In [10]:

```
#%matplotlib notebook
import os
import sys
sys.path.append("numpy_path")
import numpy as np
import struct
from matplotlib import pyplot as plt
import keras
from keras.models import Sequential, load_model
from keras.layers import Dense, Dropout, Activation
from keras.optimizers import RMSprop
import keras.callbacks as cb
from keras.callbacks import EarlyStopping, ModelCheckpoint
from math import cos, sin, pi
import math
from statistics import mean
import os.path

shape_size = 48

# define loss history
class LossHistory(cb.Callback):
    def on_train_begin(self, logs={}):
        self.losses = []
    def on_batch_end(self, batch, logs={}):
        batch_loss = logs.get('loss')
        self.losses.append(batch_loss)

#plot losses
def plot_losses(losses):
    plt.plot(losses)
    plt.title('Loss per batch')
    plt.show()

def feature_scaling(X):
    X = X.T
    for i in range(7):
        mean = X[i].mean()
        std = X[i].std()
        X[i] = [(x - mean)/std for x in X[i]]
    return X.T
```

In [11]:
```
data_size = 0
dummy1 = [0]*200
dummy2 = [0]*6
SP = np.array(np.reshape(dummy1, (1, 200)))
SH = np.array(np.reshape(dummy2, (1, 6)))
for i in range(2, 65):
    path = 'meep_code/data/DATA'+str(i)
    if not os.path.exists(path):
