In [ ]:
```
'''
ref1(dataset): http://www.cis.fordham.edu/wisdm/dataset.php
ref2(paper): http://www.cis.fordham.edu/wisdm/includes/files/sensorKDD-2010.pdf
ref3:
https://towardsdatascience.com/human-activity-recognition-har-tutorial-with-keras-
ref4:
https://blog.goodaudience.com/introduction-to-1d-convolutional-neural-networks-in-
'''
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt

#1
##gpus = tf.config.experimental.list_physical_devices('GPU')
##tf.config.experimental.set_memory_growth(gpus[0], True)

#2. 데이터 파싱 및 전처리
def parse_end(s):
    try:
        return float(s[-1])
    except:
        return np.nan

def read_data(file_path):
# columns: 'user', 'activity', 'timestamp', 'x-accl', 'y-accl', 'z-accl';
    labels =     {'Walking'   :0,
                  'Jogging'   :1,
                  'Upstairs'  :2,
                  'Sitting'    :3,
                  'Downstairs':4,
                  'Standing'   :5}
    data = np.loadtxt(file_path, delimiter=",", usecols=(0,1, 3, 4, 5), # timestam
                      converters={1:lambda name: labels[name.decode()],
                                  5: parse_end})
    data = data[~np.isnan(data).any(axis=1)] # nan 값을 포함한 행 제거
    return data

# 데이터 로드
data = read_data("./DATA/WISDM_ar_v1.1/WISDM_ar_v1.1_raw.txt")
##print("user:",     np.unique(data[:,0])) # 36 users
##print("activity:", np.unique(data[:,1])) # 6 activity

#3: normalize x, y, z
mean = np.mean(data[:,2:], axis = 0)
std  = np.std(data[:,2:], axis = 0)
data[:,2:] = (data[:,2:]-mean)/std
##data[:,2:] = (data[:,2:])/np.max(data[:,2:], axis = 0) # [ -1, 1]
##print(np.mean(data[:, 2:], axis = 0)) # [0, 0, 0]
##print(np.std(data[:, 2:], axis = 0))  # [1, 1, 1]

#데이터 분할 (훈련 데이터와 테스트 데이터)
x_train = data[data[:,0] <= 28] #[28, 36]
x_test  = data[data[:,0]  > 28]

#4. 데이터 세그먼트화 및 라벨링 (-1, TIME_PERIODS, 3)
TIME_PERIODS  = 80  # 세그먼트 길이
STEP_DISTANCE = 40 # if STEP_DISTANCE = TIME_PERIODS, then no overlap
def data_segments(data):
```

```
    segments = []
    labels = []
    for i in range(0, len(data)-TIME_PERIODS,  STEP_DISTANCE):
        X = data[i:i+TIME_PERIODS, 2:].tolist() # x, y, z

        # label as the most activity in this segment
        values, counts = np.unique(data[i:i+TIME_PERIODS, 1], return_counts=True)
        label = values[np.argmax(counts)] # 세그먼트 내 가장 빈번한 활동으로 라벨

        segments.append(X)
        labels.append(label)

    # reshape (-1, TIME_PERIODS, 3)
    segments = np.array(segments, dtype= np.float32).reshape(-1, TIME_PERIODS, 3)
    labels   = np.asarray(labels)
    return segments, labels

# 훈련 및 테스트 데이터 세그먼트화
x_train, y_train = data_segments(x_train)
x_test, y_test = data_segments(x_test)
print("x_train.shape=", x_train.shape)
print("x_test.shape=",  x_test.shape)

# 라벨 원-핫 인코딩
y_train = tf.keras.utils.to_categorical(y_train)
