## Second component of main path (≈3 lines)

X = Conv2D(filters = F2, kernel\_size = f, strides = (1,1), padding = 'same', kernel\_initializer = initializer(seed=0))(X)

X = BatchNormalization(axis = 3)(X, training = training) # Default axis

X = Activation('relu')(X)

## Third component of main path (≈2 lines)

X = Conv2D(filters = F3, kernel\_size = 1, strides = (1,1), padding = 'valid', kernel\_initializer = initializer(seed=0))(X)

X = BatchNormalization(axis = 3)(X, training = training) # Default axis

## Final step: Add shortcut value to main path, and pass it through a RELU activation (≈2 lines)

X = Add()([X, X\_shortcut])

X = Activation('relu')(X)

## Second component of main path (≈3 lines)

X = Conv2D(filters = F2, kernel\_size = f, strides = (1, 1), padding='same', kernel\_initializer = initializer(seed=0))(X)

X = BatchNormalization(axis = 3)(X, training=training)

X = Activation('relu')(X)

## Third component of main path (≈2 lines)

X = Conv2D(filters = F3, kernel\_size = 1, strides = (1, 1), padding='valid', kernel\_initializer = initializer(seed=0))(X)

X = BatchNormalization(axis = 3)(X, training=training)

##### SHORTCUT PATH ##### (≈2 lines)

X\_shortcut = Conv2D(filters = F3, kernel\_size = 1, strides = (s, s), padding='valid', kernel\_initializer = initializer(seed=0))(X\_shortcut)

X\_shortcut = BatchNormalization(axis = 3)(X\_shortcut, training=training)

## Stage 3 (≈4 lines)

X = convolutional\_block(X, f = 3, filters = [128, 128, 512], s = 2)

X = identity\_block(X, 3, [128, 128, 512])

X = identity\_block(X, 3, [128, 128, 512])

X = identity\_block(X, 3, [128, 128, 512])

## Stage 4 (≈6 lines)

X = convolutional\_block(X, f = 3, filters = [256, 256, 1024], s = 2)

X = identity\_block(X, 3, [256, 256, 1024])

X = identity\_block(X, 3, [256, 256, 1024])

X = identity\_block(X, 3, [256, 256, 1024])

X = identity\_block(X, 3, [256, 256, 1024])

X = identity\_block(X, 3, [256, 256, 1024])

## Stage 5 (≈3 lines)

X = convolutional\_block(X, f = 3, filters = [512, 512, 2048], s = 2)

X = identity\_block(X, 3, [512, 512, 2048])

X = identity\_block(X, 3, [512, 512, 2048])

## AVGPOOL (≈1 line). Use "X = AveragePooling2D(...)(X)"

X = AveragePooling2D(pool\_size=(2, 2))(X)