```
import numpy as np
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import SimpleRNN, Dense
```

```
text = "This is GeeksforGeeks a software training institute"
chars = sorted(list(set(text)))
char_to_index = {char: i for i, char in enumerate(chars)}
index_to_char = {i: char for i, char in enumerate(chars)}
```

```
####
char_to_index
####
```

```
{' ': 0,
 'G': 1,
 'T': 2,
 'a': 3,
 'e': 4,
 'f': 5,
 'g': 6,
 'h': 7,
 'i': 8,
 'k': 9,
 'n': 10,
 'o': 11,
 'r': 12,
 's': 13,
 't': 14,
 'u': 15,
 'w': 16}
```

```
 ####
set(text)
####
```

```
{' ',
 'G',
 'T',
 'a',
 'e',
 'f',
 'g',
 'h',
 'i',
 'k',
 'n',
 'o',
 'r',
 's',
 't',
 'u',
 'w'}
```

```
####
({char: i for i, char in enumerate(chars)}), ({i: char for i, char in enumerate(chars)})
####
```

```
({' ': 0,
  'G': 1,
  'T': 2,
  'a': 3,
  'e': 4,
  'f': 5,
  'g': 6,
  'h': 7,
  'i': 8,
  'k': 9,
  'n': 10,
  'o': 11,
  'r': 12,
  's': 13,
  't': 14,
  'u': 15,
  'w': 16},
 {0: ' ',
  1: 'G',
  2: 'T',
  3: 'a',
  4: 'e',
  5: 'f',
  6: 'g',
  7: 'h',
```

```
 8: 'i',
 9: 'k',
 10: 'n',
 11: 'o',
 12: 'r',
 13: 's',
 14: 't',
 15: 'u',
 16: 'w'})
```

```
####
print(set(text))
print(list(set(text)))
print(sorted(list(set(text))))
####
```

```
{'o', 'i', 'a', 'T', 'f', 'g', 'k', 'G', 't', 'h', 'w', 'n', 'r', 'u', 'e', ' ', 's'}
['o', 'i', 'a', 'T', 'f', 'g', 'k', 'G', 't', 'h', 'w', 'n', 'r', 'u', 'e', ' ', 's']
[' ', 'G', 'T', 'a', 'e', 'f', 'g', 'h', 'i', 'k', 'n', 'o', 'r', 's', 't', 'u', 'w']
```

```
####
len(text) - 3
####
```

```
48
```

```
seq_length = 3
sequences  = []
labels     = []

for i in range(len(text) - seq_length):
  seq = text[i:i + seq_length]
  label = text[i + seq_length]
  sequences.append([char_to_index[char] for char in seq])
  labels.append(char_to_index[label])

  X = np.array(sequences)
  y = np.array(labels)
  print(X, y)
```

```
[[2 7 8]] [13]
[[ 2  7  8]
 [ 7  8 13]] [13  0]
[[ 2  7  8]
 [ 7  8 13]
 [ 8 13  0]] [13  0  8]
[[ 2  7  8]
 [ 7  8 13]
 [ 8 13  0]
 [13  0  8]] [13  0  8 13]
[[ 2  7  8]
 [ 7  8 13]
 [ 8 13  0]
 [13  0  8]
 [ 0  8 13]] [13  0  8 13  0]
[[ 2  7  8]
 [ 7  8 13]
 [ 8 13  0]
 [13  0  8]
 [ 0  8 13]
 [ 8 13  0]] [13  0  8 13  0  1]
[[ 2  7  8]
 [ 7  8 13]
 [ 8 13  0]
 [13  0  8]
 [ 0  8 13]
 [ 8 13  0]
 [13  0  1]] [13  0  8 13  0  1  4]
[[ 2  7  8]
 [ 7  8 13]
 [ 8 13  0]
 [13  0  8]
 [ 0  8 13]
 [ 8 13  0]
 [13  0  1]
 [ 0  1  4]] [13  0  8 13  0  1  4  4]
[[ 2  7  8]
 [ 7  8 13]
 [ 8 13  0]
 [13  0  8]
 [ 0  8 13]
 [ 8 13  0]
 [13  0  1]
 [ 0  1  4]
 [ 1  4  4]] [13  0  8 13  0  1  4  4  9]
[[ 2  7  8]
```

```
 [ 7  8 13]
 [ 8 13  0]
 [13  0  8]
 [ 0  8 13]
 [ 8 13  0]
 [13  0  1]
 [ 0  1  4]
 [ 1  4  4]
 [ 4  4  9]] [13  0  8 13  0  1  4  4  9 13]
[[ 2  7  8]
 [ 7  8 13]
 [ 8 13  0]
```

```
####
seq_length = 7
sequences  = []
labels     = []

for i in range(len(text) - seq_length):
  seq = text[i:i + seq_length]
  label = text[i + seq_length]
  sequences.append([char_to_index[char] for char in seq])
  labels.append(char_to_index[label])

  X = np.array(sequences)
  y = np.array(labels)
  print(X, y)

####
```

```
 [ 4  0 14 12  3  8 10]
 [ 0 14 12  3  8 10  8]
 [14 12  3  8 10  8 10]
 [12  3  8 10  8 10  6]
 [ 3  8 10  8 10  6  0]
 [ 8 10  8 10  6  0  8]
 [10  8 10  6  0  8 10]
 [ 8 10  6  0  8 10 13]
 [10  6  0  8 10 13 14]
 [ 6  0  8 10 13 14  8]
 [ 0  8 10 13 14  8 14]
 [ 8 10 13 14  8 14 15]] [ 0  1  4  4  9 13  5 11 12  1  4  4  9 13  0  3  0 13 11  5 14 16  3 12
  4  0 14 12  3  8 10  8 10  6  0  8 10 13 14  8 14 15 14]
[[ 2  7  8 13  0  8 13]
 [ 7  8 13  0  8 13  0]
 [ 8 13  0  8 13  0  1]
 [13  0  8 13  0  1  4]
 [ 0  8 13  0  1  4  4]
 [ 8 13  0  1  4  4  9]
 [13  0  1  4  4  9 13]
 [ 0  1  4  4  9 13  5]
 [ 1  4  4  9 13  5 11]
 [ 4  4  9 13  5 11 12]
 [ 4  9 13  5 11 12  1]
 [ 9 13  5 11 12  1  4]
 [13  5 11 12  1  4  4]
 [ 5 11 12  1  4  4  9]
 [11 12  1  4  4  9 13]
 [12  1  4  4  9 13  0]
 [ 1  4  4  9 13  0  3]
 [ 4  4  9 13  0  3  0]
 [ 4  9 13  0  3  0 13]
 [ 9 13  0  3  0 13 11]
 [13  0  3  0 13 11  5]
 [ 0  3  0 13 11  5 14]
 [ 3  0 13 11  5 14 16]