y_test = tf.keras.utils.to_categorical(y_test)
##print("y_train=", y_train)
##print("y_test=", y_test)

#5. 1D CNN 모델 생성
model = tf.keras.Sequential()
model.add(tf.keras.layers.Input(shape=(TIME_PERIODS,3))) # shape=(80,3)
model.add(tf.keras.layers.Conv1D(filters=100,
                                 kernel_size=11, activation='relu'))
model.add(tf.keras.layers.MaxPool1D())
model.add(tf.keras.layers.BatchNormalization())

model.add(tf.keras.layers.Conv1D(filters=10, kernel_size=5, activation='relu'))
model.add(tf.keras.layers.MaxPool1D())
model.add(tf.keras.layers.Dropout( rate=0.5))

model.add(tf.keras.layers.Flatten())
model.add(tf.keras.layers.Dense(units=6, activation='softmax'))
model.summary()

#6. 모델 컴파일 및 학습
opt = tf.keras.optimizers.RMSprop(learning_rate=0.01)
model.compile(optimizer=opt, loss='categorical_crossentropy', metrics=['accuracy']
ret = model.fit(x_train, y_train, epochs=100, batch_size=400,
               validation_data = (x_test, y_test), verbose=2)  # validation_split=

#7. 모델 평가
train_loss, train_acc = model.evaluate(x_train, y_train, verbose=2)
test_loss, test_acc = model.evaluate(x_test,  y_test, verbose=2)

#8. 정확도 및 손실 시각화
plt.title("Accuracy")
plt.plot(ret.history['accuracy'],     "b-", label="train accuracy")
plt.plot(ret.history['val_accuracy'], "r-", label="val accuracy")
plt.plot(ret.history['loss'],         "g-", label="train loss")
```

```
plt.xlabel('epochs')
plt.ylabel('accuracy')
plt.legend(loc="best")
plt.show()


#9. 샘플 활동 데이터 시각화
activity = ('Walking','Jogging','Upstairs', 'Sitting','Downstairs','Standing')
train_label  = np.argmax(y_train, axis = 1)

plot_data =[]
n = 1
for i in range(6):
    plot_data.append(np.where(train_label == i)[0][n]) # n-th data

fig, ax = plt.subplots(6, sharex=True,  sharey=True)
fig.tight_layout()
for i in range(6):
    k = plot_data[i]
    ax[i].plot(x_train[k], label=activity[i])
    ax[i].set_title(activity[i])
plt.show()
```

```
x_train.shape= (20868, 80, 3)
x_test.shape= (6584, 80, 3)
```

Model: "sequential"

Layer (type)	Output Shape
conv1d (Conv1D)	(None, 70, 100)
max_pooling1d (MaxPooling1D)	(None, 35, 100)
batch_normalization (BatchNormalization)	(None, 35, 100)
conv1d_1 (Conv1D)	(None, 31, 10)
max_pooling1d_1 (MaxPooling1D)	(None, 15, 10)
dropout (Dropout)	(None, 15, 10)
flatten (Flatten)	(None, 150)
dense (Dense)	(None, 6)

Total params: 9,716 (37.95 KB)

Trainable params: 9,516 (37.17 KB)

Non-trainable params: 200 (800.00 B)

```
Epoch 1/100
53/53 - 2s - 35ms/step - accuracy: 0.6431 - loss: 1.2171 - val_accuracy: 0.5855 - v
al_loss: 1.1468
Epoch 2/100
53/53 - 1s - 18ms/step - accuracy: 0.7228 - loss: 0.7882 - val_accuracy: 0.5536 - v
al_loss: 1.4088
Epoch 3/100
53/53 - 1s - 18ms/step - accuracy: 0.7565 - loss: 0.6379 - val_accuracy: 0.6615 - v
