -----



----- Recall matrix (Row sum=1) -----





4.3.2.3. Feature Importance, Correctly Classified point

```
In [76]: clf = SGDClassifier(alpha=alpha[best_alpha], penalty='l2', loss='log',
        random_state=42)
        clf.fit(train_x_onehotCoding,train_y)
        test_point_index = 1
        no_feature = 500
        predicted_cls = sig_clf.predict(test_x_onehotCoding[test_point_index])
        print("Predicted Class :", predicted_cls[0])
        print("Predicted Class Probabilities:", np.round(sig_clf.predict_proba(
        test_x_onehotCoding[test_point_index]),4))
        print("Actual Class :", test_y[test_point_index])
        indices = np.argsort(-clf.coef_)[predicted_cls-1][:,:no_feature]
        print("-"*50)
        get_impfeature_names(indices[0], test_df['TEXT'].iloc[test_point_index]
        ],test_df['Gene'].iloc[test_point_index],test_df['Variation'].iloc[test
        _point_index], no_feature)
```

```
Predicted Class : 1
Predicted Class Probabilities: [[6.949e-01 9.600e-03 7.000e-04 1.271e-0
1 1.381e-01 2.450e-02 5.000e-03
1.000e-04 0.000e+00]]
Actual Class : 1
-----
193 Text feature [immobilized] present in test data point [True]
311 Text feature [mapping] present in test data point [True]
450 Text feature [valuable] present in test data point [True]
```

491 Text feature [gives] present in test data point [True]
Out of the top 500 features 4 are present in query point

4.3.2.4. Feature Importance, Inorrectly Classified point

```
In [77]: test_point_index = 100
no_feature = 500
predicted_cls = sig_clf.predict(test_x_onehotCoding[test_point_index])
print("Predicted Class :", predicted_cls[0])
print("Predicted Class Probabilities:", np.round(sig_clf.predict_proba(
test_x_onehotCoding[test_point_index]),4))
print("Actual Class :", test_y[test_point_index])
indices = np.argsort(-clf.coef_)[predicted_cls-1][:,no_feature]
print("-"*50)
get_impfeature_names(indices[0], test_df['TEXT'].iloc[test_point_index
],test_df['Gene'].iloc[test_point_index],test_df['Variation'].iloc[test
_point_index], no_feature)
```

```
Predicted Class : 4
Predicted Class Probabilities: [[1.420e-02 1.940e-02 1.200e-03 9.374e-0
1 7.000e-03 3.500e-03 1.670e-02
 4.000e-04 1.000e-04]]
Actual Class : 4
```

```
-----
151 Text feature [suppressor] present in test data point [True]
161 Text feature [germline] present in test data point [True]
183 Text feature [tagged] present in test data point [True]
189 Text feature [edges] present in test data point [True]
266 Text feature [keratinocyte] present in test data point [True]
288 Text feature [tgfbrii] present in test data point [True]
333 Text feature [heterozygosity] present in test data point [True]
335 Text feature [nonsense] present in test data point [True]
412 Text feature [tgf] present in test data point [True]
419 Text feature [emsa] present in test data point [True]
Out of the top 500 features 10 are present in query point
```

4.4. Linear Support Vector Machines

4.4.1. Hyper paramter tuning

```
In [78]: # read more about support vector machines with linear kernals here http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

# -----
# default parameters
# SVC(C=1.0, kernel='rbf', degree=3, gamma='auto', coef0=0.0, shrinking
# =True, probability=False, tol=0.001,
# cache_size=200, class_weight=None, verbose=False, max_iter=-1, decisi
# on_function_shape='ovr', random_state=None)

# Some of methods of SVM()
# fit(X, y, [sample_weight])    Fit the SVM model according to the give
# n training data.
# predict(X)    Perform classification on samples in X.
# -----
# video link: https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/mathematical-derivation-copy-8/
# -----

# find more about CalibratedClassifierCV here at http://scikit-learn.org/stable/modules/generated/sklearn.calibration.CalibratedClassifierCV.html
# -----
# default paramters
# sklearn.calibration.CalibratedClassifierCV(base_estimator=None, metho
# d='sigmoid', cv=3)
#
# some of the methods of CalibratedClassifierCV()
# fit(X, y[, sample_weight])    Fit the calibrated model
# get_params([deep])    Get parameters for this estimator.
# predict(X)    Predict the target of new samples.
# predict_proba(X)    Posterior probabilities of classification
#-----
```

```

# video link:
#-----

alpha = [10 ** x for x in range(-5, 3)]
cv_log_error_array = []
for i in alpha:
    print("for C =", i)
    # clf = SVC(C=i, kernel='linear', probability=True, class_weight='balanced')
    clf = SGDClassifier(class_weight='balanced', alpha=i, penalty='l2',
                        loss='hinge', random_state=42)
    clf.fit(train_x_onehotCoding, train_y)
    sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
    sig_clf.fit(train_x_onehotCoding, train_y)
    sig_clf_probs = sig_clf.predict_proba(cv_x_onehotCoding)
    cv_log_error_array.append(log_loss(cv_y, sig_clf_probs, labels=clf.
classes_, eps=1e-15))
    print("Log Loss :", log_loss(cv_y, sig_clf_probs))

fig, ax = plt.subplots()
ax.plot(alpha, cv_log_error_array, c='g')
for i, txt in enumerate(np.round(cv_log_error_array, 3)):
    ax.annotate((alpha[i], str(txt)), (alpha[i], cv_log_error_array[i]))
plt.grid()
plt.title("Cross Validation Error for each alpha")
plt.xlabel("Alpha i's")
plt.ylabel("Error measure")
plt.show()

best_alpha = np.argmin(cv_log_error_array)
# clf = SVC(C=i, kernel='linear', probability=True, class_weight='balanced')
clf = SGDClassifier(class_weight='balanced', alpha=alpha[best_alpha],
                    penalty='l2', loss='hinge', random_state=42)
clf.fit(train_x_onehotCoding, train_y)
sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
sig_clf.fit(train_x_onehotCoding, train_y)

```

```

# summarizing data
svm_best_alpha = alpha[best_alpha]
svm_encoding = "one hot"

predict_y = sig_clf.predict_proba(train_x_onehotCoding)
svm_train_log_loss = log_loss(y_train, predict_y, labels=clf.classes_,
eps=1e-15)
print('For values of best alpha = ', alpha[best_alpha], "The train log
loss is:", log_loss(y_train, predict_y, labels=clf.classes_, eps=1e-15
))

predict_y = sig_clf.predict_proba(cv_x_onehotCoding)
svm_cv_log_loss = log_loss(y_cv, predict_y, labels=clf.classes_, eps=1e
-15)
print('For values of best alpha = ', alpha[best_alpha], "The cross vali
dation log loss is:", log_loss(y_cv, predict_y, labels=clf.classes_, eps
=1e-15))

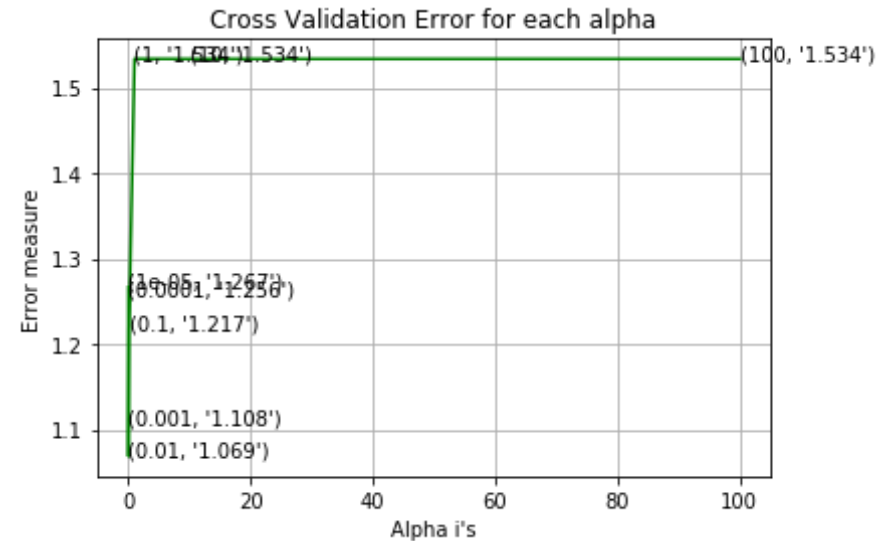
predict_y = sig_clf.predict_proba(test_x_onehotCoding)
svm_test_log_loss = log_loss(y_test, predict_y, labels=clf.classes_, ep
s=1e-15)
print('For values of best alpha = ', alpha[best_alpha], "The test log l
oss is:", log_loss(y_test, predict_y, labels=clf.classes_, eps=1e-15))

for C = 1e-05
Log Loss : 1.266540606806883
for C = 0.0001
Log Loss : 1.2564797967437902
for C = 0.001
Log Loss : 1.107789544075327
for C = 0.01
Log Loss : 1.0685021782755935
for C = 0.1
Log Loss : 1.2174209860621685
for C = 1
Log Loss : 1.5339399133809912
for C = 10
Log Loss : 1.533838956301563

```

TOR C = 100

Log Loss : 1.5338389716064285



For values of best alpha = 0.01 The train log loss is: 0.723913870452315

For values of best alpha = 0.01 The cross validation log loss is: 1.0685021782755935