 [ 0 13 11  5 14 16  3]
 [13 11  5 14 16  3 12]
 [11  5 14 16  3 12  4]
 [ 5 14 16  3 12  4  0]
 [14 16  3 12  4  0 14]
 [16  3 12  4  0 14 12]
 [ 3 12  4  0 14 12  3]
 [12  4  0 14 12  3  8]
 [ 4  0 14 12  3  8 10]
 [ 0 14 12  3  8 10  8]
 [14 12  3  8 10  8 10]
 [12  3  8 10  8 10  6]
 [ 3  8 10  8 10  6  0]
 [ 8 10  8 10  6  0  8]
 [10  8 10  6  0  8 10]
 [ 8 10  6  0  8 10 13]
 [10  6  0  8 10 13 14]
 [ 6  0  8 10 13 14  8]
 [ 0  8 10 13 14  8 14]
 [ 8 10 13 14  8 14 15]
 [10 13 14  8 14 15 14]] [ 0  1  4  4  9 13  5 11 12  1  4  4  9 13  0  3  0 13 11  5 14 16  3 12
  4  0 14 12  3  8 10  8 10  6  0  8 10 13 14  8 14 15 14  4]
```

```
####
text[0:3], text[3]
####
```

```
('Thi', 's')
```

```
X_one_hot = tf.one_hot(X, len(chars))
y_one_hot = tf.one_hot(y, len(chars))
```

```
####
X_one_hot = tf.one_hot(X, len(chars))
y_one_hot = tf.one_hot(y, len(chars))
####
```

```
####
X_one_hot
####
```

```
<tf.Tensor: shape=(48, 3, 17), dtype=float32, numpy=
array([[[0., 0., 1., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.]],

       [[0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.]],

       [[0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        [1., 0., 0., ..., 0., 0., 0.]],

       ...,

       [[0., 0., 0., ..., 1., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 1., 0., 0.]],

       [[0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 1., 0., 0.],
        [0., 0., 0., ..., 0., 1., 0.]],

       [[0., 0., 0., ..., 1., 0., 0.],
        [0., 0., 0., ..., 0., 1., 0.],
        [0., 0., 0., ..., 1., 0., 0.]]], dtype=float32)>
```

```
# Building the RNN Model
```

```
####
X_one_hot.shape
####
```

```
TensorShape([48, 3, 17])
```

```
model = Sequential()
model.add(SimpleRNN(50, input_shape=(seq_length, len(chars)), activation='relu'))
model.add(Dense(len(chars), activation='softmax'))
```

```
/usr/local/lib/python3.12/dist-packages/keras/src/layers/rnn/rnn.py:199: UserWarning: Do not pass an `input_shape
  super().__init__(**kwargs)
```

```
####
model = Sequential()
model.add(SimpleRNN(50, input_shape=(seq_length, len(chars)), activation='relu'))
model.add(Dense(len(chars), activation='softmax'))
####
```

```
/usr/local/lib/python3.12/dist-packages/keras/src/layers/rnn/rnn.py:199: UserWarning: Do not pass an `input_shape
  super().__init__(**kwargs)
```

```
# Compiling and Training the Model
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.fit(X_one_hot, y_one_hot, epochs=100)
```

```
Epoch 1/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 2s 33ms/step - accuracy: 0.0000e+00 - loss: 2.8499
Epoch 2/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step - accuracy: 0.0243 - loss: 2.8210
Epoch 3/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step - accuracy: 0.0243 - loss: 2.8008
Epoch 4/100
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step - accuracy: 0.0972 - loss: 2.7762
Epoch 5/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - accuracy: 0.1007 - loss: 2.7662
Epoch 6/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step - accuracy: 0.1493 - loss: 2.7420
Epoch 7/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - accuracy: 0.1667 - loss: 2.7244
Epoch 8/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step - accuracy: 0.2361 - loss: 2.6914
Epoch 9/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - accuracy: 0.2465 - loss: 2.6926
Epoch 10/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - accuracy: 0.3229 - loss: 2.6562
Epoch 11/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step - accuracy: 0.3368 - loss: 2.6432
Epoch 12/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step - accuracy: 0.3021 - loss: 2.6214
Epoch 13/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - accuracy: 0.3333 - loss: 2.5903
Epoch 14/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step - accuracy: 0.3368 - loss: 2.5739
Epoch 15/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step - accuracy: 0.3160 - loss: 2.5532
Epoch 16/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step - accuracy: 0.3264 - loss: 2.5360