        #miss.append(i)
        print('Missing batch:' + str(i))
        continue

    files = next(os.walk(path))[2] #dir is your directory path as string]
    num_data = len(files)
    data_size += num_data
    skip = []

    coordinates = np.genfromtxt('meep_code/data/DATA'+str(i)+'_sh.txt')
    xc, yc = coordinates[:, 0], coordinates[:, 1]
    xc = np.reshape(xc, (num_data, shape_size))
    yc = np.reshape(yc, (num_data, shape_size))

    for j in range(num_data):
        tmp = np.genfromtxt(path+'/'+'DATA'+str(i)+'_sp'+str(j)+'.txt')
        valid = True
        for q in range(200):
            if math.isnan(float(tmp[q])):
                print('Batch '+str(i)+'\tsample '+str(j)+' has NAN value')
                valid = False
                break
            if tmp[q] > 3:
                print('Batch '+str(i)+'\tsample '+str(j)+' has extreme value'
                valid = False
                break
        if not valid:
            #skip.append(j)
            continue
        SP = np.concatenate((SP, np.reshape(tmp, (1, 200))))
        tmp = []
        for q in range(6):
            tmp.append(math.sqrt(xc[j][q]**2 + yc[j][q]**2))
        SH = np.concatenate((SH, np.reshape(np.array(tmp), (1, 6))))
    print('Batch '+str(i)+' has \t'+str(num_data))