For values of best alpha = 0.01 The test log loss is: 1.1624786111909287

4.4.2. Testing model with best hyper parameters

```
In [79]: # read more about support vector machines with linear kernels here http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

# -----
# default parameters
# SVC(C=1.0, kernel='rbf', degree=3, gamma='auto', coef0=0.0, shrinking
# =True, probability=False, tol=0.001,
# cache_size=200, class_weight=None, verbose=False, max_iter=-1, decisi
# on_function_shape='ovr', random_state=None)
```

```
# Some of methods of SVM()
# fit(X, y, [sample_weight])    Fit the SVM model according to the give
n training data.
# predict(X)    Perform classification on samples in X.
# -----
# video link: https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/mathematical-derivation-copy-8/
# -----

# clf = SVC(C=alpha[best_alpha],kernel='linear',probability=True, class
_weight='balanced')
clf = SGDClassifier(alpha=alpha[best_alpha], penalty='l2', loss='hinge'
, random_state=42,class_weight='balanced')
svm_misclassified = predict_and_plot_confusion_matrix(train_x_onehotCod
ing, train_y,cv_x_onehotCoding,cv_y, clf)
```

Log loss : 1.0685021782755935

Number of mis-classified points : 0.32142857142857145

----- Confusion matrix -----

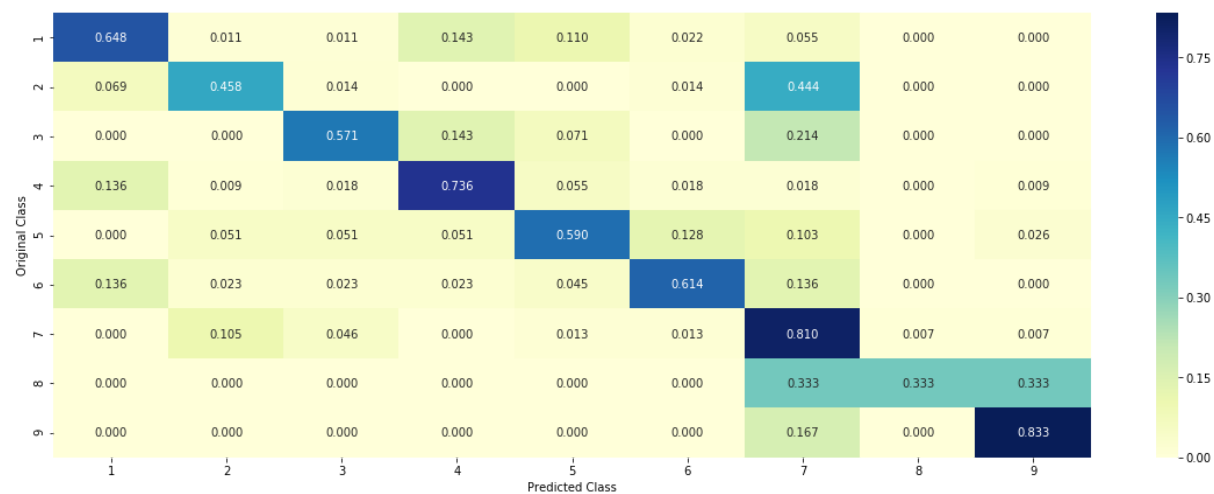


----- Precision matrix (Column Sum=1) -----

--



----- Recall matrix (Row sum=1) -----



4.3.3. Feature Importance

4.3.3.1. For Correctly classified point


```
In [80]: clf = SGDClassifier(alpha=alpha[best_alpha], penalty='l2', loss='hinge'
, random_state=42)
clf.fit(train_x_onehotCoding,train_y)
test_point_index = 1
# test_point_index = 100
no_feature = 500
predicted_cls = sig_clf.predict(test_x_onehotCoding[test_point_index])
print("Predicted Class :", predicted_cls[0])
print("Predicted Class Probabilities:", np.round(sig_clf.predict_proba(
test_x_onehotCoding[test_point_index]),4))
print("Actual Class :", test_y[test_point_index])
indices = np.argsort(-clf.coef_)[predicted_cls-1][:,no_feature]
print("-"*50)
get_impfeature_names(indices[0], test_df['TEXT'].iloc[test_point_index
],test_df['Gene'].iloc[test_point_index],test_df['Variation'].iloc[test
_point_index], no_feature)
```

```
Predicted Class : 5
Predicted Class Probabilities: [[0.2051 0.0777 0.019  0.1495 0.392  0.0
573 0.0898 0.0004 0.0092]]
```

```
Actual Class : 1
```

```
-----
36 Text feature [aspartic] present in test data point [True]
39 Text feature [v1833m] present in test data point [True]
390 Text feature [polymorphisms] present in test data point [True]
423 Text feature [bioinformatics] present in test data point [True]
456 Text feature [classifying] present in test data point [True]
Out of the top 500 features 5 are present in query point
```

4.3.3.2. For Incorrectly classified point

```
In [81]: test_point_index = 100
no_feature = 500
predicted_cls = sig_clf.predict(test_x_onehotCoding[test_point_index])
print("Predicted Class :", predicted_cls[0])
print("Predicted Class Probabilities:", np.round(sig_clf.predict_proba(
test_x_onehotCoding[test_point_index]),4))
print("Actual Class :", test_y[test_point_index])
```

```
indices = np.argsort(-clf.coef_)[predicted_cls-1][:,:no_feature]
print("-"*50)
get_impfeature_names(indices[0], test_df['TEXT'].iloc[test_point_index],
test_df['Gene'].iloc[test_point_index],test_df['Variation'].iloc[test_point_index], no_feature)
```

Predicted Class : 4

Predicted Class Probabilities: [[3.240e-02 1.320e-02 6.300e-03 8.988e-01
2.000e-02 9.100e-03 1.740e-02
6.000e-04 2.300e-03]]

Actual Class : 4

```
-----
40 Text feature [germline] present in test data point [True]
58 Text feature [nonsense] present in test data point [True]
76 Text feature [heterozygosity] present in test data point [True]
77 Text feature [tagged] present in test data point [True]
81 Text feature [suppressor] present in test data point [True]
84 Text feature [edges] present in test data point [True]
99 Text feature [emsa] present in test data point [True]
116 Text feature [familial] present in test data point [True]
209 Text feature [sequential] present in test data point [True]
327 Text feature [denmark] present in test data point [True]
349 Text feature [undertook] present in test data point [True]
351 Text feature [localization] present in test data point [True]
355 Text feature [tgf] present in test data point [True]
431 Text feature [lod] present in test data point [True]
498 Text feature [families] present in test data point [True]
Out of the top 500 features 15 are present in query point
```

4.5 Random Forest Classifier

4.5.1. Hyper paramter tuning (With One hot Encoding)

```
In [82]: # -----
# default parameters
# sklearn.ensemble.RandomForestClassifier(n_estimators=10, criterion='g
```

```

ini', max_depth=None, min_samples_split=2,
# min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='auto',
# max_leaf_nodes=None, min_impurity_decrease=0.0,
# min_impurity_split=None, bootstrap=True, oob_score=False, n_jobs=1, random_state=None, verbose=0, warm_start=False,
# class_weight=None)

# Some of methods of RandomForestClassifier()
# fit(X, y, [sample_weight])    Fit the SVM model according to the given training data.
# predict(X)    Perform classification on samples in X.
# predict_proba(X)    Perform classification on samples in X.

# some of attributes of RandomForestClassifier()
# feature_importances_ : array of shape = [n_features]
# The feature importances (the higher, the more important the feature).

# -----
# video link: https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/random-forest-and-their-construction-2/
# -----

# find more about CalibratedClassifierCV here at http://scikit-learn.org/stable/modules/generated/sklearn.calibration.CalibratedClassifierCV.html
# -----
# default parameters
# sklearn.calibration.CalibratedClassifierCV(base_estimator=None, method='sigmoid', cv=3)
#
# some of the methods of CalibratedClassifierCV()
# fit(X, y[, sample_weight])    Fit the calibrated model
# get_params([deep])    Get parameters for this estimator.
# predict(X)    Predict the target of new samples.
# predict_proba(X)    Posterior probabilities of classification
# -----
# video link:
# -----

```

```

alpha = [100,200,500,1000,2000]
max_depth = [5, 10]
cv_log_error_array = []
for i in alpha:
    for j in max_depth:
        print("for n_estimators =", i,"and max depth = ", j)
        clf = RandomForestClassifier(n_estimators=i, criterion='gini',
max_depth=j, random_state=42, n_jobs=-1)
        clf.fit(train_x_onehotCoding, train_y)
        sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
        sig_clf.fit(train_x_onehotCoding, train_y)
        sig_clf_probs = sig_clf.predict_proba(cv_x_onehotCoding)
        cv_log_error_array.append(log_loss(cv_y, sig_clf_probs, labels=
clf.classes_, eps=1e-15))
        print("Log Loss :",log_loss(cv_y, sig_clf_probs))

'''fig, ax = plt.subplots()
features = np.dot(np.array(alpha)[: ,None],np.array(max_depth)[None]).ra
vel()
ax.plot(features, cv_log_error_array,c='g')
for i, txt in enumerate(np.round(cv_log_error_array,3)):
    ax.annotate((alpha[int(i/2)],max_depth[int(i%2)],str(txt)), (featur
es[i],cv_log_error_array[i]))
plt.grid()
plt.title("Cross Validation Error for each alpha")
plt.xlabel("Alpha i's")
plt.ylabel("Error measure")
plt.show()
'''

best_alpha = np.argmin(cv_log_error_array)
clf = RandomForestClassifier(n_estimators=alpha[int(best_alpha/2)], cri
terion='gini', max_depth=max_depth[int(best_alpha%2)], random_state=42,
n_jobs=-1)
clf.fit(train_x_onehotCoding, train_y)
sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
sig_clf.fit(train_x_onehotCoding, train_y)