Epoch 17/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step - accuracy: 0.3646 - loss: 2.5127
Epoch 18/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step - accuracy: 0.3750 - loss: 2.4818
Epoch 19/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step - accuracy: 0.4410 - loss: 2.4332
Epoch 20/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - accuracy: 0.3889 - loss: 2.4292
Epoch 21/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step - accuracy: 0.4132 - loss: 2.3931
Epoch 22/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step - accuracy: 0.4236 - loss: 2.3625
Epoch 23/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step - accuracy: 0.4132 - loss: 2.3340
Epoch 24/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step - accuracy: 0.4340 - loss: 2.3042
Epoch 25/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step - accuracy: 0.4271 - loss: 2.2879
Epoch 26/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step - accuracy: 0.4271 - loss: 2.2599
Epoch 27/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step - accuracy: 0.4792 - loss: 2.1826
Epoch 28/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - accuracy: 0.4722 - loss: 2.1709
Epoch 29/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step - accuracy: 0.4410 - loss: 2.1835
```

```
####
# Compiling and Training the Model
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.fit(X_one_hot, y_one_hot, epochs=100)
####
```

```
Epoch 1/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 2s 40ms/step - accuracy: 0.0407 - loss: 2.8634
Epoch 2/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - accuracy: 0.0814 - loss: 2.8347
Epoch 3/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - accuracy: 0.1326 - loss: 2.8027
Epoch 4/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.1430 - loss: 2.7764
Epoch 5/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - accuracy: 0.1174 - loss: 2.7529
Epoch 6/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - accuracy: 0.1326 - loss: 2.7334
Epoch 7/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - accuracy: 0.1326 - loss: 2.7075
Epoch 8/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.1430 - loss: 2.6938
Epoch 9/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.1686 - loss: 2.6628
Epoch 10/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.1837 - loss: 2.6395
Epoch 11/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.1733 - loss: 2.6287
Epoch 12/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.1989 - loss: 2.6015
Epoch 13/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2756 - loss: 2.5675
Epoch 14/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.2860 - loss: 2.5507
Epoch 15/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2652 - loss: 2.5197
Epoch 16/100
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.2756 - loss: 2.4971
Epoch 17/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step - accuracy: 0.3371 - loss: 2.4508
Epoch 18/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step - accuracy: 0.3475 - loss: 2.4268
Epoch 19/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step - accuracy: 0.3059 - loss: 2.4131
Epoch 20/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step - accuracy: 0.3419 - loss: 2.3786
Epoch 21/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step - accuracy: 0.2955 - loss: 2.3449
Epoch 22/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step - accuracy: 0.3267 - loss: 2.3044
Epoch 23/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step - accuracy: 0.3570 - loss: 2.2823
Epoch 24/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step - accuracy: 0.3930 - loss: 2.2594
Epoch 25/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step - accuracy: 0.3920 - loss: 2.2443
Epoch 26/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - accuracy: 0.4441 - loss: 2.1811
Epoch 27/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - accuracy: 0.4129 - loss: 2.1586
Epoch 28/100
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step - accuracy: 0.4536 - loss: 2.1498
Epoch 29/100
```