al_loss: 1.0196
Epoch 4/100
53/53 - 1s - 18ms/step - accuracy: 0.8012 - loss: 0.5421 - val_accuracy: 0.7005 - v
al_loss: 0.8214
Epoch 5/100
53/53 - 1s - 18ms/step - accuracy: 0.8585 - loss: 0.4175 - val_accuracy: 0.7269 - v
al_loss: 1.0172
Epoch 6/100
53/53 - 1s - 17ms/step - accuracy: 0.8732 - loss: 0.3732 - val_accuracy: 0.7584 - v
al_loss: 0.7621
Epoch 7/100
53/53 - 1s - 17ms/step - accuracy: 0.8857 - loss: 0.3455 - val_accuracy: 0.7205 - v
al_loss: 0.9322
Epoch 8/100
53/53 - 1s - 17ms/step - accuracy: 0.8923 - loss: 0.3297 - val_accuracy: 0.7732 - v
al_loss: 0.9956
Epoch 9/100
53/53 - 1s - 18ms/step - accuracy: 0.8972 - loss: 0.3579 - val_accuracy: 0.8215 - v
al_loss: 1.4567
Epoch 10/100
53/53 - 1s - 18ms/step - accuracy: 0.9058 - loss: 0.2998 - val_accuracy: 0.8077 - v
al_loss: 1.2650
Epoch 11/100
53/53 - 1s - 17ms/step - accuracy: 0.9104 - loss: 0.3263 - val_accuracy: 0.7840 - v
al_loss: 1.3688
Epoch 12/100
53/53 - 1s - 17ms/step - accuracy: 0.9165 - loss: 0.2547 - val_accuracy: 0.8035 - v
al_loss: 1.1075
Epoch 13/100
53/53 - 1s - 18ms/step - accuracy: 0.9160 - loss: 0.2554 - val_accuracy: 0.8220 - v
al_loss: 1.4018
Epoch 14/100
53/53 - 1s - 18ms/step - accuracy: 0.9223 - loss: 0.2385 - val_accuracy: 0.7881 - v
al_loss: 1.3282
Epoch 15/100
53/53 - 1s - 18ms/step - accuracy: 0.9242 - loss: 0.2392 - val_accuracy: 0.7968 - v
al_loss: 1.2310
Epoch 16/100
53/53 - 1s - 17ms/step - accuracy: 0.9223 - loss: 0.2394 - val_accuracy: 0.7910 - v
al_loss: 1.1665
Epoch 17/100
53/53 - 1s - 19ms/step - accuracy: 0.9264 - loss: 0.2327 - val_accuracy: 0.7169 - v
al_loss: 1.6911
Epoch 18/100
53/53 - 1s - 18ms/step - accuracy: 0.9302 - loss: 0.2283 - val_accuracy: 0.8091 - v
al_loss: 1.8411
Epoch 19/100
53/53 - 1s - 18ms/step - accuracy: 0.9296 - loss: 0.2207 - val_accuracy: 0.8188 - v
al_loss: 2.2193
Epoch 20/100
53/53 - 1s - 18ms/step - accuracy: 0.9309 - loss: 0.2199 - val_accuracy: 0.8146 - v
al_loss: 0.9659
```

```
Epoch 21/100
53/53 - 1s - 17ms/step - accuracy: 0.9324 - loss: 0.2282 - val_accuracy: 0.8653 - v
al_loss: 0.6439
Epoch 22/100
53/53 - 1s - 17ms/step - accuracy: 0.9362 - loss: 0.2049 - val_accuracy: 0.8335 - v
al_loss: 1.9098
Epoch 23/100
53/53 - 1s - 17ms/step - accuracy: 0.9355 - loss: 0.1975 - val_accuracy: 0.7477 - v
al_loss: 1.2485
Epoch 24/100
53/53 - 1s - 17ms/step - accuracy: 0.9351 - loss: 0.2066 - val_accuracy: 0.7740 - v
al_loss: 1.2998
Epoch 25/100
53/53 - 1s - 18ms/step - accuracy: 0.9369 - loss: 0.1997 - val_accuracy: 0.8193 - v
al_loss: 2.1584
Epoch 26/100
53/53 - 1s - 18ms/step - accuracy: 0.9385 - loss: 0.1902 - val_accuracy: 0.8261 - v
al_loss: 2.5789
Epoch 27/100
53/53 - 1s - 17ms/step - accuracy: 0.9378 - loss: 0.2042 - val_accuracy: 0.8422 - v
al_loss: 2.4481
Epoch 28/100
