IMAGE AUGUMENTATION

```
!pip install split-folders
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting split-folders
  Downloading split_folders-0.5.1-py3-none-any.whl (8.4 kB)
Installing collected packages: split-folders
Successfully installed split-folders-0.5.1
```

```
import splitfolders
# To only split into training and validation set, set a tuple to `ratio`, i.e, `(.8, .2)`.
splitfolders.ratio("/content/drive/MyDrive/flowers", output="/content/drive/MyDrive/flowers",seed=1337, ratio=(.8, .1, .1), group_prefix=None, move=False)
```

```
Copying files: 0 files [00:00, ? files/s]
```

```
import keras
from keras.preprocessing.image import ImageDataGenerator
```

```
def getdata():
  train_datagen=ImageDataGenerator (rescale=1./255,
  shear_range=0.2,
  rotation_range=180,
  zoom_range=0.2,
  horizontal_flip=True)
  test_datagen=ImageDataGenerator (rescale=1./255)
  x_train = train_datagen.flow_from_directory (r'/content/output/train', target_size = (64,64), batch_size = 32, class_mode= 'categorical')
  x_test = test_datagen.flow_from_directory(r'/content/output/test', target_size = (64,64), batch_size = 32, class_mode= 'categorical')
  return x_train,x_test;
```

CREATE MODEL

```
#To define linear intialisan
from keras.models import Sequential
#To add Layers import Dense
from keras.layers import Dense
#To create Convolution kernel import Convolution2D
from keras.layers import Convolution2D
#import Maxpooling layer
from keras.layers import MaxPooling2D
#import Flatten Layer
from keras.layers import Flatten
```

```python
import warnings
warnings.filterwarnings('ignore')
```

```python
model=Sequential()
```

ADDING LAYERS

```python
#add convolutional layer
model.add(Convolution2D(112, (7,7),input_shape=(64,64,3),
activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
```

```python
#add hidden layer
model.add(Dense(32,activation='relu'))
model.add(Dense(64,activation='relu'))
```

COMPILE THE MODEL

```python
#add output Layer
model.add(Dense(5, activation='softmax'))
model.compile(loss = 'categorical_crossentropy', optimizer = "adam",
metrics = ["accuracy"])
```

```python
model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_2 (Conv2D)           (None, 58, 58, 112)       16576

 max_pooling2d_2 (MaxPooling  (None, 29, 29, 112)      0
 2D)

 flatten_2 (Flatten)         (None, 94192)             0

 dense_4 (Dense)             (None, 32)                3014176

 dense_5 (Dense)             (None, 64)                2112

 dense_6 (Dense)             (None, 5)                 325

=================================================================
Total params: 3,033,189
Trainable params: 3,033,189
Non-trainable params: 0
_________________________________________________________________
```

FIT THE MODEL

```python
x_train,x_test=getdata()
tom=model.fit_generator(x_train , steps_per_epoch=len(x_train),
epochs=100, validation_data=x_test, validation_steps=len(x_test))
```

```
Found 3463 images belonging to 5 classes.
Found 506 images belonging to 5 classes.