```
start_seq = "This is G"
generated_text = start_seq

for i in range(50):
  x = np.array([[char_to_index[char] for char in generated_text[-seq_length:]]])
  x_one_hot = tf.one_hot(x, len(chars))
  print(x_one_hot.shape)
  prediction = model.predict(x_one_hot)
  next_index = np.argmax(prediction)
  next_char = index_to_char[next_index]
  generated_text += next_char

print("Generated Text:")
print(generated_text)
```

```
(1, 3, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 186ms/step
(1, 3, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 56ms/step
(1, 3, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
(1, 3, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 55ms/step
(1, 3, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
(1, 3, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
(1, 3, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
(1, 3, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
(1, 3, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 61ms/step
(1, 3, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
(1, 3, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 69ms/step
(1, 3, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
(1, 3, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
(1, 3, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step
(1, 3, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
(1, 3, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step
(1, 3, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
(1, 3, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
(1, 3, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
(1, 3, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step
(1, 3, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
(1, 3, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
(1, 3, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step
```

```
(1, 3, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step
(1, 3, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step
(1, 3, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step
(1, 3, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
(1, 3, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step
(1, 3, 17)
```

```
####
start_seq[-seq_length:]
####
```

```
's G'
```

```
####
# Test-1

start_seq = "This is G"
generated_text = start_seq

for i in range(50):
  x = np.array([[char_to_index[char] for char in generated_text[-seq_length:]]])
  x_one_hot = tf.one_hot(x, len(chars))
  print(x_one_hot.shape)
  prediction = model.predict(x_one_hot)
  next_index = np.argmax(prediction)
  next_char = index_to_char[next_index]
  generated_text += next_char

print("Generated Text:")
print(generated_text)

####
```

```
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step
(1, 7, 17)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 65ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 64ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 58ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 61ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 64ms/step
```

```
####
# Test-2

start_seq = "Software "
generated_text = start_seq

for i in range(50):
  x = np.array([[char_to_index[char] for char in generated_text[-seq_length:]]])
  x_one_hot = tf.one_hot(x, len(chars))
  print(x_one_hot.shape)
  prediction = model.predict(x_one_hot)
  next_index = np.argmax(prediction)
  next_char = index_to_char[next_index]
  generated_text += next_char

print("Generated Text:")
print(generated_text)


####
```

```
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 59ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 150ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 56ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 63ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 61ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 55ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 64ms/step
(1, 7, 17)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 96ms/step
(1, 7, 17)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 65ms/step
```