53/53 - 1s - 17ms/step - accuracy: 0.9378 - loss: 0.1914 - val_accuracy: 0.5893 - v
al_loss: 2.3342
Epoch 29/100
53/53 - 1s - 17ms/step - accuracy: 0.9371 - loss: 0.2203 - val_accuracy: 0.7154 - v
al_loss: 2.3624
Epoch 30/100
53/53 - 1s - 17ms/step - accuracy: 0.9331 - loss: 0.2196 - val_accuracy: 0.8168 - v
al_loss: 1.6621
Epoch 31/100
53/53 - 1s - 17ms/step - accuracy: 0.9396 - loss: 0.1878 - val_accuracy: 0.8082 - v
al_loss: 1.1757
Epoch 32/100
53/53 - 1s - 17ms/step - accuracy: 0.9445 - loss: 0.1790 - val_accuracy: 0.7562 - v
al_loss: 1.7958
Epoch 33/100
53/53 - 1s - 17ms/step - accuracy: 0.9419 - loss: 0.2688 - val_accuracy: 0.7807 - v
al_loss: 1.9176
Epoch 34/100
53/53 - 1s - 17ms/step - accuracy: 0.9414 - loss: 0.1817 - val_accuracy: 0.8057 - v
al_loss: 1.5486
Epoch 35/100
53/53 - 1s - 18ms/step - accuracy: 0.9437 - loss: 0.1793 - val_accuracy: 0.7629 - v
al_loss: 1.9957
Epoch 36/100
53/53 - 1s - 17ms/step - accuracy: 0.9451 - loss: 0.1741 - val_accuracy: 0.8285 - v
al_loss: 1.0106
Epoch 37/100
53/53 - 1s - 17ms/step - accuracy: 0.9432 - loss: 0.2140 - val_accuracy: 0.8146 - v
al_loss: 2.6621
Epoch 38/100
53/53 - 1s - 17ms/step - accuracy: 0.9456 - loss: 0.2281 - val_accuracy: 0.8366 - v
al_loss: 0.8054
Epoch 39/100
53/53 - 1s - 17ms/step - accuracy: 0.9457 - loss: 0.1705 - val_accuracy: 0.7910 - v
al_loss: 0.8396
Epoch 40/100
53/53 - 1s - 17ms/step - accuracy: 0.9467 - loss: 0.1704 - val_accuracy: 0.8475 - v
al_loss: 0.8366
```

```
Epoch 41/100
53/53 - 1s - 17ms/step - accuracy: 0.9474 - loss: 0.1633 - val_accuracy: 0.7242 - v
al_loss: 2.2953
Epoch 42/100
53/53 - 1s - 17ms/step - accuracy: 0.9460 - loss: 0.1725 - val_accuracy: 0.7427 - v
al_loss: 1.2563
Epoch 43/100
53/53 - 1s - 17ms/step - accuracy: 0.9461 - loss: 0.1663 - val_accuracy: 0.8158 - v
al_loss: 0.9405
Epoch 44/100
53/53 - 1s - 17ms/step - accuracy: 0.9467 - loss: 0.1674 - val_accuracy: 0.7342 - v
al_loss: 3.3773
Epoch 45/100
53/53 - 1s - 17ms/step - accuracy: 0.9456 - loss: 0.1765 - val_accuracy: 0.7772 - v
al_loss: 1.6103
Epoch 46/100
53/53 - 1s - 17ms/step - accuracy: 0.9470 - loss: 0.1778 - val_accuracy: 0.7166 - v
al_loss: 2.6957
Epoch 47/100
53/53 - 1s - 17ms/step - accuracy: 0.9469 - loss: 0.1860 - val_accuracy: 0.8156 - v
al_loss: 1.5897
Epoch 48/100
53/53 - 1s - 17ms/step - accuracy: 0.9479 - loss: 0.2089 - val_accuracy: 0.7808 - v
al_loss: 2.2465
Epoch 49/100
53/53 - 1s - 17ms/step - accuracy: 0.9496 - loss: 0.1740 - val_accuracy: 0.7609 - v
al_loss: 2.8075
Epoch 50/100
53/53 - 1s - 17ms/step - accuracy: 0.9512 - loss: 0.1646 - val_accuracy: 0.8070 - v
al_loss: 1.5386
Epoch 51/100
53/53 - 1s - 17ms/step - accuracy: 0.9489 - loss: 0.1691 - val_accuracy: 0.8264 - v
al_loss: 1.4755
Epoch 52/100
53/53 - 1s - 18ms/step - accuracy: 0.9495 - loss: 0.1581 - val_accuracy: 0.7251 - v
al_loss: 3.2647