```

```

# summarizing data
rf_1_best_alpha = best_alpha
rf_1_encoding = "one hot"

predict_y = sig_clf.predict_proba(train_x_onehotCoding)
rf_1_train_log_loss = log_loss(y_train, predict_y, labels=clf.classes_,
                               eps=1e-15)
print('For values of best estimator = ', alpha[int(best_alpha/2)], "The
      train log loss is:", log_loss(y_train, predict_y, labels=clf.classes_,
                                   eps=1e-15))

predict_y = sig_clf.predict_proba(cv_x_onehotCoding)
rf_1_cv_log_loss = log_loss(y_cv, predict_y, labels=clf.classes_, eps=1
                             e-15)
print('For values of best estimator = ', alpha[int(best_alpha/2)], "The
      cross validation log loss is:", log_loss(y_cv, predict_y, labels=clf.cl
asses_, eps=1e-15))

predict_y = sig_clf.predict_proba(test_x_onehotCoding)
rf_1_test_log_loss = log_loss(y_test, predict_y, labels=clf.classes_, e
ps=1e-15)
print('For values of best estimator = ', alpha[int(best_alpha/2)], "The
      test log loss is:", log_loss(y_test, predict_y, labels=clf.classes_, ep
s=1e-15))

```

```

for n_estimators = 100 and max depth = 5
Log Loss : 1.2175299378574889
for n_estimators = 100 and max depth = 10
Log Loss : 1.1539699434400768
for n_estimators = 200 and max depth = 5
Log Loss : 1.200387501262619
for n_estimators = 200 and max depth = 10
Log Loss : 1.1465693185557682
for n_estimators = 500 and max depth = 5
Log Loss : 1.1899905835637603
for n_estimators = 500 and max depth = 10
Log Loss : 1.136723200064212
for n_estimators = 1000 and max depth = 5
Log Loss : 1.1862924320074288
for n_estimators = 1000 and max depth = 10

```

```

Log Loss : 1.1305238082853042
for n_estimators = 2000 and max depth = 5
Log Loss : 1.186432501985376
for n_estimators = 2000 and max depth = 10
Log Loss : 1.1266426596769563
For values of best estimator = 2000 The train log loss is: 0.651854924
0487791
For values of best estimator = 2000 The cross validation log loss is:
1.1266426596769563
For values of best estimator = 2000 The test log loss is: 1.1619795885
896385

```

4.5.2. Testing model with best hyper parameters (One Hot Encoding)

```

In [83]: # -----
# default parameters
# sklearn.ensemble.RandomForestClassifier(n_estimators=10, criterion='g
ini', max_depth=None, min_samples_split=2,
# min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='aut
o', max_leaf_nodes=None, min_impurity_decrease=0.0,
# min_impurity_split=None, bootstrap=True, oob_score=False, n_jobs=1, r
andom_state=None, verbose=0, warm_start=False,
# class_weight=None)

# Some of methods of RandomForestClassifier()
# fit(X, y, [sample_weight])    Fit the SVM model according to the give
n training data.
# predict(X)    Perform classification on samples in X.
# predict_proba (X)    Perform classification on samples in X.

# some of attributes of RandomForestClassifier()
# feature_importances_ : array of shape = [n_features]
# The feature importances (the higher, the more important the feature).

# -----
# video link: https://www.appliedaicourse.com/course/applied-ai-course-
online/lessons/random-forest-and-their-construction-2/

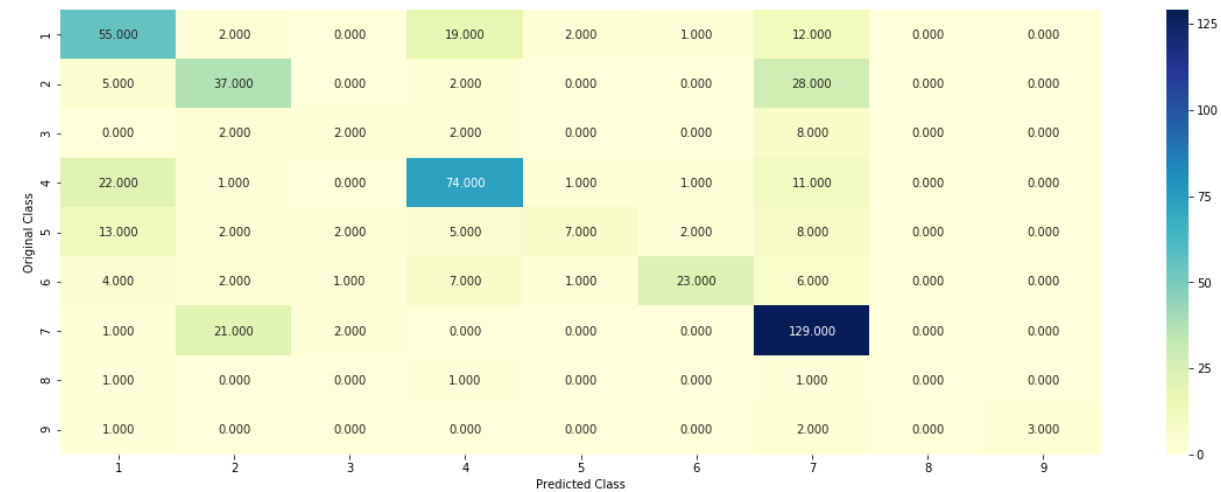
```

```
# -----
clf = RandomForestClassifier(n_estimators=alpha[int(best_alpha/2)], cri
terion='gini', max_depth=max_depth[int(best_alpha%2)], random_state=42,
n_jobs=-1)
rf_l_misclassified = predict_and_plot_confusion_matrix(train_x_onehotCo
ding, train_y,cv_x_onehotCoding,cv_y, clf)
```

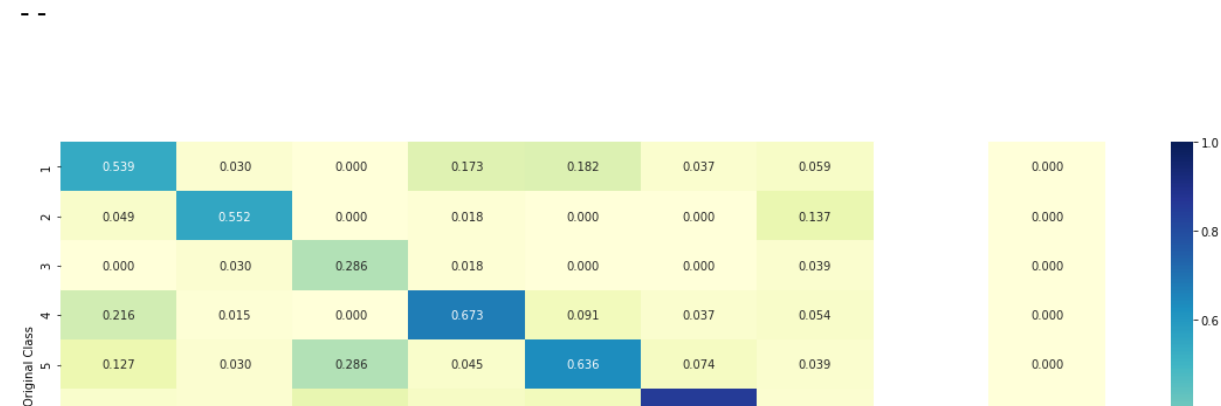
Log loss : 1.1266426596769563

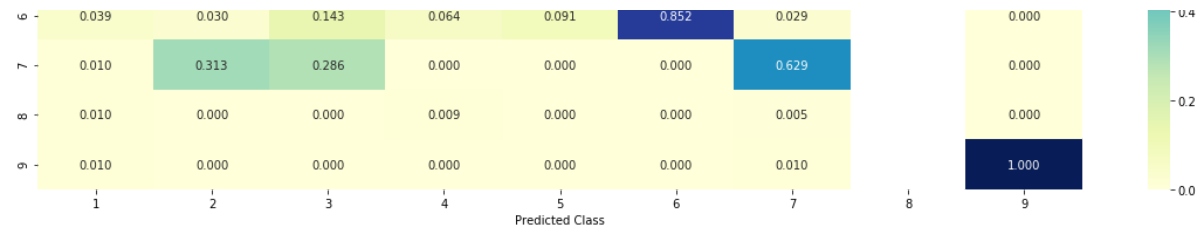
Number of mis-classified points : 0.37969924812030076