Epoch 1/100
109/109 [==============================] - 44s 406ms/step - loss:
1.1920 - accuracy: 0.4973 - val_loss: 1.3547 - val_accuracy: 0.4531
Epoch 2/100
109/109 [==============================] - 44s 404ms/step - loss:
1.1537 - accuracy: 0.5299 - val_loss: 1.1252 - val_accuracy: 0.5938
Epoch 3/100
109/109 [==============================] - 44s 407ms/step - loss:
1.1497 - accuracy: 0.5261 - val_loss: 1.3031 - val_accuracy: 0.4453
Epoch 4/100
109/109 [==============================] - 43s 396ms/step - loss:
1.1235 - accuracy: 0.5334 - val_loss: 1.2084 - val_accuracy: 0.5234
Epoch 5/100
109/109 [==============================] - 49s 444ms/step - loss:
1.1239 - accuracy: 0.5458 - val_loss: 0.9888 - val_accuracy: 0.5859
Epoch 6/100
109/109 [==============================] - 45s 408ms/step - loss:
1.0538 - accuracy: 0.5712 - val_loss: 1.0753 - val_accuracy: 0.6016
Epoch 7/100
109/109 [==============================] - 47s 434ms/step - loss:
1.0273 - accuracy: 0.5842 - val_loss: 1.0493 - val_accuracy: 0.6094
Epoch 8/100
109/109 [==============================] - 43s 395ms/step - loss:
0.9981 - accuracy: 0.6058 - val_loss: 0.9783 - val_accuracy: 0.5938
Epoch 9/100
109/109 [==============================] - 44s 399ms/step - loss:
1.0165 - accuracy: 0.5862 - val_loss: 1.0112 - val_accuracy: 0.6094
Epoch 10/100
109/109 [==============================] - 44s 398ms/step - loss:
0.9886 - accuracy: 0.6093 - val_loss: 1.0801 - val_accuracy: 0.5391
Epoch 11/100
109/109 [==============================] - 43s 396ms/step - loss:
0.9924 - accuracy: 0.6032 - val_loss: 0.9350 - val_accuracy: 0.7500
Epoch 12/100
109/109 [==============================] - 43s 395ms/step - loss:
0.9673 - accuracy: 0.6157 - val_loss: 1.2098 - val_accuracy: 0.5078
Epoch 13/100
109/109 [==============================] - 43s 395ms/step - loss:
0.9592 - accuracy: 0.6128 - val_loss: 1.0900 - val_accuracy: 0.6250
Epoch 14/100
109/109 [==============================] - 44s 399ms/step - loss:
0.9635 - accuracy: 0.6107 - val_loss: 0.8533 - val_accuracy: 0.6484
Epoch 15/100
109/109 [==============================] - 44s 398ms/step - loss:
```

```
0.9537 - accuracy: 0.6122 - val_loss: 0.9958 - val_accuracy: 0.5859
Epoch 16/100
109/109 [==============================] - 43s 395ms/step - loss:
0.9313 - accuracy: 0.6333 - val_loss: 1.2101 - val_accuracy: 0.5469
Epoch 17/100
109/109 [==============================] - 43s 394ms/step - loss:
0.9362 - accuracy: 0.6266 - val_loss: 0.9873 - val_accuracy: 0.6250
Epoch 18/100
109/109 [==============================] - 44s 398ms/step - loss:
0.9286 - accuracy: 0.6298 - val_loss: 0.9595 - val_accuracy: 0.6328
Epoch 19/100
109/109 [==============================] - 44s 398ms/step - loss:
0.9231 - accuracy: 0.6362 - val_loss: 1.0403 - val_accuracy: 0.5938
Epoch 20/100
109/109 [==============================] - 43s 401ms/step - loss:
0.9039 - accuracy: 0.6445 - val_loss: 0.9216 - val_accuracy: 0.6562
Epoch 21/100
109/109 [==============================] - 44s 398ms/step - loss:
0.8973 - accuracy: 0.6442 - val_loss: 0.9285 - val_accuracy: 0.6172