```

```
Batch 44 has    106
Batch 45 has    35
Batch 46 has    100
Batch 47 has    100
Batch 48 has    287
Batch 49 has    13
Batch 50 has    37
Batch 51 has    37
Batch 52 has    106
Batch 53 has    35
Batch 54 has    100
Batch 55 has    100
Batch 56 has    287
Batch 57 has    35
Batch 58 has    100
Batch 59 has    100
Batch 60 has    287
Batch 61 has    95
Batch 62 has    272
Batch 63 has    272
```

In [12]:

```
print('Total # of data: ' + str(len(SP)))
x = np.genfromtxt('meep_code/data/SP_xaxis.txt')
SP_F, SH_F = np.reshape(SP[1], (1, 200)),np.reshape(SH[1], (1, 6))
for i in range(2, len(SP)):
    peak = 0
    for j in range(1, 200):
        if SP[i][j - 1] >= 0.6 >=SP[i][j]:
            peak += 1
    if peak == 1:
        SP_F = np.concatenate((SP_F, np.reshape(SP[i], (1, 200))))
        SH_F = np.concatenate((SH_F, np.reshape(SH[i], (1, 6))))
```

```
Total # of data: 4989
```

In [13]:

```
DATA = np.append(SP_F, SH_F, axis = 1)
np.random.shuffle(DATA)

Y = DATA[:, :200]
X = DATA[:,200:]

train_size = int(len(DATA) * 0.8)

train_X = X[0:train_size, :]
train_Y = Y[0:train_size, :]
test_X = X[train_size:, :]
test_Y = Y[train_size:, :]
```

In [14]:

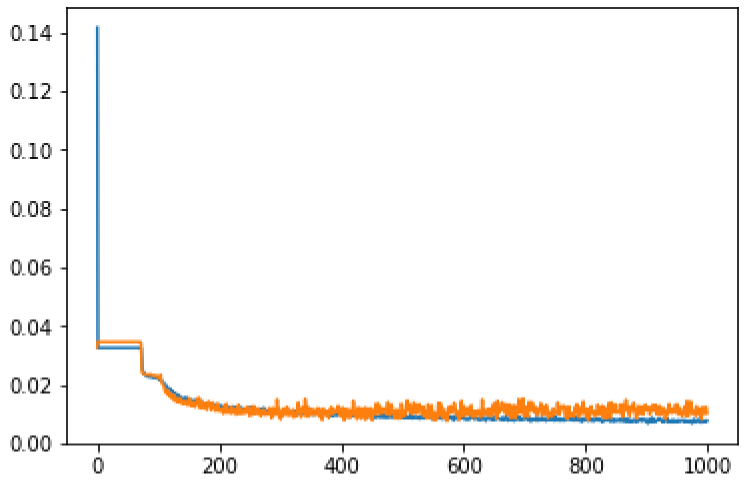
```
in_dim = 6
out_dim = 200