----- Confusion matrix -----

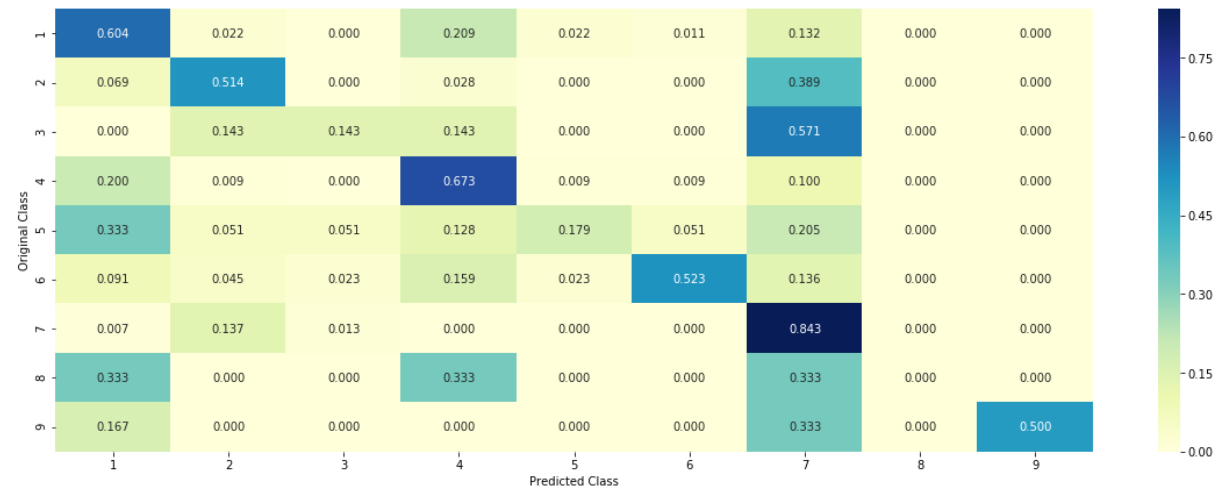


----- Precision matrix (Column Sum=1) -----





----- Recall matrix (Row sum=1) -----



4.5.3. Feature Importance

4.5.3.1. Correctly Classified point

```
In [84]: # test_point_index = 10
clf = RandomForestClassifier(n_estimators=alpha[int(best_alpha/2)], criterion='gini', max_depth=max_depth[int(best_alpha%2)], random_state=42, n_jobs=-1)
clf.fit(train_x_onehotCoding, train_y)
sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
sig_clf.fit(train_x_onehotCoding, train_y)
```



```

test_point_index = 1
no_feature = 100
predicted_cls = sig_clf.predict(test_x_onehotCoding[test_point_index])
print("Predicted Class :", predicted_cls[0])
print("Predicted Class Probabilities:", np.round(sig_clf.predict_proba(
test_x_onehotCoding[test_point_index]),4))
print("Actual Class :", test_y[test_point_index])
indices = np.argsort(-clf.feature_importances_)
print("-"*50)
get_impfeature_names(indices[:no_feature], test_df['TEXT'].iloc[test_po
int_index],test_df['Gene'].iloc[test_point_index],test_df['Variation'].
iloc[test_point_index], no_feature)

```

Predicted Class : 1

Predicted Class Probabilities: [[0.4434 0.0286 0.0184 0.0814 0.2621 0.1348 0.0229 0.003 0.0053]]

Actual Class : 1

```

-----
0 Text feature [kinase] present in test data point [True]
1 Text feature [activating] present in test data point [True]
3 Text feature [activation] present in test data point [True]
5 Text feature [suppressor] present in test data point [True]
6 Text feature [tyrosine] present in test data point [True]
7 Text feature [phosphorylation] present in test data point [True]
9 Text feature [function] present in test data point [True]
10 Text feature [nonsense] present in test data point [True]
13 Text feature [constitutive] present in test data point [True]
15 Text feature [signaling] present in test data point [True]
16 Text feature [loss] present in test data point [True]
18 Text feature [missense] present in test data point [True]
20 Text feature [activate] present in test data point [True]
21 Text feature [downstream] present in test data point [True]
25 Text feature [yeast] present in test data point [True]
35 Text feature [stability] present in test data point [True]
37 Text feature [cells] present in test data point [True]
39 Text feature [protein] present in test data point [True]
40 Text feature [patients] present in test data point [True]
42 Text feature [clinical] present in test data point [True]
43 Text feature [functional] present in test data point [True]
45 Text feature [variants] present in test data point [True]

```

```

43 Text feature [variants] present in test data point [True]
50 Text feature [pathogenic] present in test data point [True]
51 Text feature [response] present in test data point [True]
52 Text feature [ligand] present in test data point [True]
55 Text feature [phosphorylated] present in test data point [True]
57 Text feature [amplification] present in test data point [True]
59 Text feature [uncertain] present in test data point [True]
60 Text feature [cell] present in test data point [True]
64 Text feature [defective] present in test data point [True]
66 Text feature [deleterious] present in test data point [True]
67 Text feature [brca1] present in test data point [True]
70 Text feature [repair] present in test data point [True]
74 Text feature [carriers] present in test data point [True]
79 Text feature [useful] present in test data point [True]
80 Text feature [proteins] present in test data point [True]
81 Text feature [sensitivity] present in test data point [True]
85 Text feature [neutral] present in test data point [True]
86 Text feature [resistant] present in test data point [True]
87 Text feature [unstable] present in test data point [True]
89 Text feature [factor] present in test data point [True]
91 Text feature [potential] present in test data point [True]
96 Text feature [patient] present in test data point [True]
Out of the top 100 features 43 are present in query point

```

4.5.3.2. Inorrectly Classified point

```

In [85]: test_point_index = 100
no_feature = 100
predicted_cls = sig_clf.predict(test_x_onehotCoding[test_point_index])
print("Predicted Class :", predicted_cls[0])
print("Predicted Class Probabilities:", np.round(sig_clf.predict_proba(
test_x_onehotCoding[test_point_index]),4))
print("Actuall Class :", test_y[test_point_index])
indices = np.argsort(-clf.feature_importances_)
print("-"*50)
get_impfeature_names(indices[:no_feature], test_df['TEXT'].iloc[test_po
int_index],test_df['Gene'].iloc[test_point_index],test_df['Variation'].
iloc[test_point_index], no_feature)

```

```
Predicted Class : 4
Predicted Class Probabilities: [[0.1339 0.076 0.0215 0.5529 0.0511 0.0
528 0.099 0.0057 0.0071]]
Actual Class : 4
-----
0 Text feature [kinase] present in test data point [True]
1 Text feature [activating] present in test data point [True]
3 Text feature [activation] present in test data point [True]
5 Text feature [suppressor] present in test data point [True]
9 Text feature [function] present in test data point [True]
10 Text feature [nonsense] present in test data point [True]
11 Text feature [inhibitor] present in test data point [True]
13 Text feature [constitutive] present in test data point [True]
14 Text feature [growth] present in test data point [True]
15 Text feature [signaling] present in test data point [True]
16 Text feature [loss] present in test data point [True]
18 Text feature [missense] present in test data point [True]
26 Text feature [receptor] present in test data point [True]
28 Text feature [extracellular] present in test data point [True]
29 Text feature [proliferation] present in test data point [True]
31 Text feature [constitutively] present in test data point [True]
32 Text feature [transforming] present in test data point [True]
37 Text feature [cells] present in test data point [True]
39 Text feature [protein] present in test data point [True]
40 Text feature [patients] present in test data point [True]
42 Text feature [clinical] present in test data point [True]
43 Text feature [functional] present in test data point [True]
45 Text feature [variants] present in test data point [True]
51 Text feature [response] present in test data point [True]
52 Text feature [ligand] present in test data point [True]
55 Text feature [phosphorylated] present in test data point [True]
57 Text feature [amplification] present in test data point [True]
60 Text feature [cell] present in test data point [True]
65 Text feature [survival] present in test data point [True]
68 Text feature [null] present in test data point [True]
69 Text feature [inhibition] present in test data point [True]
74 Text feature [carriers] present in test data point [True]
77 Text feature [lines] present in test data point [True]
80 Text feature [proteins] present in test data point [True]
```

```
88 Text feature [frameshift] present in test data point [True]
89 Text feature [factor] present in test data point [True]
90 Text feature [pathways] present in test data point [True]
98 Text feature [active] present in test data point [True]
Out of the top 100 features 38 are present in query point
```

4.5.3. Hyper paramter tuning (With Response Coding)

```
In [86]: # -----
# default parameters
# sklearn.ensemble.RandomForestClassifier(n_estimators=10, criterion='g
ini', max_depth=None, min_samples_split=2,
# min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='aut
o', max_leaf_nodes=None, min_impurity_decrease=0.0,
# min_impurity_split=None, bootstrap=True, oob_score=False, n_jobs=1, r
andom_state=None, verbose=0, warm_start=False,
# class_weight=None)

# Some of methods of RandomForestClassifier()
# fit(X, y, [sample_weight])    Fit the SVM model according to the give
n training data.
# predict(X)    Perform classification on samples in X.
# predict_proba (X)    Perform classification on samples in X.

# some of attributes of RandomForestClassifier()
# feature_importances_ : array of shape = [n_features]
# The feature importances (the higher, the more important the feature).

# -----
# video link: https://www.appliedaicourse.com/course/applied-ai-course-
online/lessons/random-forest-and-their-construction-2/
# -----

# find more about CalibratedClassifierCV here at http://scikit-learn.or
g/stable/modules/generated/sklearn.calibration.CalibratedClassifierCV.h
tml
# -----
```

```

# default paramters
# sklearn.calibration.CalibratedClassifierCV(base_estimator=None, metho
d='sigmoid', cv=3)
#
# some of the methods of CalibratedClassifierCV()
# fit(X, y[, sample_weight])    Fit the calibrated model
# get_params([deep])           Get parameters for this estimator.
# predict(X)                    Predict the target of new samples.
# predict_proba(X)              Posterior probabilities of classification
#-----
# video link:
#-----

alpha = [10,50,100,200,500,1000]
max_depth = [2,3,5,10]
cv_log_error_array = []
for i in alpha:
    for j in max_depth:
        print("for n_estimators =", i,"and max depth = ", j)
        clf = RandomForestClassifier(n_estimators=i, criterion='gini',
max_depth=j, random_state=42, n_jobs=-1)
        clf.fit(train_x_responseCoding, train_y)
        sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
        sig_clf.fit(train_x_responseCoding, train_y)
        sig_clf_probs = sig_clf.predict_proba(cv_x_responseCoding)
        cv_log_error_array.append(log_loss(cv_y, sig_clf_probs, labels=
clf.classes_, eps=1e-15))
        print("Log Loss :",log_loss(cv_y, sig_clf_probs))
    ...
fig, ax = plt.subplots()
features = np.dot(np.array(alpha)[: ,None],np.array(max_depth)[None]).ra
vel()
ax.plot(features, cv_log_error_array,c='g')
for i, txt in enumerate(np.round(cv_log_error_array,3)):
    ax.annotate((alpha[int(i/4)],max_depth[int(i%4)],str(txt)), (featur
es[i],cv_log_error_array[i]))
plt.grid()
plt.title("Cross Validation Error for each alpha")
plt.xlabel("Alpha i's")