Epoch 22/100
109/109 [==============================] - 44s 401ms/step - loss:
0.9138 - accuracy: 0.6416 - val_loss: 1.1500 - val_accuracy: 0.6016
Epoch 23/100
109/109 [==============================] - 44s 400ms/step - loss:
0.8856 - accuracy: 0.6543 - val_loss: 1.0287 - val_accuracy: 0.6562
Epoch 24/100
109/109 [==============================] - 44s 402ms/step - loss:
0.8831 - accuracy: 0.6512 - val_loss: 1.1725 - val_accuracy: 0.5781
Epoch 25/100
109/109 [==============================] - 44s 400ms/step - loss:
0.8939 - accuracy: 0.6451 - val_loss: 1.0966 - val_accuracy: 0.5625
Epoch 26/100
109/109 [==============================] - 44s 399ms/step - loss:
0.8821 - accuracy: 0.6543 - val_loss: 0.9534 - val_accuracy: 0.6719
Epoch 27/100
109/109 [==============================] - 43s 398ms/step - loss:
0.8715 - accuracy: 0.6581 - val_loss: 0.8288 - val_accuracy: 0.6406
Epoch 28/100
109/109 [==============================] - 44s 400ms/step - loss:
0.8829 - accuracy: 0.6546 - val_loss: 0.8732 - val_accuracy: 0.6797
Epoch 29/100
109/109 [==============================] - 43s 396ms/step - loss:
0.8376 - accuracy: 0.6720 - val_loss: 1.1241 - val_accuracy: 0.5547
Epoch 30/100
109/109 [==============================] - 44s 398ms/step - loss:
0.8462 - accuracy: 0.6656 - val_loss: 1.0432 - val_accuracy: 0.6016
Epoch 31/100
109/109 [==============================] - 43s 396ms/step - loss:
0.8577 - accuracy: 0.6593 - val_loss: 0.8796 - val_accuracy: 0.6719
Epoch 32/100
```

```
109/109 [==============================] - 43s 398ms/step - loss:
0.8250 - accuracy: 0.6769 - val_loss: 0.9506 - val_accuracy: 0.6562
Epoch 33/100
109/109 [==============================] - 43s 397ms/step - loss:
0.8461 - accuracy: 0.6717 - val_loss: 0.9850 - val_accuracy: 0.6562
Epoch 34/100
109/109 [==============================] - 43s 395ms/step - loss:
0.8492 - accuracy: 0.6702 - val_loss: 1.0259 - val_accuracy: 0.5703
Epoch 35/100
109/109 [==============================] - 43s 396ms/step - loss:
0.8470 - accuracy: 0.6593 - val_loss: 1.0286 - val_accuracy: 0.6094
Epoch 36/100
109/109 [==============================] - 43s 395ms/step - loss:
0.8540 - accuracy: 0.6610 - val_loss: 0.9687 - val_accuracy: 0.6562
Epoch 37/100
109/109 [==============================] - 43s 397ms/step - loss:
0.8195 - accuracy: 0.6694 - val_loss: 0.8824 - val_accuracy: 0.6250
Epoch 38/100
109/109 [==============================] - 43s 394ms/step - loss:
0.8119 - accuracy: 0.6705 - val_loss: 0.8769 - val_accuracy: 0.6406
Epoch 39/100
109/109 [==============================] - 43s 394ms/step - loss:
0.7992 - accuracy: 0.6803 - val_loss: 0.8641 - val_accuracy: 0.6641
Epoch 40/100
109/109 [==============================] - 43s 396ms/step - loss:
0.8162 - accuracy: 0.6728 - val_loss: 0.7716 - val_accuracy: 0.6953
Epoch 41/100
109/109 [==============================] - 43s 396ms/step - loss:
0.8160 - accuracy: 0.6841 - val_loss: 1.1424 - val_accuracy: 0.5859
Epoch 42/100
109/109 [==============================] - 43s 398ms/step - loss:
0.8172 - accuracy: 0.6824 - val_loss: 0.9372 - val_accuracy: 0.6641
Epoch 43/100
109/109 [==============================] - 43s 397ms/step - loss:
0.8193 - accuracy: 0.6751 - val_loss: 1.1626 - val_accuracy: 0.5938
Epoch 44/100
109/109 [==============================] - 44s 399ms/step - loss:
0.8174 - accuracy: 0.6757 - val_loss: 0.8816 - val_accuracy: 0.6719
Epoch 45/100
109/109 [==============================] - 43s 395ms/step - loss:
0.8147 - accuracy: 0.6774 - val_loss: 0.8948 - val_accuracy: 0.6641
Epoch 46/100
109/109 [==============================] - 43s 396ms/step - loss:
0.7990 - accuracy: 0.6818 - val_loss: 0.9974 - val_accuracy: 0.6094
Epoch 47/100
109/109 [==============================] - 43s 396ms/step - loss:
0.7928 - accuracy: 0.6835 - val_loss: 0.9565 - val_accuracy: 0.6641
Epoch 48/100
109/109 [==============================] - 43s 395ms/step - loss:
0.7868 - accuracy: 0.6956 - val_loss: 0.7995 - val_accuracy: 0.6562
```

```
Epoch 49/100
109/109 [==============================] - 43s 395ms/step - loss:
0.7722 - accuracy: 0.6956 - val_loss: 0.7633 - val_accuracy: 0.6875
Epoch 50/100
109/109 [==============================] - 43s 398ms/step - loss:
0.7818 - accuracy: 0.6968 - val_loss: 0.9549 - val_accuracy: 0.7266
Epoch 51/100
109/109 [==============================] - 43s 399ms/step - loss:
0.7671 - accuracy: 0.7055 - val_loss: 1.0400 - val_accuracy: 0.6484
Epoch 52/100
109/109 [==============================] - 44s 398ms/step - loss:
0.7725 - accuracy: 0.6942 - val_loss: 0.9840 - val_accuracy: 0.6094
Epoch 53/100
109/109 [==============================] - 43s 397ms/step - loss:
0.7723 - accuracy: 0.6945 - val_loss: 0.8121 - val_accuracy: 0.7266
Epoch 54/100
109/109 [==============================] - 44s 398ms/step - loss:
0.7544 - accuracy: 0.7005 - val_loss: 0.8650 - val_accuracy: 0.6719
Epoch 55/100
109/109 [==============================] - 43s 400ms/step - loss:
0.7413 - accuracy: 0.7049 - val_loss: 1.0008 - val_accuracy: 0.6484
Epoch 56/100
109/109 [==============================] - 44s 400ms/step - loss:
0.7834 - accuracy: 0.6896 - val_loss: 1.0291 - val_accuracy: 0.6641
Epoch 57/100
109/109 [==============================] - 44s 402ms/step - loss:
0.7592 - accuracy: 0.6936 - val_loss: 0.8758 - val_accuracy: 0.6875
Epoch 58/100
109/109 [==============================] - 43s 398ms/step - loss:
0.7563 - accuracy: 0.7086 - val_loss: 1.0372 - val_accuracy: 0.6016
Epoch 59/100
109/109 [==============================] - 43s 397ms/step - loss:
0.7581 - accuracy: 0.6965 - val_loss: 0.9029 - val_accuracy: 0.6797
Epoch 60/100
109/109 [==============================] - 46s 423ms/step - loss:
0.7518 - accuracy: 0.7023 - val_loss: 0.9261 - val_accuracy: 0.6406
Epoch 61/100
109/109 [==============================] - 43s 393ms/step - loss:
0.7413 - accuracy: 0.7098 - val_loss: 1.0991 - val_accuracy: 0.5547
Epoch 62/100
109/109 [==============================] - 43s 395ms/step - loss:
0.7408 - accuracy: 0.7092 - val_loss: 1.0991 - val_accuracy: 0.6250
Epoch 63/100
109/109 [==============================] - 43s 392ms/step - loss:
0.7402 - accuracy: 0.7138 - val_loss: 1.0513 - val_accuracy: 0.6328
Epoch 64/100
109/109 [==============================] - 43s 391ms/step - loss:
0.7480 - accuracy: 0.7014 - val_loss: 1.0600 - val_accuracy: 0.7031
Epoch 65/100
109/109 [==============================] - 43s 390ms/step - loss:
```

```
0.7352 - accuracy: 0.7083 - val_loss: 1.0083 - val_accuracy: 0.6641
Epoch 66/100
109/109 [==============================] - 43s 393ms/step - loss:
0.7340 - accuracy: 0.7101 - val_loss: 0.8945 - val_accuracy: 0.6953
Epoch 67/100
109/109 [==============================] - 43s 393ms/step - loss:
0.7199 - accuracy: 0.7205 - val_loss: 0.9399 - val_accuracy: 0.6406
Epoch 68/100
109/109 [==============================] - 43s 395ms/step - loss:
0.7255 - accuracy: 0.7167 - val_loss: 0.8557 - val_accuracy: 0.7031
Epoch 69/100
109/109 [==============================] - 43s 393ms/step - loss:
0.7188 - accuracy: 0.7156 - val_loss: 1.0336 - val_accuracy: 0.6641
Epoch 70/100
109/109 [==============================] - 43s 395ms/step - loss:
0.7178 - accuracy: 0.7104 - val_loss: 0.8658 - val_accuracy: 0.6953
Epoch 71/100
109/109 [==============================] - 43s 396ms/step - loss:
0.7259 - accuracy: 0.7164 - val_loss: 1.0510 - val_accuracy: 0.6094
Epoch 72/100
109/109 [==============================] - 43s 393ms/step - loss:
0.7390 - accuracy: 0.7083 - val_loss: 0.9564 - val_accuracy: 0.7031
Epoch 73/100
109/109 [==============================] - 43s 393ms/step - loss:
0.7039 - accuracy: 0.7216 - val_loss: 1.1533 - val_accuracy: 0.6250
Epoch 74/100
109/109 [==============================] - 43s 393ms/step - loss:
0.7190 - accuracy: 0.7156 - val_loss: 0.8456 - val_accuracy: 0.6875
Epoch 75/100
109/109 [==============================] - 43s 393ms/step - loss:
0.7236 - accuracy: 0.7133 - val_loss: 0.9153 - val_accuracy: 0.6797
Epoch 76/100
109/109 [==============================] - 43s 393ms/step - loss:
0.6915 - accuracy: 0.7242 - val_loss: 0.8804 - val_accuracy: 0.6484
Epoch 77/100
109/109 [==============================] - 43s 390ms/step - loss:
0.7096 - accuracy: 0.7213 - val_loss: 1.1283 - val_accuracy: 0.6562
Epoch 78/100
109/109 [==============================] - 43s 392ms/step - loss:
0.7026 - accuracy: 0.7251 - val_loss: 0.8741 - val_accuracy: 0.6406
Epoch 79/100
109/109 [==============================] - 43s 391ms/step - loss:
0.6858 - accuracy: 0.7248 - val_loss: 0.9821 - val_accuracy: 0.6484
Epoch 80/100
109/109 [==============================] - 43s 394ms/step - loss:
0.7025 - accuracy: 0.7133 - val_loss: 0.8709 - val_accuracy: 0.7266
Epoch 81/100
109/109 [==============================] - 43s 395ms/step - loss:
0.7059 - accuracy: 0.7260 - val_loss: 1.0875 - val_accuracy: 0.6484
Epoch 82/100
```

```
109/109 [==============================] - 43s 395ms/step - loss:
0.6870 - accuracy: 0.7300 - val_loss: 0.9539 - val_accuracy: 0.6797
Epoch 83/100
109/109 [==============================] - 43s 395ms/step - loss:
0.6774 - accuracy: 0.7378 - val_loss: 0.9052 - val_accuracy: 0.6719
Epoch 84/100