simulator = Sequential()
simulator.add(Dense(20, activation='relu', input_dim=in_dim))
simulator.add(Dropout(0.2))
simulator.add(Dense(500, activation='relu'))
simulator.add(Dropout(0.5))
simulator.add(Dense(500, activation='relu'))
simulator.add(Dropout(0.5))
# simulator.add(Dense(200, activation='relu'))
# simulator.add(Dropout(0.5))
simulator.add(Dense(200, activation='relu'))
simulator.add(Dropout(0.5))
simulator.add(Dense(200, activation='relu'))
simulator.add(Dropout(0.2))
simulator.add(Dense(out_dim, activation='sigmoid'))
simulator.compile(loss=keras.losses.mean_squared_error,
                  optimizer=keras.optimizers.Adam(lr = 0.001))
```

In [15]:

```
simulator.summary()
```

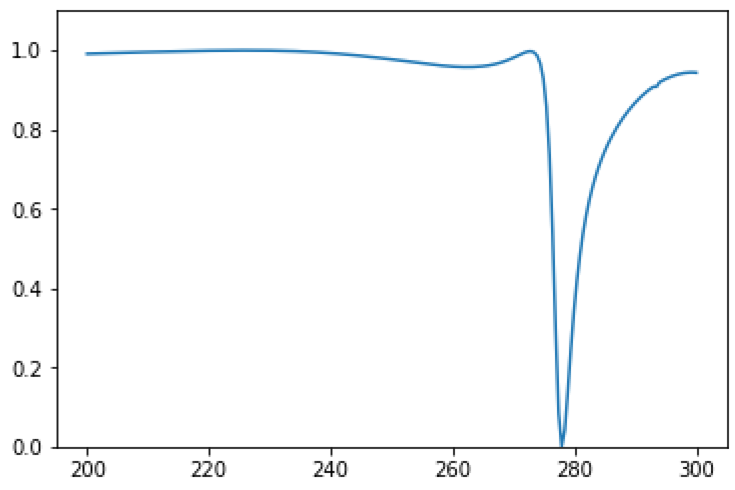
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
dense_19 (Dense)             (None, 20)                140       
_________________________________________________________________
dropout_16 (Dropout)         (None, 20)                0         
_________________________________________________________________
dense_20 (Dense)             (None, 500)               10500     
_________________________________________________________________
dropout_17 (Dropout)         (None, 500)               0         