```

```

plt.ylabel("Error measure")
plt.show()
'''

best_alpha = np.argmin(cv_log_error_array)
clf = RandomForestClassifier(n_estimators=alpha[int(best_alpha/4)], cri
terion='gini', max_depth=max_depth[int(best_alpha%4)], random_state=42,
n_jobs=-1)
clf.fit(train_x_responseCoding, train_y)
sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
sig_clf.fit(train_x_responseCoding, train_y)

# summarizing data
rf_best_alpha = None
rf_encoding = "Response"

predict_y = sig_clf.predict_proba(train_x_responseCoding)
rf_train_log_loss = log_loss(y_train, predict_y, labels=clf.classes_, e
ps=1e-15)
print('For values of best alpha = ', alpha[int(best_alpha/4)], "The tra
in log loss is:", log_loss(y_train, predict_y, labels=clf.classes_, eps=
1e-15))

predict_y = sig_clf.predict_proba(cv_x_responseCoding)
rf_cv_log_loss = log_loss(y_cv, predict_y, labels=clf.classes_, eps=1e-
15)
print('For values of best alpha = ', alpha[int(best_alpha/4)], "The cro
ss validation log loss is:", log_loss(y_cv, predict_y, labels=clf.classe
s_, eps=1e-15))

predict_y = sig_clf.predict_proba(test_x_responseCoding)
rf_test_log_loss = log_loss(y_test, predict_y, labels=clf.classes_, eps
=1e-15)
print('For values of best alpha = ', alpha[int(best_alpha/4)], "The tes
t log loss is:", log_loss(y_test, predict_y, labels=clf.classes_, eps=1e
-15))

```

for n_estimators = 10 and max depth = 2

Log Loss = 2.0055727704200074

```

Log Loss : 2.0955731704289974
for n_estimators = 10 and max depth = 3
Log Loss : 1.6169047701570358
for n_estimators = 10 and max depth = 5
Log Loss : 1.334644866223306
for n_estimators = 10 and max depth = 10
Log Loss : 1.8684542512117832
for n_estimators = 50 and max depth = 2
Log Loss : 1.5981096118335103
for n_estimators = 50 and max depth = 3
Log Loss : 1.302113589115677
for n_estimators = 50 and max depth = 5
Log Loss : 1.2204741533560595
for n_estimators = 50 and max depth = 10
Log Loss : 1.5206574735807847
for n_estimators = 100 and max depth = 2
Log Loss : 1.428841782153051
for n_estimators = 100 and max depth = 3
Log Loss : 1.3591674313596889
for n_estimators = 100 and max depth = 5
Log Loss : 1.2124447767246602
for n_estimators = 100 and max depth = 10
Log Loss : 1.4802820509982069
for n_estimators = 200 and max depth = 2
Log Loss : 1.5072538991210236
for n_estimators = 200 and max depth = 3
Log Loss : 1.363742833710391
for n_estimators = 200 and max depth = 5
Log Loss : 1.2432195503143726
for n_estimators = 200 and max depth = 10
Log Loss : 1.5100889686676109
for n_estimators = 500 and max depth = 2
Log Loss : 1.5473048681421222
for n_estimators = 500 and max depth = 3
Log Loss : 1.449004490635136
for n_estimators = 500 and max depth = 5
Log Loss : 1.236387935824159
for n_estimators = 500 and max depth = 10
Log Loss : 1.5199206558773126

```

```

for n_estimators = 1000 and max depth = 2

```

```

for n_estimators = 1000 and max depth = 2
Log Loss : 1.4992695129074927
for n_estimators = 1000 and max depth = 3
Log Loss : 1.4400487851396384
for n_estimators = 1000 and max depth = 5
Log Loss : 1.232973888662283
for n_estimators = 1000 and max depth = 10
Log Loss : 1.4713474028418514
For values of best alpha = 100 The train log loss is: 0.05469713948553
081
For values of best alpha = 100 The cross validation log loss is: 1.212
4447767246456
For values of best alpha = 100 The test log loss is: 1.346605713759460
5

```

4.5.4. Testing model with best hyper parameters (Response Coding)

```

In [87]: # -----
# default parameters
# sklearn.ensemble.RandomForestClassifier(n_estimators=10, criterion='g
ini', max_depth=None, min_samples_split=2,
# min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='aut
o', max_leaf_nodes=None, min_impurity_decrease=0.0,
# min_impurity_split=None, bootstrap=True, oob_score=False, n_jobs=1, r
andom_state=None, verbose=0, warm_start=False,
# class_weight=None)

# Some of methods of RandomForestClassifier()
# fit(X, y, [sample_weight])    Fit the SVM model according to the give
n training data.
# predict(X)    Perform classification on samples in X.
# predict_proba (X)    Perform classification on samples in X.

# some of attributes of RandomForestClassifier()
# feature_importances_ : array of shape = [n_features]
# The feature importances (the higher, the more important the feature).

# -----

```


video link: <https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/random-forest-and-their-construction-2/>
 # -----

```
clf = RandomForestClassifier(max_depth=max_depth[int(best_alpha%4)], n_estimators=alpha[int(best_alpha/4)], criterion='gini', max_features='auto', random_state=42)
rf_misclassified = predict_and_plot_confusion_matrix(train_x_responseCoding, train_y,cv_x_responseCoding,cv_y, clf)
```

Log loss : 1.2124448164820254

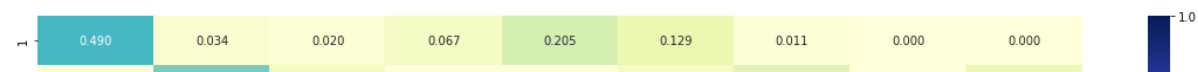
Number of mis-classified points : 0.4755639097744361

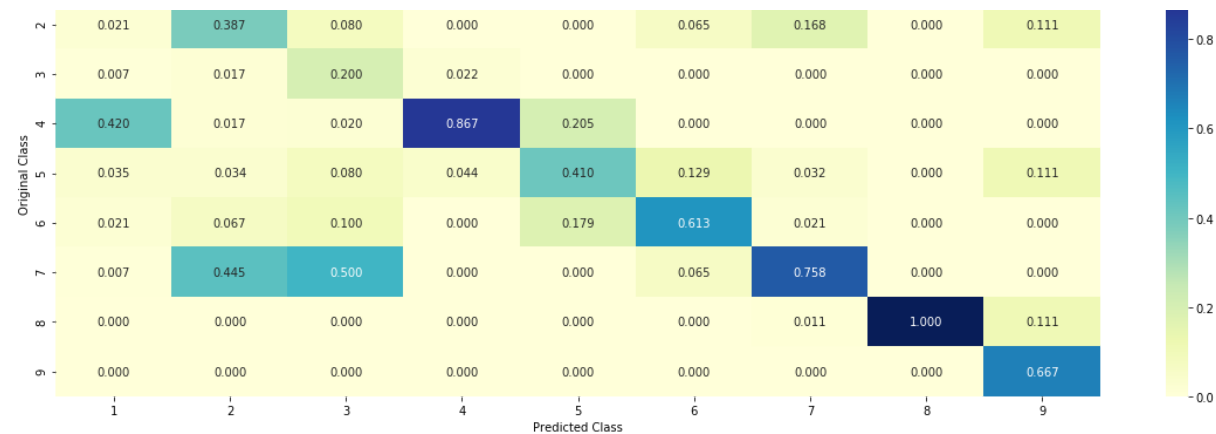
----- Confusion matrix -----



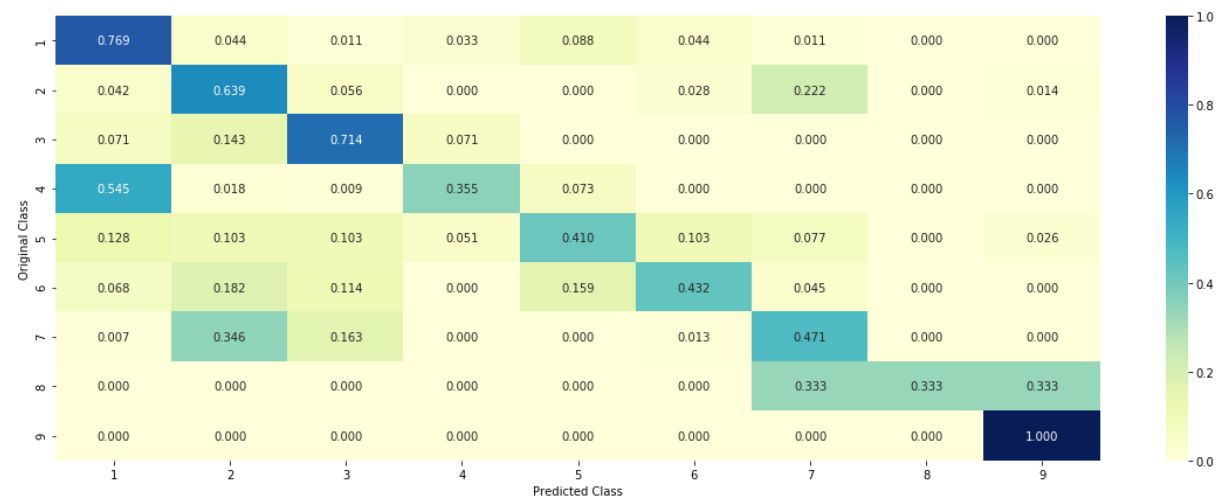
----- Precision matrix (Column Sum=1) -----