109/109 [==============================] - 43s 394ms/step - loss:
0.6892 - accuracy: 0.7277 - val_loss: 0.7434 - val_accuracy: 0.7422
Epoch 85/100
109/109 [==============================] - 43s 394ms/step - loss:
0.6687 - accuracy: 0.7372 - val_loss: 0.9350 - val_accuracy: 0.6562
Epoch 86/100
109/109 [==============================] - 43s 394ms/step - loss:
0.6689 - accuracy: 0.7320 - val_loss: 1.2160 - val_accuracy: 0.6953
Epoch 87/100
109/109 [==============================] - 43s 394ms/step - loss:
0.6861 - accuracy: 0.7306 - val_loss: 1.0035 - val_accuracy: 0.7031
Epoch 88/100
109/109 [==============================] - 43s 393ms/step - loss:
0.6676 - accuracy: 0.7392 - val_loss: 1.0701 - val_accuracy: 0.6641
Epoch 89/100
109/109 [==============================] - 44s 399ms/step - loss:
0.6444 - accuracy: 0.7401 - val_loss: 1.4203 - val_accuracy: 0.6094
Epoch 90/100
109/109 [==============================] - 43s 396ms/step - loss:
0.6684 - accuracy: 0.7375 - val_loss: 0.8594 - val_accuracy: 0.6953
Epoch 91/100
109/109 [==============================] - 43s 397ms/step - loss:
0.6536 - accuracy: 0.7413 - val_loss: 1.0417 - val_accuracy: 0.7031
Epoch 92/100
109/109 [==============================] - 43s 397ms/step - loss:
0.6782 - accuracy: 0.7349 - val_loss: 1.2230 - val_accuracy: 0.6172
Epoch 93/100
109/109 [==============================] - 43s 393ms/step - loss:
0.6777 - accuracy: 0.7283 - val_loss: 1.3130 - val_accuracy: 0.6641
Epoch 94/100
109/109 [==============================] - 43s 392ms/step - loss:
0.6665 - accuracy: 0.7320 - val_loss: 1.1505 - val_accuracy: 0.6172
Epoch 95/100
109/109 [==============================] - 43s 392ms/step - loss:
0.6543 - accuracy: 0.7436 - val_loss: 0.9266 - val_accuracy: 0.6719
Epoch 96/100
109/109 [==============================] - 43s 394ms/step - loss:
0.6644 - accuracy: 0.7303 - val_loss: 1.0623 - val_accuracy: 0.6953
Epoch 97/100
109/109 [==============================] - 43s 395ms/step - loss:
0.6555 - accuracy: 0.7401 - val_loss: 1.1927 - val_accuracy: 0.6484
Epoch 98/100
109/109 [==============================] - 43s 393ms/step - loss:
0.6685 - accuracy: 0.7375 - val_loss: 1.2203 - val_accuracy: 0.6875
```

```
Epoch 99/100
109/109 [==============================] - 43s 393ms/step - loss:
0.6551 - accuracy: 0.7404 - val_loss: 1.0924 - val_accuracy: 0.6797
Epoch 100/100
109/109 [==============================] - 43s 395ms/step - loss:
0.6473 - accuracy: 0.7387 - val_loss: 1.1365 - val_accuracy: 0.6406
```

SAVE THE MODEL

```
model.save("/content/drive/MyDrive/flowermodel.h5")
```

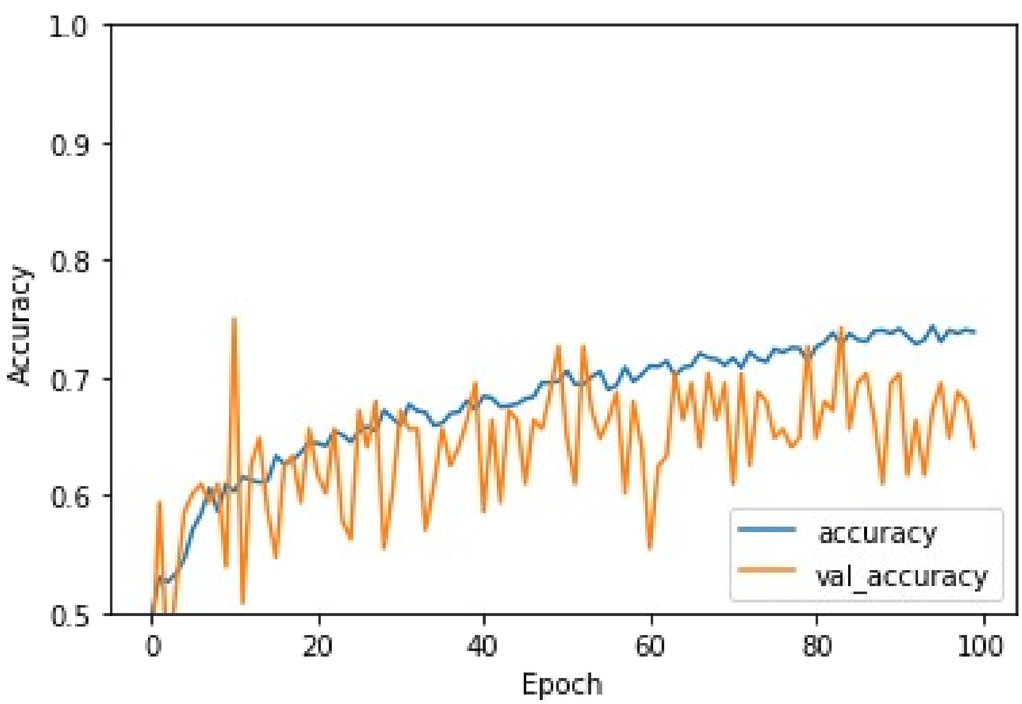
TEST THE MODEL

```
import matplotlib.pyplot as plt

plt.plot(tom.history['accuracy'], label='accuracy')
plt.plot(tom.history['val_accuracy'], label = 'val_accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.ylim([0.5, 1])
plt.legend(loc='lower right')

test_loss, test_acc = model.evaluate(x_test, verbose=2)
```

```
16/16 - 3s - loss: 1.0302 - accuracy: 0.6877 - 3s/epoch - 186ms/step
```




```
print(test_acc)
```

```
0.687747061252594
```