Epoch 53/100
53/53 - 1s - 18ms/step - accuracy: 0.9452 - loss: 0.1706 - val_accuracy: 0.7778 - v
al_loss: 1.3177
Epoch 54/100
53/53 - 1s - 19ms/step - accuracy: 0.9495 - loss: 0.1646 - val_accuracy: 0.8355 - v
al_loss: 2.0945
Epoch 55/100
53/53 - 1s - 18ms/step - accuracy: 0.9482 - loss: 0.1573 - val_accuracy: 0.7980 - v
al_loss: 3.1059
Epoch 56/100
53/53 - 1s - 18ms/step - accuracy: 0.9479 - loss: 0.1697 - val_accuracy: 0.8048 - v
al_loss: 1.5775
Epoch 57/100
53/53 - 1s - 18ms/step - accuracy: 0.9500 - loss: 0.1601 - val_accuracy: 0.7931 - v
al_loss: 2.7859
Epoch 58/100
53/53 - 1s - 18ms/step - accuracy: 0.9490 - loss: 0.1594 - val_accuracy: 0.8209 - v
al_loss: 2.6276
Epoch 59/100
53/53 - 1s - 17ms/step - accuracy: 0.9523 - loss: 0.1560 - val_accuracy: 0.8091 - v
al_loss: 2.8051
Epoch 60/100
53/53 - 1s - 17ms/step - accuracy: 0.9516 - loss: 0.1579 - val_accuracy: 0.8275 - v
al_loss: 1.6447
```

```
Epoch 61/100
53/53 - 1s - 17ms/step - accuracy: 0.9502 - loss: 0.1556 - val_accuracy: 0.8320 - v
al_loss: 2.3769
Epoch 62/100
53/53 - 1s - 17ms/step - accuracy: 0.9500 - loss: 0.1567 - val_accuracy: 0.8273 - v
al_loss: 1.5128
Epoch 63/100
53/53 - 1s - 17ms/step - accuracy: 0.9509 - loss: 0.1583 - val_accuracy: 0.7783 - v
al_loss: 1.9446
Epoch 64/100
53/53 - 1s - 17ms/step - accuracy: 0.9506 - loss: 0.1599 - val_accuracy: 0.8335 - v
al_loss: 3.5861
Epoch 65/100
53/53 - 1s - 17ms/step - accuracy: 0.9532 - loss: 0.1551 - val_accuracy: 0.8135 - v
al_loss: 1.8153
Epoch 66/100
53/53 - 1s - 17ms/step - accuracy: 0.9540 - loss: 0.1529 - val_accuracy: 0.8115 - v
al_loss: 3.2809
Epoch 67/100
53/53 - 1s - 17ms/step - accuracy: 0.9514 - loss: 0.1568 - val_accuracy: 0.8221 - v
al_loss: 2.9591
Epoch 68/100
53/53 - 1s - 17ms/step - accuracy: 0.9530 - loss: 0.1518 - val_accuracy: 0.7779 - v
al_loss: 3.6053
Epoch 69/100
53/53 - 1s - 17ms/step - accuracy: 0.9546 - loss: 0.1547 - val_accuracy: 0.7740 - v
al_loss: 1.2669
Epoch 70/100
53/53 - 1s - 17ms/step - accuracy: 0.9526 - loss: 0.1507 - val_accuracy: 0.8062 - v
al_loss: 1.8083
Epoch 71/100
53/53 - 1s - 17ms/step - accuracy: 0.9565 - loss: 0.1991 - val_accuracy: 0.8218 - v
al_loss: 2.3014
Epoch 72/100
53/53 - 1s - 17ms/step - accuracy: 0.9595 - loss: 0.1345 - val_accuracy: 0.8217 - v
al_loss: 1.9222
Epoch 73/100
53/53 - 1s - 17ms/step - accuracy: 0.9606 - loss: 0.1372 - val_accuracy: 0.8300 - v
al_loss: 2.4971
Epoch 74/100
53/53 - 1s - 17ms/step - accuracy: 0.9604 - loss: 0.1312 - val_accuracy: 0.8067 - v
al_loss: 1.4510
Epoch 75/100
53/53 - 1s - 17ms/step - accuracy: 0.9563 - loss: 0.1622 - val_accuracy: 0.8437 - v
al_loss: 1.6137
Epoch 76/100
53/53 - 1s - 17ms/step - accuracy: 0.9605 - loss: 0.1256 - val_accuracy: 0.8279 - v
al_loss: 1.6985
Epoch 77/100
53/53 - 1s - 17ms/step - accuracy: 0.9601 - loss: 0.1331 - val_accuracy: 0.8152 - v