_________________________________________________________________
dense_21 (Dense)             (None, 500)               250500    
_________________________________________________________________
dropout_18 (Dropout)         (None, 500)               0         
_________________________________________________________________
dense_22 (Dense)             (None, 200)               100200    
_________________________________________________________________
dropout_19 (Dropout)         (None, 200)               0         
_________________________________________________________________
dense_23 (Dense)             (None, 200)               40200     
_________________________________________________________________
dropout_20 (Dropout)         (None, 200)               0         
_________________________________________________________________
dense_24 (Dense)             (None, 200)               40200     
=================================================================
Total params: 441,740
Trainable params: 441,740
Non-trainable params: 0
_________________________________________________________________
```

In [16]:
```python
history = simulator.fit(train_X, train_Y,
                    epochs=1000,
                    batch_size=20,
                    validation_data=(test_X, test_Y),
                    verbose=2)

train_score = simulator.evaluate(train_X, train_Y, batch_size=20)
test_score = simulator.evaluate(test_X, test_Y, batch_size= 50)
print(train_score)
print(test_score)
plt.plot(history.history['loss'], label='train')
plt.plot(history.history['val_loss'], label='test')
```

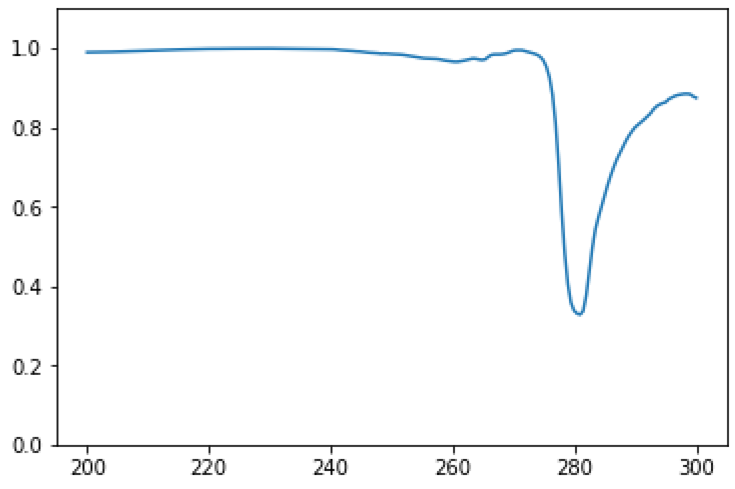


In [17]:
```python
x = np.genfromtxt('meep_code/data/SP_xaxis.txt')
for i in range(len(test_X)):
    print('Test '+str(i))
    print('True spectrum: ')
    plt.ylim(0, 1.1)
    plt.plot(x, test_Y[i])
    plt.show()
    print('Predicted spectrum: ')
    plt.ylim(0, 1.1)
    plt.plot(x, np.reshape(simulator.predict(np.reshape(test_X[i], (1, 6))),
    plt.show()
```

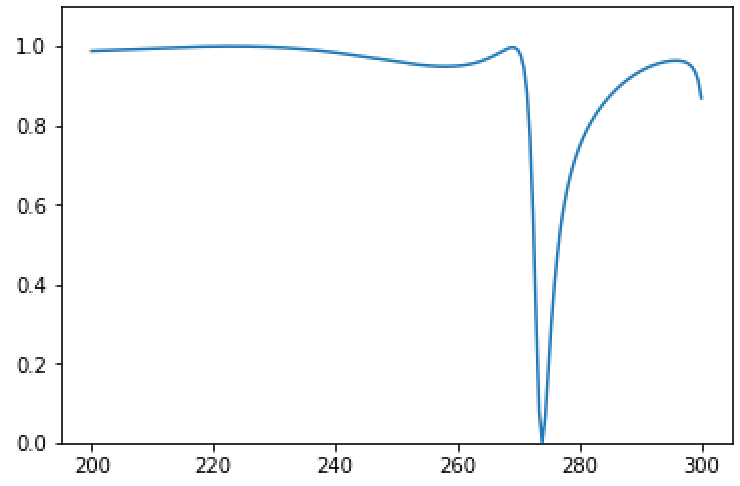
```
Test 0
True spectrum:
```



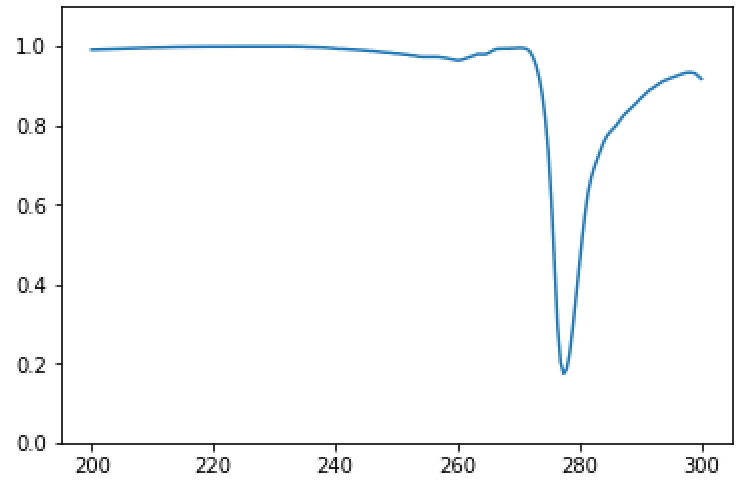
```
Predicted spectrum:
```



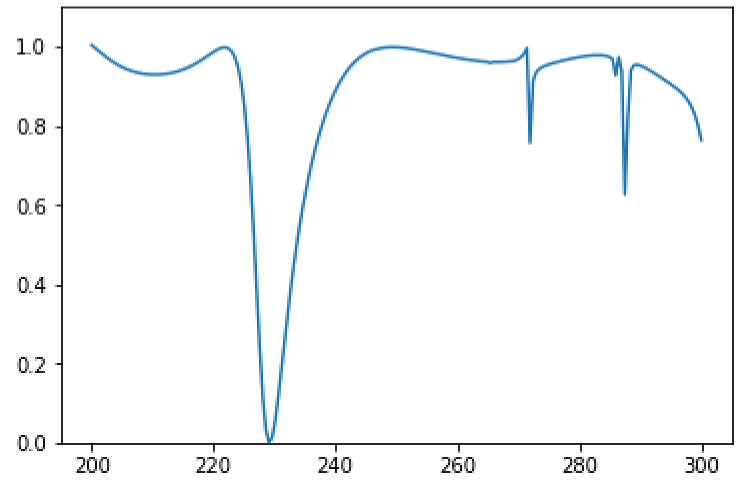
```
Test 1
True spectrum:
```



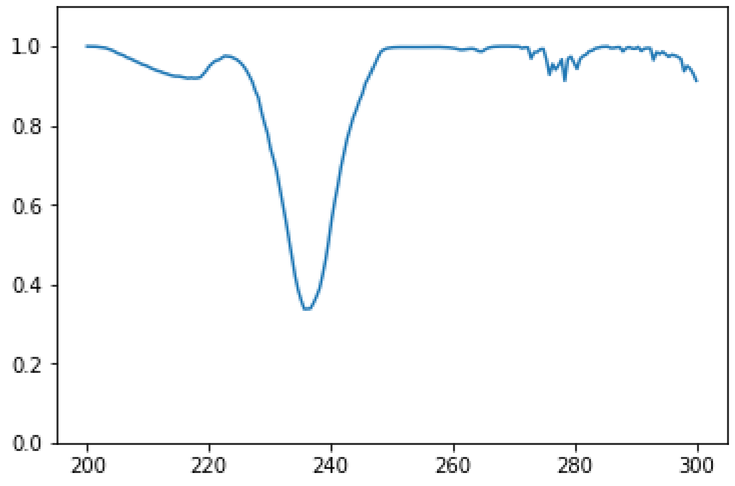
```
Predicted spectrum:
```



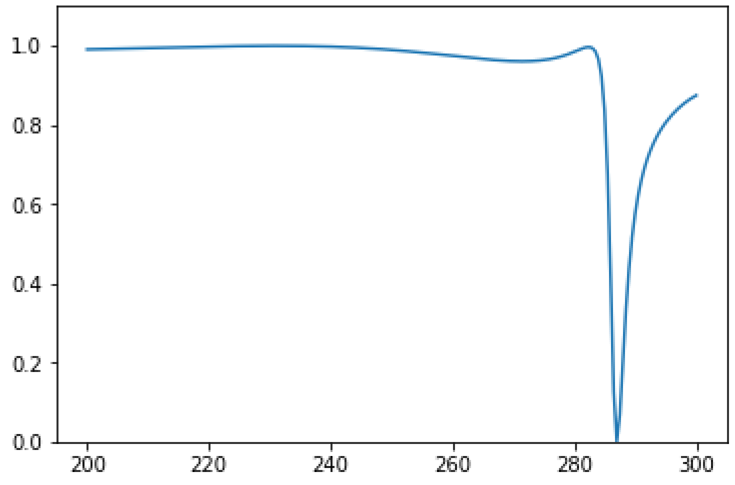
```
Test 2
True spectrum:
```



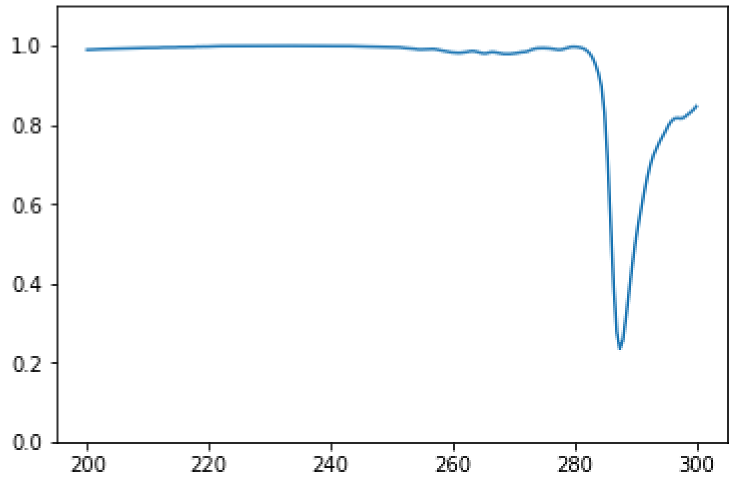
```
Predicted spectrum:
```



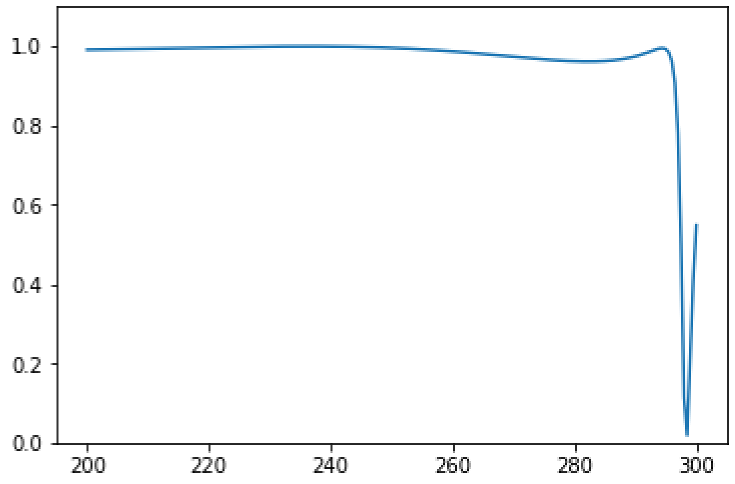
```
Test 3
True spectrum:
```



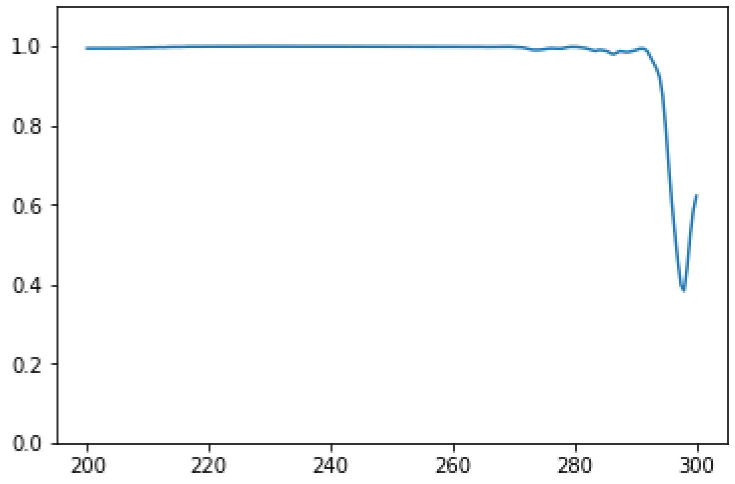
```
Predicted spectrum:
```



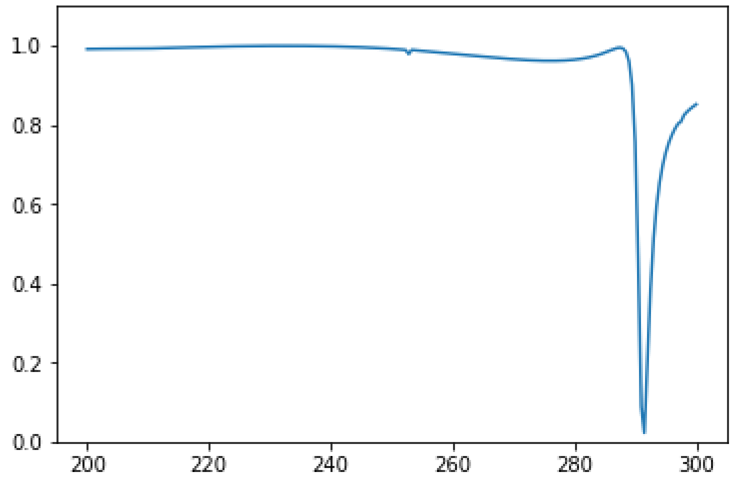
```
Test 4
True spectrum:
```



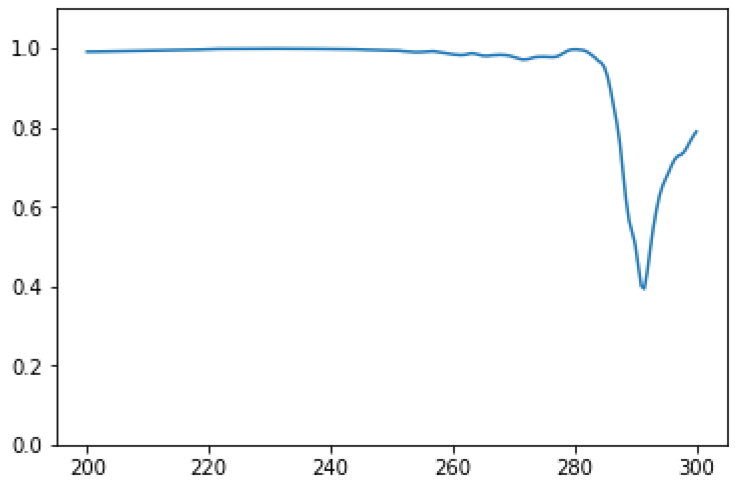
```
Predicted spectrum:
```



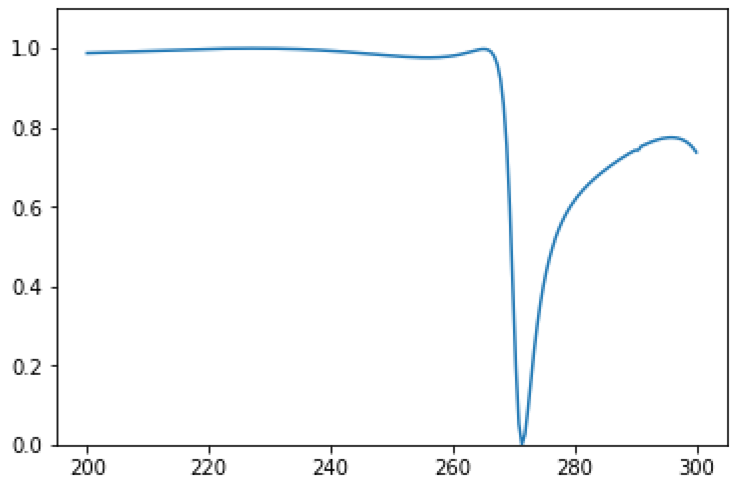
```
Test 5
True spectrum:
```



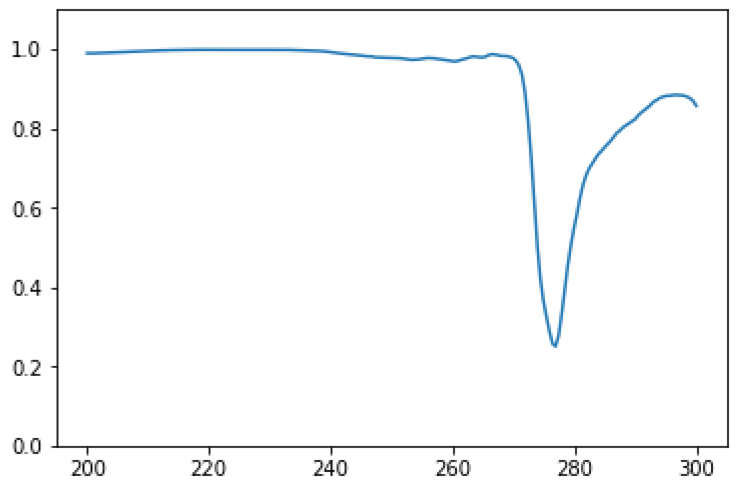
```
Predicted spectrum:
```



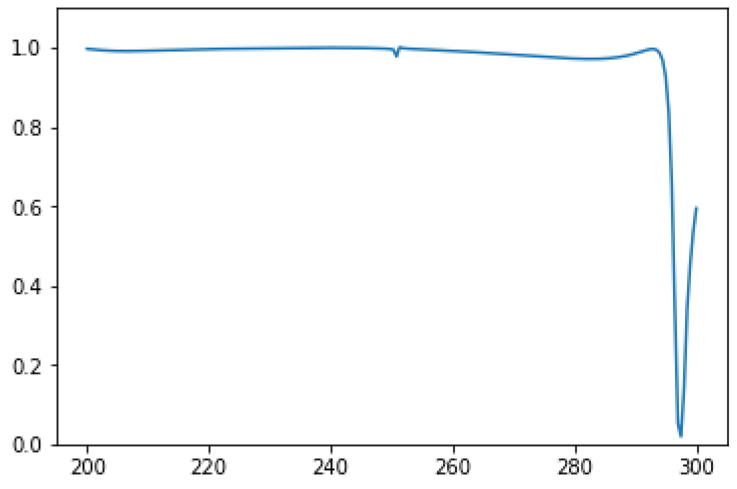
```
Test 6
True spectrum:
```



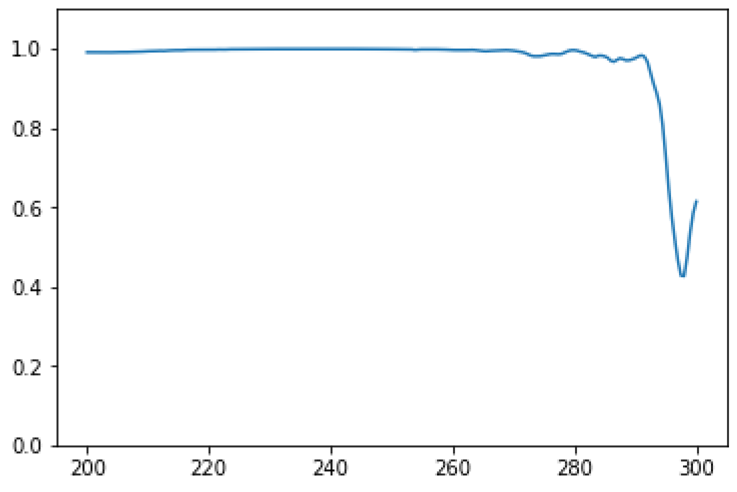
```
Predicted spectrum:
```



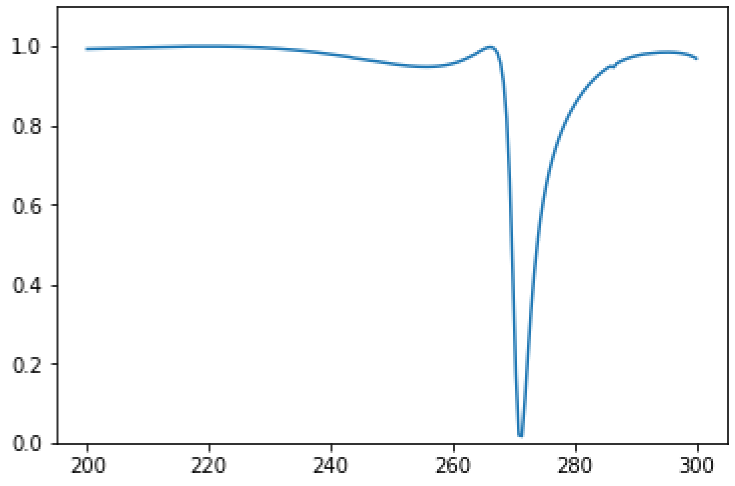
```
Test 7
True spectrum:
```



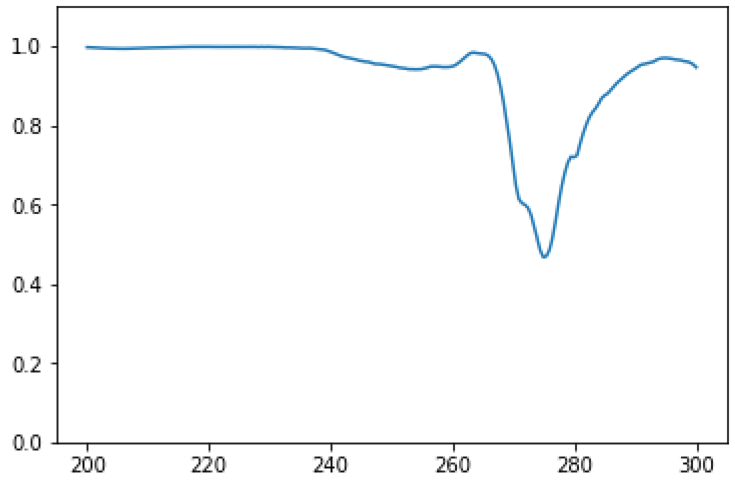
```
Predicted spectrum:
```



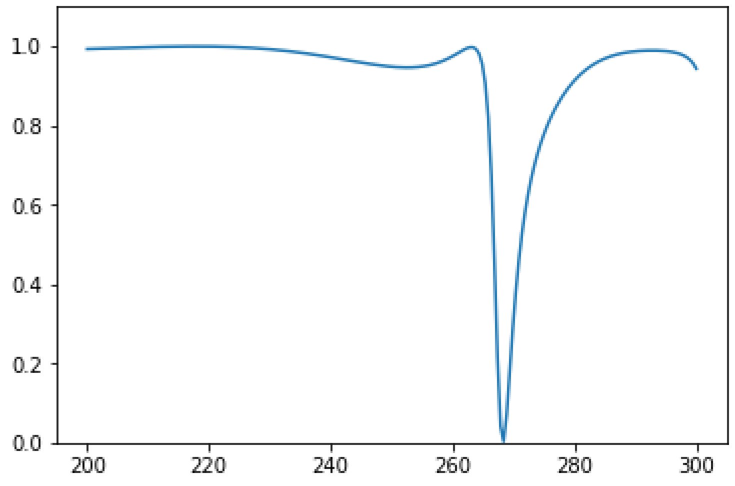
```
Test 8
True spectrum:
```



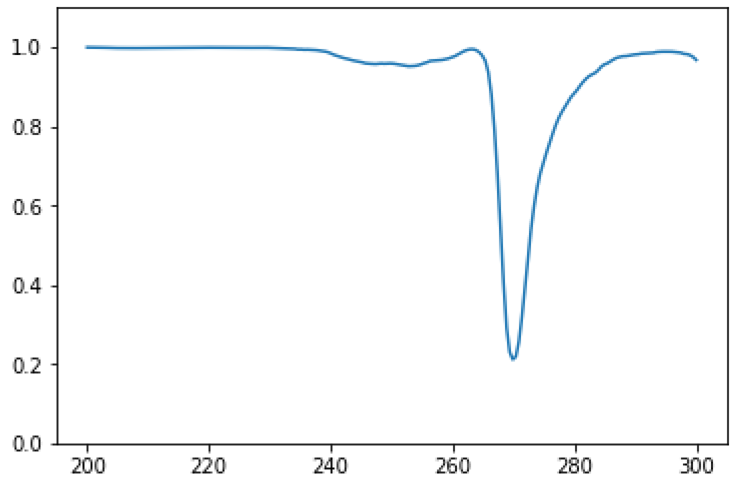
```
Predicted spectrum:
```



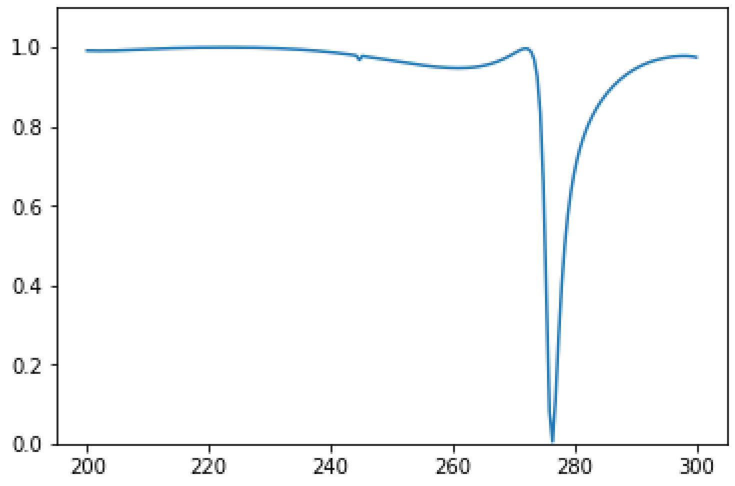
```
Test 9
True spectrum:
```



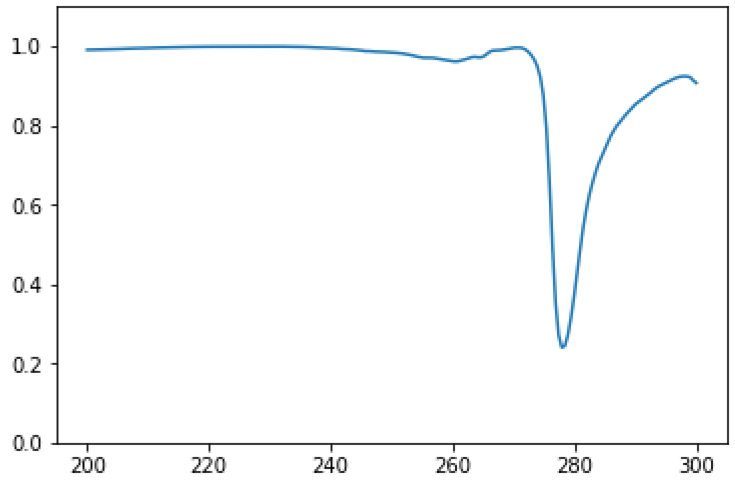
```
Predicted spectrum:
```



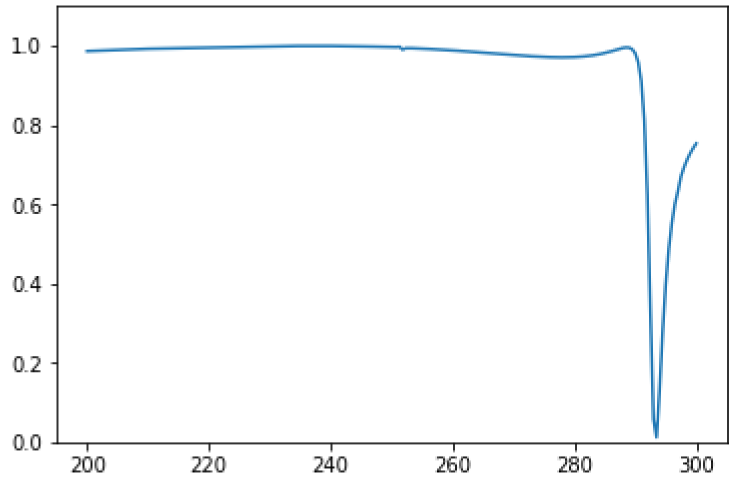
```
Test 10
True spectrum:
```



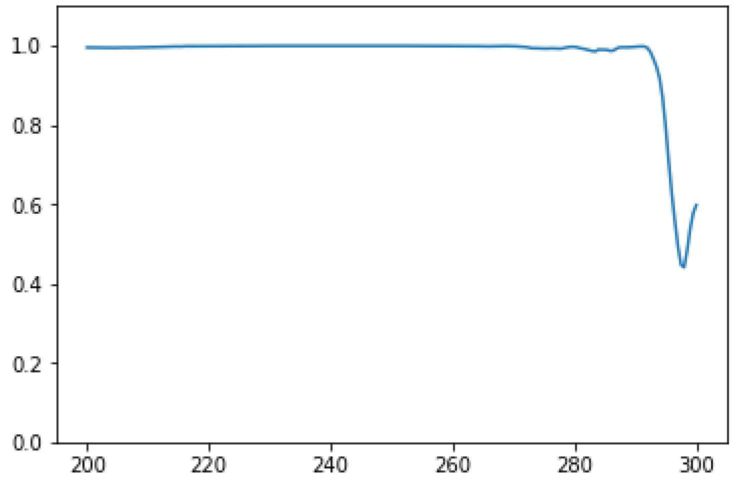
```
Predicted spectrum:
```



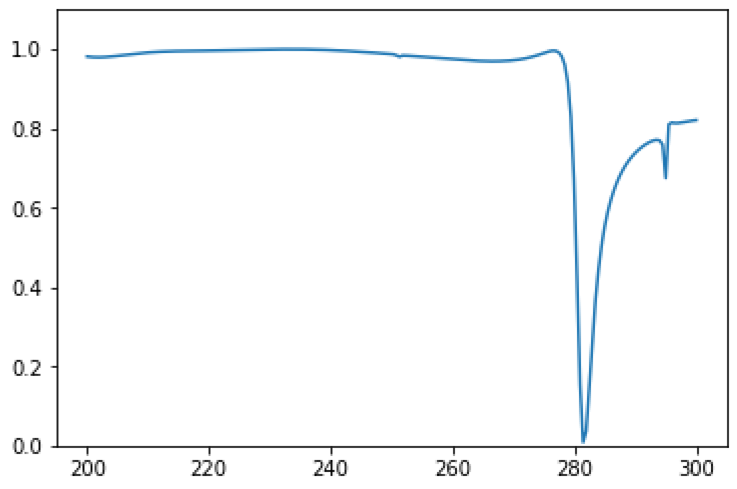
```
Test 11
True spectrum:
```



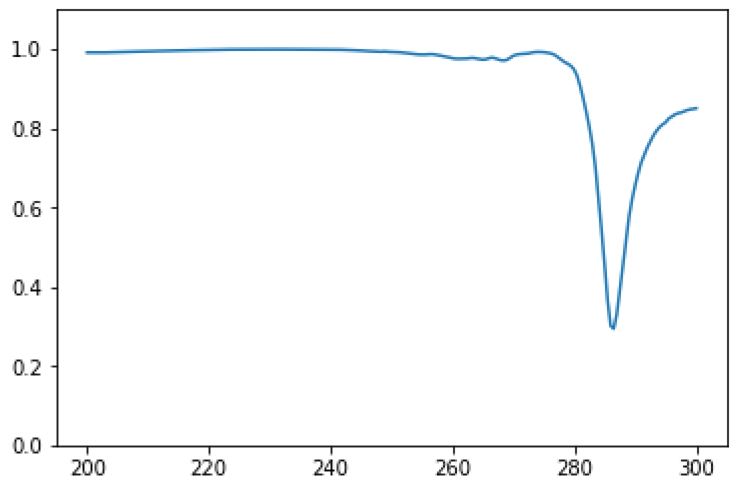
```
Predicted spectrum:
```



```
Test 12