--





----- Recall matrix (Row sum=1) -----



4.5.5. Feature Importance

4.5.5.1. Correctly Classified point

```
In [88]: clf = RandomForestClassifier(n_estimators=alpha[int(best_alpha/4)], criterion='gini', max_depth=max_depth[int(best_alpha%4)], random_state=42,
```

```

n_jobs=-1)
clf.fit(train_x_responseCoding, train_y)
sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
sig_clf.fit(train_x_responseCoding, train_y)

test_point_index = 1
no_feature = 27
predicted_cls = sig_clf.predict(test_x_responseCoding[test_point_index]
.reshape(1,-1))
print("Predicted Class :", predicted_cls[0])
print("Predicted Class Probabilities:", np.round(sig_clf.predict_proba(
test_x_responseCoding[test_point_index].reshape(1,-1)),4))
print("Actual Class :", test_y[test_point_index])
indices = np.argsort(-clf.feature_importances_)
print("-"*50)
for i in indices:
    if i<9:
        print("Gene is important feature")
    elif i<18:
        print("Variation is important feature")
    else:
        print("Text is important feature")

```

```

Predicted Class : 1
Predicted Class Probabilities: [[0.2961 0.0091 0.0817 0.0963 0.2833 0.2
05 0.008 0.0091 0.0114]]
Actual Class : 1

```

```

-----
Variation is important feature
Variation is important feature
Variation is important feature
Variation is important feature
Gene is important feature
Variation is important feature
Variation is important feature
Text is important feature
Text is important feature
Text is important feature

```

```
Text is important feature
Gene is important feature
Text is important feature
Variation is important feature
Gene is important feature
Gene is important feature
Text is important feature
Gene is important feature
Gene is important feature
Variation is important feature
Variation is important feature
Text is important feature
Text is important feature
Gene is important feature
Text is important feature
Gene is important feature
Gene is important feature
```

4.5.5.2. Incorrectly Classified point

```
In [89]: test_point_index = 100
predicted_cls = sig_clf.predict(test_x_responseCoding[test_point_index]
.reshape(1,-1))
print("Predicted Class :", predicted_cls[0])
print("Predicted Class Probabilities:", np.round(sig_clf.predict_proba(
test_x_responseCoding[test_point_index].reshape(1,-1)),4))
print("Actual Class :", test_y[test_point_index])
indices = np.argsort(-clf.feature_importances_)
print("-"*50)
for i in indices:
    if i<9:
        print("Gene is important feature")
    elif i<18:
        print("Variation is important feature")
    else:
        print("Text is important feature")
```

Predicted Class : 1

```
Predicted Class Probabilities: [[0.4824 0.0119 0.0705 0.3378 0.0169 0.0289 0.0088 0.0207 0.0221]]
```

```
Actual Class : 4
```

```
-----  
Variation is important feature  
Variation is important feature  
Variation is important feature  
Variation is important feature  
Gene is important feature  
Variation is important feature  
Variation is important feature  
Text is important feature  
Text is important feature  
Text is important feature  
Text is important feature  
Gene is important feature  
Text is important feature  
Variation is important feature  
Gene is important feature  
Gene is important feature  
Text is important feature  
Gene is important feature  
Gene is important feature  
Variation is important feature  
Variation is important feature  
Text is important feature  
Text is important feature  
Gene is important feature  
Text is important feature  
Gene is important feature  
Gene is important feature
```

4.7 Stack the models

4.7.1 testing with hyper parameter tuning

```
In [90]: from mlxtend.classifier import StackingClassifier
```

```
In [91]: # read more about SGDClassifier() at http://scikit-learn.org/stable/modules/generated/sklearn.linear\_model.SGDClassifier.html
# -----
# default parameters
# SGDClassifier(loss='hinge', penalty='l2', alpha=0.0001, l1_ratio=0.15, fit_intercept=True, max_iter=None, tol=None,
# shuffle=True, verbose=0, epsilon=0.1, n_jobs=1, random_state=None, learning_rate='optimal', eta0=0.0, power_t=0.5,
# class_weight=None, warm_start=False, average=False, n_iter=None)

# some of methods
# fit(X, y[, coef_init, intercept_init, ...])    Fit linear model with Stochastic Gradient Descent.
# predict(X)    Predict class labels for samples in X.

#-----
# video link: https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/geometric-intuition-1/
#-----

# read more about support vector machines with linear kernels here http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
# -----
# default parameters
# SVC(C=1.0, kernel='rbf', degree=3, gamma='auto', coef0=0.0, shrinking=True, probability=False, tol=0.001,
# cache_size=200, class_weight=None, verbose=False, max_iter=-1, decision_function_shape='ovr', random_state=None)

# Some of methods of SVM()
# fit(X, y, [sample_weight])    Fit the SVM model according to the given training data.
# predict(X)    Perform classification on samples in X.
# -----
# video link: https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/mathematical-derivation-copy-8/
```

```

# -----

# read more about support vector machines with linear kernalns here http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
# -----

# default parameters
# sklearn.ensemble.RandomForestClassifier(n_estimators=10, criterion='gini', max_depth=None, min_samples_split=2,
# min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0,
# min_impurity_split=None, bootstrap=True, oob_score=False, n_jobs=1, random_state=None, verbose=0, warm_start=False,
# class_weight=None)

# Some of methods of RandomForestClassifier()
# fit(X, y, [sample_weight])    Fit the SVM model according to the given training data.
# predict(X)    Perform classification on samples in X.
# predict_proba(X)    Perform classification on samples in X.

# some of attributes of RandomForestClassifier()
# feature_importances_ : array of shape = [n_features]
# The feature importances (the higher, the more important the feature).

# -----
# video link: https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/random-forest-and-their-construction-2/
# -----

clf1 = SGDClassifier(alpha=0.001, penalty='l2', loss='log', class_weight='balanced', random_state=0)
clf1.fit(train_x_onehotCoding, train_y)
sig_clf1 = CalibratedClassifierCV(clf1, method="sigmoid")

clf2 = SGDClassifier(alpha=1, penalty='l2', loss='hinge', class_weight='balanced', random_state=0)

```

```

clf2.fit(train_x_onehotCoding, train_y)
sig_clf2 = CalibratedClassifierCV(clf2, method="sigmoid")

clf3 = MultinomialNB(alpha=0.001)
clf3.fit(train_x_onehotCoding, train_y)
sig_clf3 = CalibratedClassifierCV(clf3, method="sigmoid")

sig_clf1.fit(train_x_onehotCoding, train_y)
LR = (log_loss(cv_y, sig_clf1.predict_proba(cv_x_onehotCoding)))
print("Logistic Regression : Log Loss: %0.2f" % (log_loss(cv_y, sig_clf1.predict_proba(cv_x_onehotCoding))))

sig_clf2.fit(train_x_onehotCoding, train_y)
SVM = (log_loss(cv_y, sig_clf2.predict_proba(cv_x_onehotCoding)))
print("Support vector machines : Log Loss: %0.2f" % (log_loss(cv_y, sig_clf2.predict_proba(cv_x_onehotCoding))))

sig_clf3.fit(train_x_onehotCoding, train_y)
NB = (log_loss(cv_y, sig_clf3.predict_proba(cv_x_onehotCoding)))
print("Naïve Bayes : Log Loss: %0.2f" % (log_loss(cv_y, sig_clf3.predict_proba(cv_x_onehotCoding))))

print("-"*50)
alpha = [0.0001,0.001,0.01,0.1,1,10]
best_alpha_loss = 999
for i in alpha:
    lr = LogisticRegression(C=i)
    sclf = StackingClassifier(classifiers=[sig_clf1, sig_clf2, sig_clf3], meta_classifier=lr, use_probas=True)
    sclf.fit(train_x_onehotCoding, train_y)
    print("Stacking Classifier : for the value of alpha: %f Log Loss: %0.3f" % (i, log_loss(cv_y, sclf.predict_proba(cv_x_onehotCoding))))
    log_error = log_loss(cv_y, sclf.predict_proba(cv_x_onehotCoding))
    if best_alpha_loss > log_error:
        best_alpha_loss = log_error
        best_alpha = i

```

Logistic Regression : Log Loss: 0.99

Support vector machines : Log Loss: 1.53

Support Vector machines : Log Loss: 1.93

Naive Bayes : Log Loss: 1.22

Stacking Classifier : for the value of alpha: 0.000100 Log Loss: 2.178
Stacking Classifier : for the value of alpha: 0.001000 Log Loss: 2.031
Stacking Classifier : for the value of alpha: 0.010000 Log Loss: 1.478
Stacking Classifier : for the value of alpha: 0.100000 Log Loss: 1.063
Stacking Classifier : for the value of alpha: 1.000000 Log Loss: 1.117
Stacking Classifier : for the value of alpha: 10.000000 Log Loss: 1.347

```
In [92]: stack_best_alpha = best_alpha  
stack_encoding = "one hot"  
print(best_alpha_loss)
```