al_loss: 1.4452
Epoch 78/100
53/53 - 1s - 17ms/step - accuracy: 0.9624 - loss: 0.1237 - val_accuracy: 0.8089 - v
al_loss: 3.3419
Epoch 79/100
53/53 - 1s - 17ms/step - accuracy: 0.9599 - loss: 0.1374 - val_accuracy: 0.7536 - v
al_loss: 3.1276
Epoch 80/100
53/53 - 1s - 18ms/step - accuracy: 0.9590 - loss: 0.1360 - val_accuracy: 0.7790 - v
al_loss: 2.4934
```

```
Epoch 81/100
53/53 - 1s - 17ms/step - accuracy: 0.9626 - loss: 0.1227 - val_accuracy: 0.7924 - v
al_loss: 2.9617
Epoch 82/100
53/53 - 1s - 17ms/step - accuracy: 0.9600 - loss: 0.1235 - val_accuracy: 0.8358 - v
al_loss: 2.4517
Epoch 83/100
53/53 - 1s - 17ms/step - accuracy: 0.9624 - loss: 0.1315 - val_accuracy: 0.8430 - v
al_loss: 2.9275
Epoch 84/100
53/53 - 1s - 17ms/step - accuracy: 0.9608 - loss: 0.1331 - val_accuracy: 0.7874 - v
al_loss: 2.0645
Epoch 85/100
53/53 - 1s - 17ms/step - accuracy: 0.9613 - loss: 0.1228 - val_accuracy: 0.8449 - v
al_loss: 2.1637
Epoch 86/100
53/53 - 1s - 17ms/step - accuracy: 0.9632 - loss: 0.1246 - val_accuracy: 0.7445 - v
al_loss: 4.3289
Epoch 87/100
53/53 - 1s - 17ms/step - accuracy: 0.9637 - loss: 0.1251 - val_accuracy: 0.7915 - v
al_loss: 2.4555
Epoch 88/100
53/53 - 1s - 17ms/step - accuracy: 0.9639 - loss: 0.1180 - val_accuracy: 0.8226 - v
al_loss: 3.2196
Epoch 89/100
53/53 - 1s - 17ms/step - accuracy: 0.9600 - loss: 0.1239 - val_accuracy: 0.8059 - v
al_loss: 3.3013
Epoch 90/100
53/53 - 1s - 17ms/step - accuracy: 0.9632 - loss: 0.1216 - val_accuracy: 0.8262 - v
al_loss: 2.7275
Epoch 91/100
53/53 - 1s - 17ms/step - accuracy: 0.9658 - loss: 0.1155 - val_accuracy: 0.8363 - v
al_loss: 3.7873
Epoch 92/100
53/53 - 1s - 17ms/step - accuracy: 0.9614 - loss: 0.1360 - val_accuracy: 0.8311 - v
al_loss: 5.1126
Epoch 93/100
53/53 - 1s - 17ms/step - accuracy: 0.9637 - loss: 0.1230 - val_accuracy: 0.8202 - v
al_loss: 1.4325
Epoch 94/100
53/53 - 1s - 18ms/step - accuracy: 0.9624 - loss: 0.1213 - val_accuracy: 0.8337 - v
al_loss: 2.1697
Epoch 95/100
53/53 - 1s - 17ms/step - accuracy: 0.9646 - loss: 0.1131 - val_accuracy: 0.8344 - v
al_loss: 2.6447
Epoch 96/100
53/53 - 1s - 17ms/step - accuracy: 0.9626 - loss: 0.1201 - val_accuracy: 0.8047 - v
al_loss: 2.9758
Epoch 97/100
53/53 - 1s - 17ms/step - accuracy: 0.9610 - loss: 0.1245 - val_accuracy: 0.8138 - v
al_loss: 2.2470
Epoch 98/100
53/53 - 1s - 17ms/step - accuracy: 0.9614 - loss: 0.1257 - val_accuracy: 0.8433 - v
al_loss: 2.1849
Epoch 99/100
53/53 - 1s - 17ms/step - accuracy: 0.9634 - loss: 0.1195 - val_accuracy: 0.8240 - v
al_loss: 2.3880
Epoch 100/100
53/53 - 1s - 17ms/step - accuracy: 0.9624 - loss: 0.1244 - val_accuracy: 0.8086 - v
al_loss: 2.7342
```

```
653/653 - 1s - 939us/step - accuracy: 0.9674 - loss: 0.0966
206/206 - 0s - 978us/step - accuracy: 0.8086 - loss: 2.7342
```