1.0630162119351134

4.7.2 testing the model with the best hyper parameters

```
In [93]: lr = LogisticRegression(C=0.1)  
sclf = StackingClassifier(classifiers=[sig_clf1, sig_clf2, sig_clf3], m  
eta_classifier=lr, use_probab=True)  
sclf.fit(train_x_onehotCoding, train_y)  
  
log_error = log_loss(train_y, sclf.predict_proba(train_x_onehotCoding))  
stack_train_log_loss = log_error  
print("Log loss (train) on the stacking classifier :",log_error)  
  
log_error = log_loss(cv_y, sclf.predict_proba(cv_x_onehotCoding))  
stack_cv_log_loss = log_error  
print("Log loss (CV) on the stacking classifier :",log_error)  
  
log_error = log_loss(test_y, sclf.predict_proba(test_x_onehotCoding))  
stack_test_log_loss = log_error  
print("Log loss (test) on the stacking classifier :",log_error)  
  
stack_misclassified = np.count_nonzero((sclf.predict(test_x_onehotCoding)  
- test_y))/test_y.shape[0]
```

```
print("Number of misclassified point :", stack_misclassified)
plot_confusion_matrix(test_y=test_y, predict_y=sclf.predict(test_x_oneh
otCoding))
```

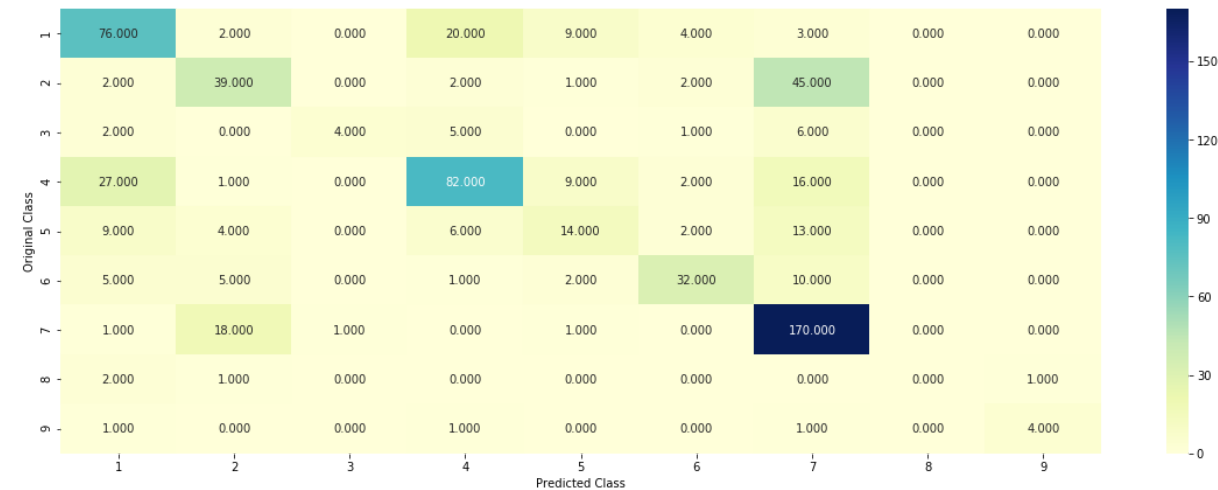
Log loss (train) on the stacking classifier : 0.6472136719980816

Log loss (CV) on the stacking classifier : 1.0630162119351134

Log loss (test) on the stacking classifier : 1.1789065541131452

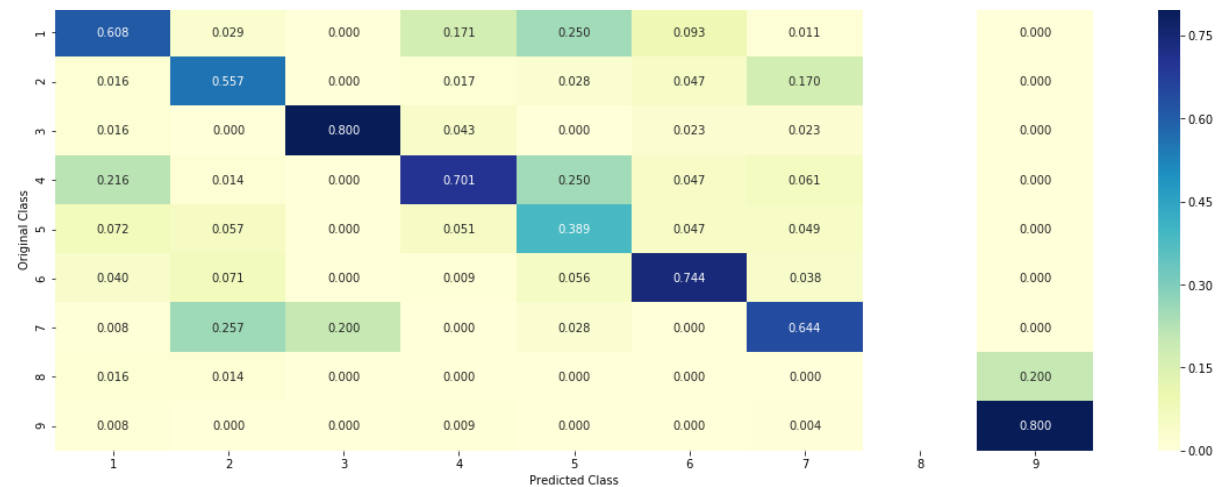
Number of misclassified point : 0.3669172932330827

----- Confusion matrix -----

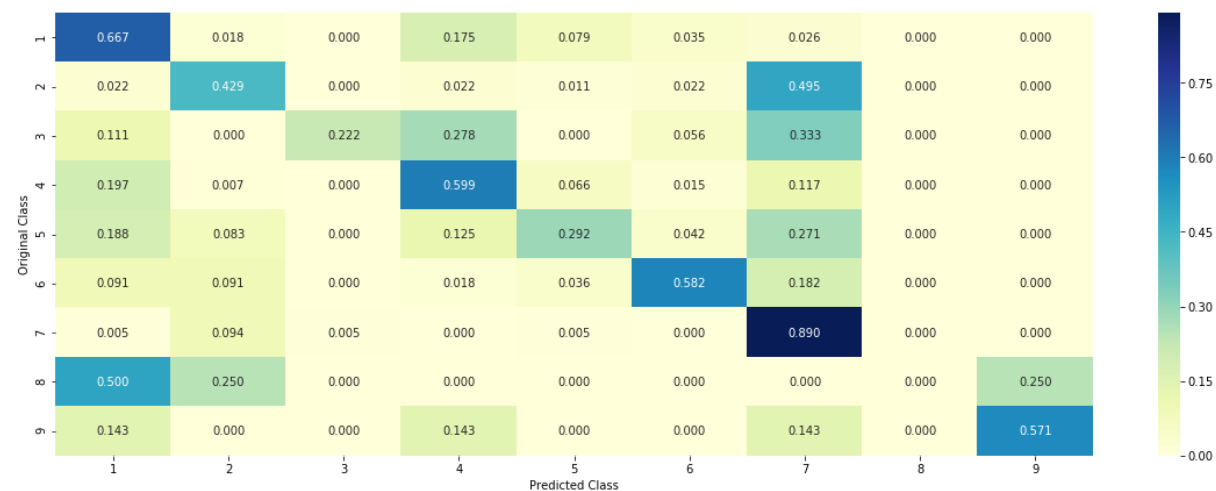


----- Precision matrix (Column Sum=1) -----

--



----- Recall matrix (Row sum=1) -----



4.7.3 Maximum Voting classifier

In [94]: `#Refer: http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingClassifier.html
from sklearn.ensemble import VotingClassifier`

```

vclf = VotingClassifier(estimators=[('lr', sig_clf1), ('svc', sig_clf2), ('rf', sig_clf3)], voting='soft')
vclf.fit(train_x_onehotCoding, train_y)

voting_best_alpha = None
voting_encoding = "One Hot"
voting_train_log_loss = log_loss(train_y, vclf.predict_proba(train_x_onehotCoding))
print("Log loss (train) on the VotingClassifier :", log_loss(train_y, vclf.predict_proba(train_x_onehotCoding)))

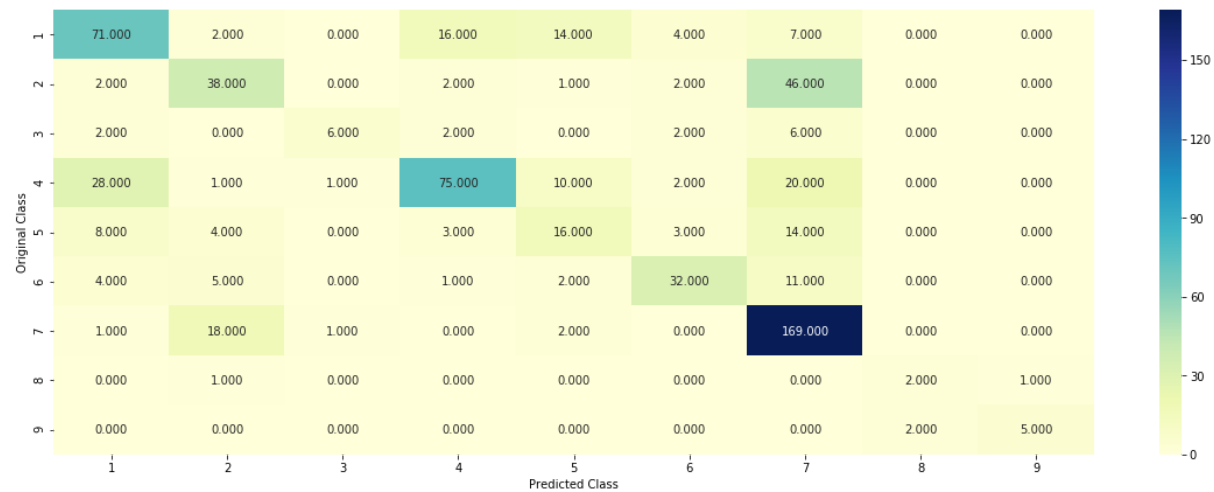
voting_cv_log_loss = log_loss(cv_y, vclf.predict_proba(cv_x_onehotCoding))
print("Log loss (CV) on the VotingClassifier :", log_loss(cv_y, vclf.predict_proba(cv_x_onehotCoding)))

voting_test_log_loss = log_loss(test_y, vclf.predict_proba(test_x_onehotCoding))
print("Log loss (test) on the VotingClassifier :", log_loss(test_y, vclf.predict_proba(test_x_onehotCoding)))

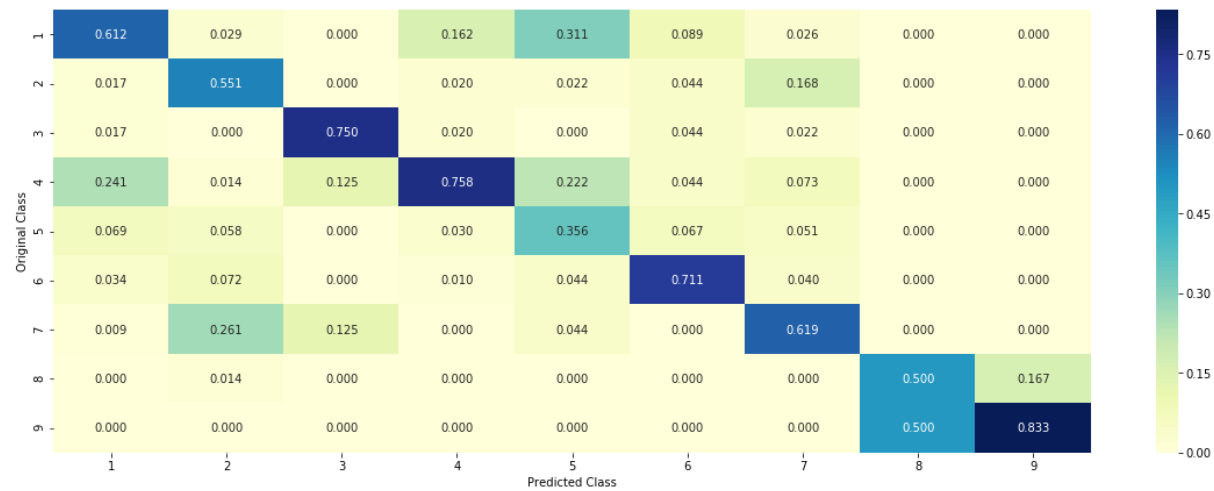
voting_misclassified = np.count_nonzero((vclf.predict(test_x_onehotCoding) - test_y)) / test_y.shape[0]
print("Number of misclassified point :", voting_misclassified)
plot_confusion_matrix(test_y=test_y, predict_y=vclf.predict(test_x_onehotCoding))

Log loss (train) on the VotingClassifier : 0.8607038792484375
Log loss (CV) on the VotingClassifier : 1.1006667191259671
Log loss (test) on the VotingClassifier : 1.1681324905006454
Number of misclassified point : 0.3774436090225564
----- Confusion matrix -----

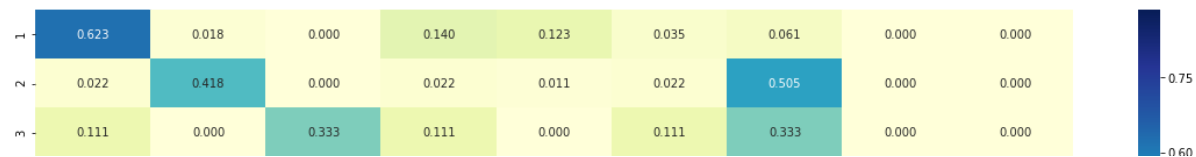
```



----- Precision matrix (Column Sum=1) -----



----- Recall matrix (Row sum=1) -----





5. Summary

```
In [95]: from prettytable import PrettyTable

# to reference from http://zetcode.com/python/prettytable/
summary = PrettyTable()
```

Feature Summary

```
In [96]: summary.field_names = ["Feature", "Best alpha", "Train log loss", "CV log
loss", "Test log loss", "Distribution"]
```

```
In [97]: summary.add_row(["Gene", gene_best_alpha, gene_train_log_loss, gene_cv_log
_loss, gene_test_log_loss, gene_stability])
summary.add_row(["Variation", variation_best_alpha, variation_train_log_l
oss, variation_cv_log_loss, variation_test_log_loss, variation_stability])
summary.add_row(["Text", text_best_alpha, text_train_log_loss, text_cv_log
_loss, text_test_log_loss, text_stability])

print("Feature Values and their stability..\n")
print(summary)
```

Feature Values and their stability..

Feature	Best alpha	Train log loss	CV log loss	Test log loss	Distribution
Gene	1	1.041718670770223	1.1722710197306832	1.257956767116093	96.9172932330827
Variation	2	1.0965469452519268	1.703931262692979	1.6981929725933396	10.601503759398497
Text	2	0.6808062934422892	1.0515673752791947	1.1663402223432042	99.613

Observation

1. Gene feature looks like most useful one
2. Test log loss in **Text** feature is 1.108 which is best and have naerly same distribution as that of Train data
3. Distribution of variation varies among train/test and cv

Model Summary

```
In [98]: summary = PrettyTable()
summary.field_names = ["Model", "Encoding", "Best Alpha", "Train logloss", "CV logloss", "Test logloss", "MisClassified"]
```

```
In [99]: summary.add_row(["Naive Bayes", nb_encoding, nb_best_alpha, round(nb_train_log_loss, 3), round(nb_cv_log_loss, 3), round(nb_test_log_loss, 3), nb_misclassified])

summary.add_row(["Logistic R.", lr_encoding, lr_best_alpha, round(lr_train_log_loss, 3), round(lr_cv_log_loss, 3), round(lr_test_log_loss, 3), lr_misclassified])
```

```
summary.add_row(["Logistic Balanced",lr_bal_encoding,lr_bal_best_alpha,round(lr_bal_train_log_loss,3),round(lr_bal_cv_log_loss,3),round(lr_bal_test_log_loss,3),lr_bal_misclassified])
summary.add_row(["Linear SVM",svm_encoding,svm_best_alpha,round(svm_train_log_loss,3),round(svm_cv_log_loss,3),round(svm_test_log_loss,3),svm_misclassified])

summary.add_row(["KNN classifier",knn_encoding,knn_best_alpha,round(knn_train_log_loss,3),round(knn_cv_log_loss,3),round(knn_test_log_loss,3),knn_misclassified])

summary.add_row(["Random Forest",rf_1_encoding,rf_1_best_alpha,round(rf_1_train_log_loss,3),round(rf_1_cv_log_loss,3),round(rf_1_test_log_loss,3),rf_1_misclassified])
summary.add_row(["Random Forest",rf_encoding,rf_best_alpha,round(rf_train_log_loss,3),round(rf_cv_log_loss,3),round(rf_test_log_loss,3),rf_misclassified])
summary.add_row(["Stacking ",stack_encoding,stack_best_alpha,round(stack_train_log_loss,3),round(stack_cv_log_loss,3),round(stack_test_log_loss,3),stack_misclassified])
summary.add_row(["Max. Voting",voting_encoding,voting_best_alpha,round(voting_train_log_loss,3),round(voting_cv_log_loss,3),round(voting_test_log_loss,3),voting_misclassified])
```

```
In [100]: print("Model and their performance...\n")
          print(summary)
```

Model and their performance...

```
+-----+-----+-----+-----+-----+
--+-----+-----+
|      Model      | Encoding | Best Alpha | Train logloss | CV logloss |
s | Test logloss |   MisClassified   |
+-----+-----+-----+-----+-----+
--+-----+-----+
|   Naive Bayes   | One hot  |    1000    |    0.994    |    1.186    |
|   1.219         | 0.3966165413533835 |
|   Logistic R.   | one hot  |    0.001    |    0.575    |    1.01     |
```


	1.109	0.34022556390977443			
Logistic Balanced	One hot	0.001		0.58	0.977
	1.099	0.33646616541353386			
Linear SVM	one hot	0.01		0.724	1.069
	1.162	0.32142857142857145			
KNN classifier	Response	5		0.491	0.969
	1.057	0.31203007518796994			
Random Forest	one hot	9		0.652	1.127
	1.162	0.37969924812030076			
Random Forest	Response	None		0.055	1.212
	1.347	0.4755639097744361			
Stacking	one hot	0.1		0.647	1.063
	1.179	0.3669172932330827			
Max. Voting	One Hot	None		0.861	1.101
	1.168	0.3774436090225564			
+-----+-----+-----+-----+-----+					
-+-----+-----+-----+-----+-----+					

Obseravation:

1. Model found **overfitted** with **Response encoding** as can see in case of KNN and Random Forest
2. Best classification is found in case of Max. Voting but suffers due to interpretability
3. **KNN classifier** gave minimum log loss for both CV and test data
4. **Random Forest** with response coding was slowest to train and gave highest mis-classification