True spectrum:
```

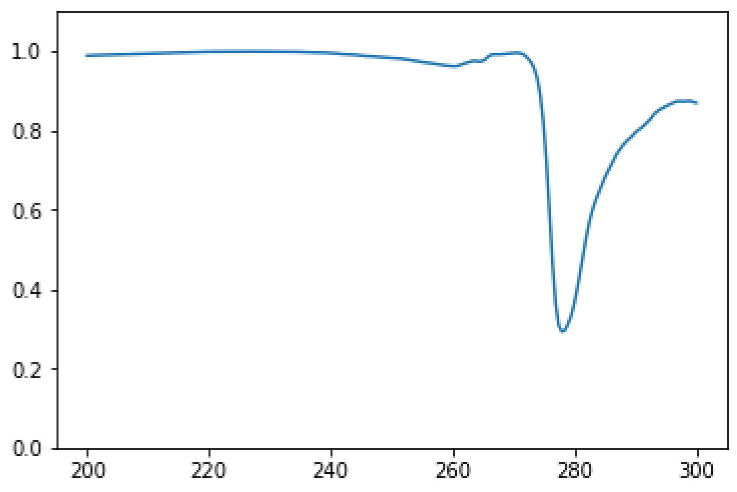


```
Predicted spectrum:
```

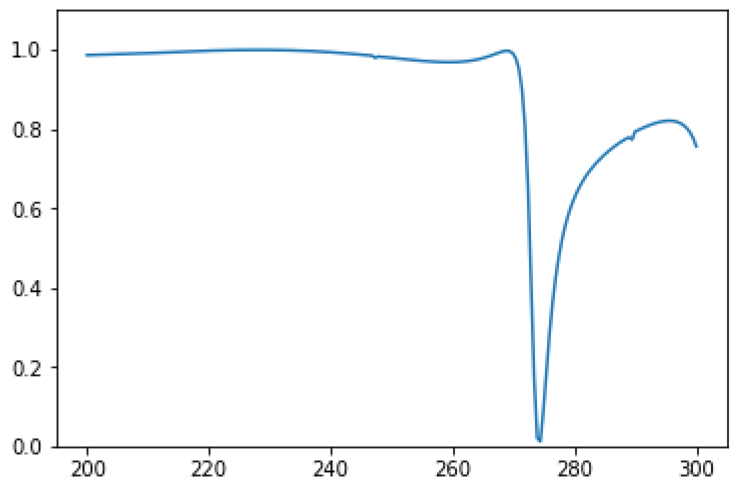


```
Test 13
True spectrum:
```

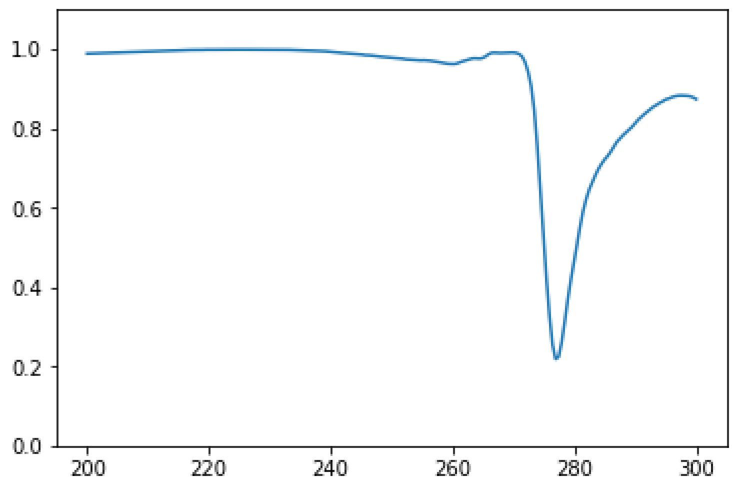
```
Predicted spectrum:
```



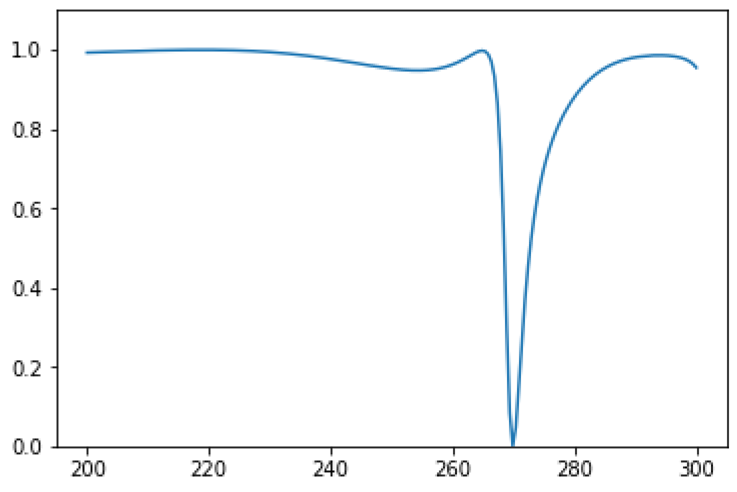
```
Test 14
True spectrum:
```



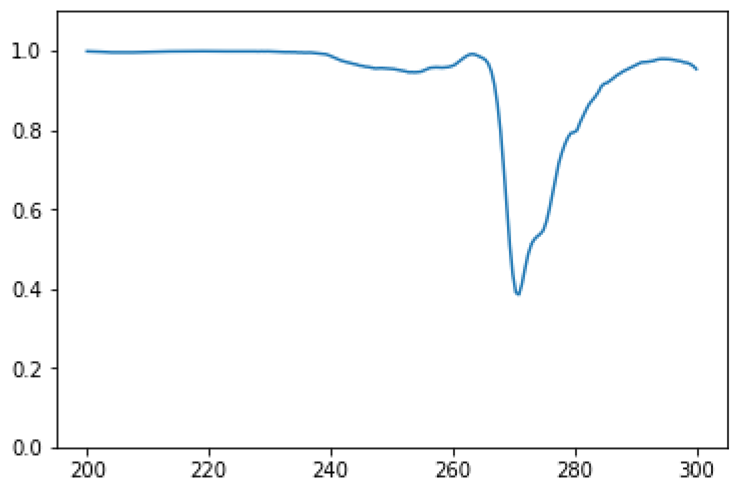
```
Predicted spectrum:
```



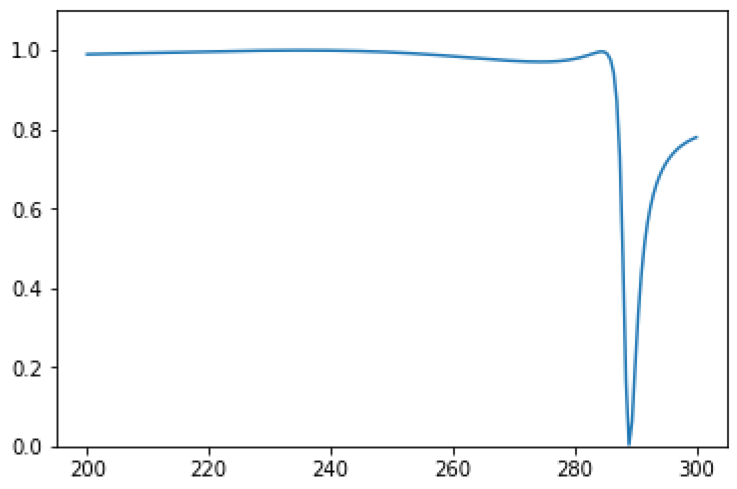
```
Test 15
True spectrum:
```



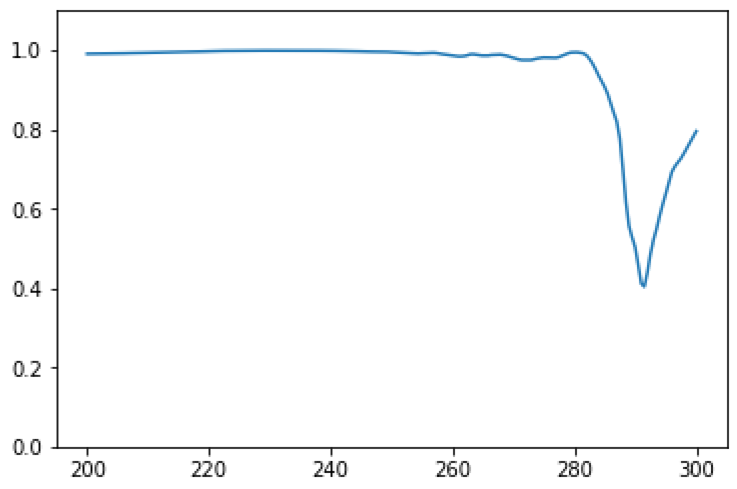
```
Predicted spectrum:
```



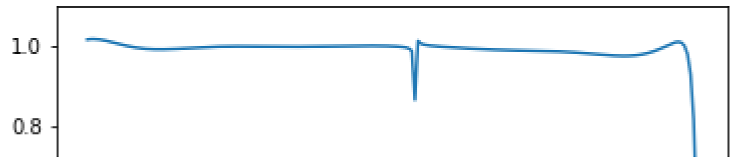
```
Test 16
True spectrum:
```



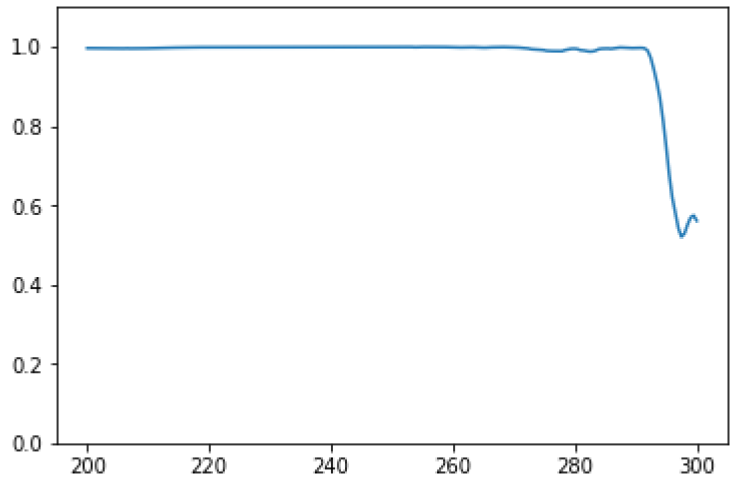
```
Predicted spectrum:
```



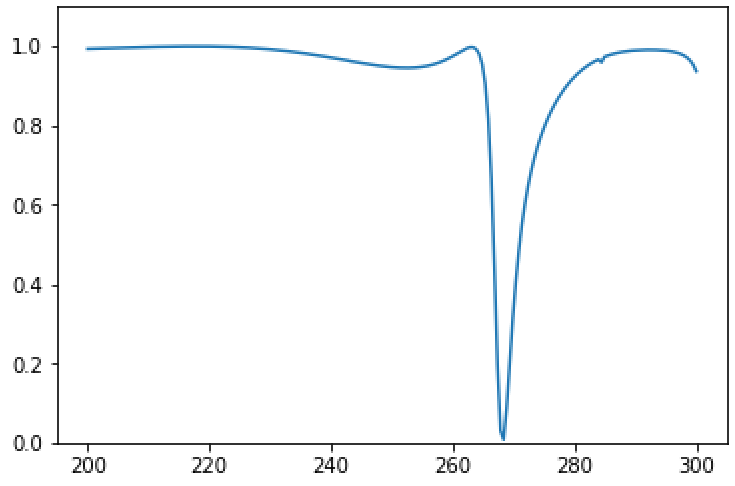
```
Test 17
True spectrum:
```



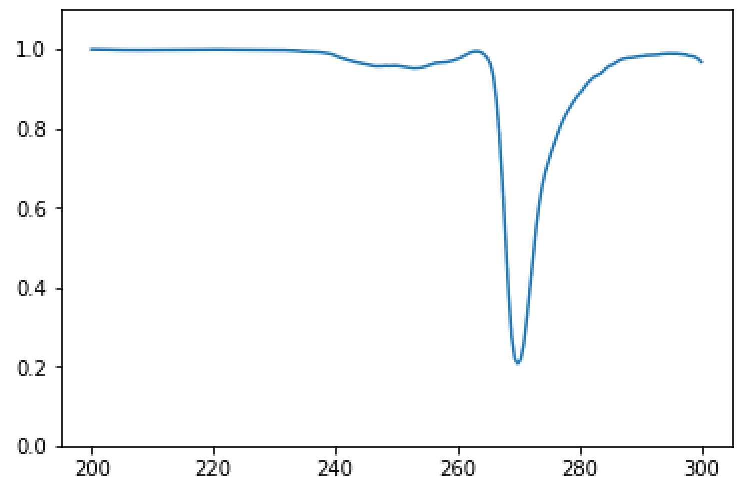
```
Predicted spectrum:
```



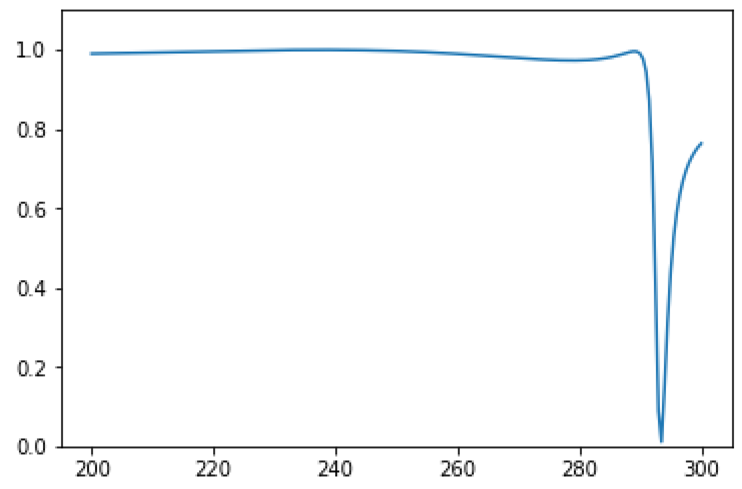
```
Test 18
True spectrum:
```



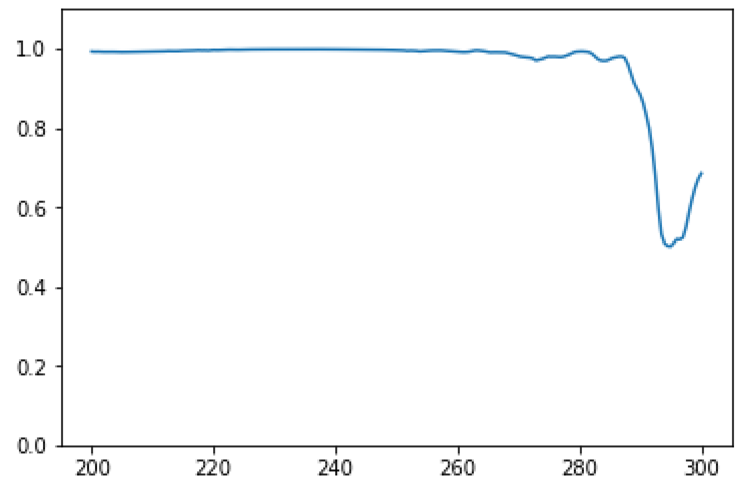
```
Predicted spectrum:
```



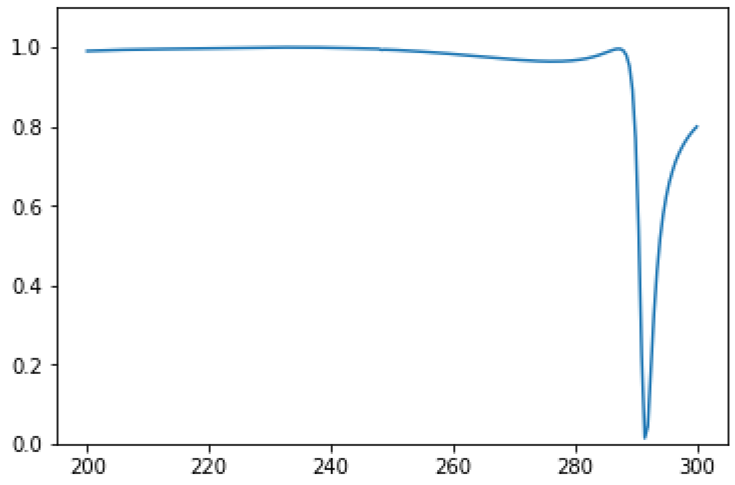
```
Test 19
True spectrum:
```



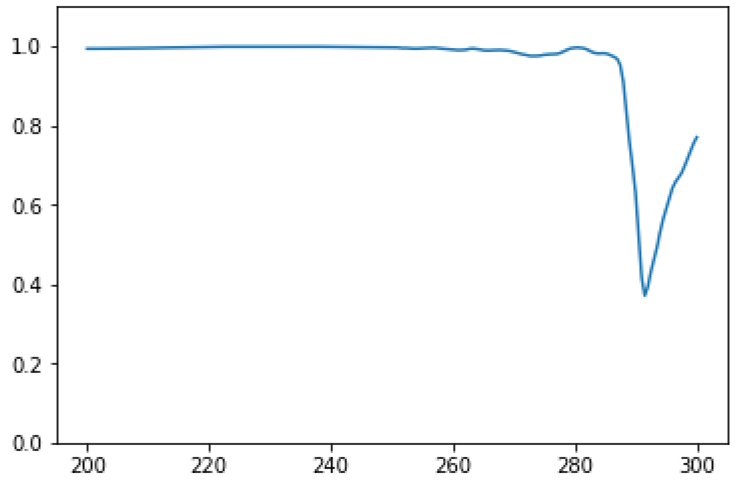
```
Predicted spectrum:
```



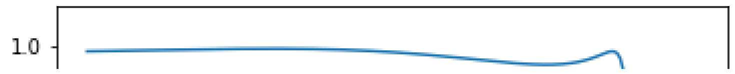
```
Test 20
True spectrum:
```



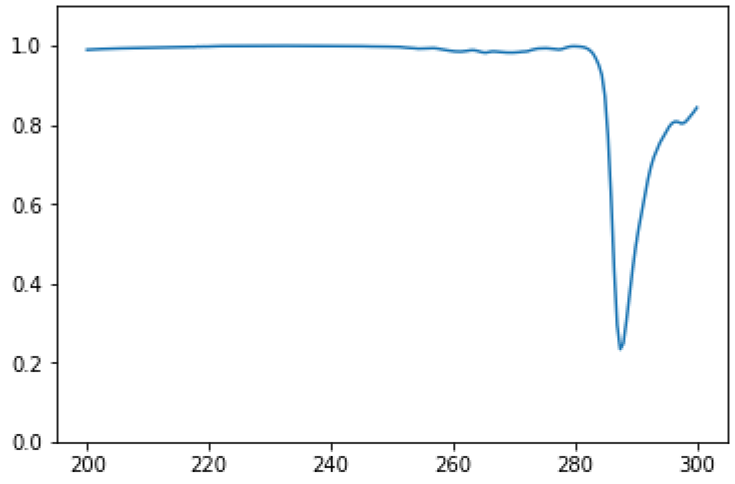
```
Predicted spectrum:
```



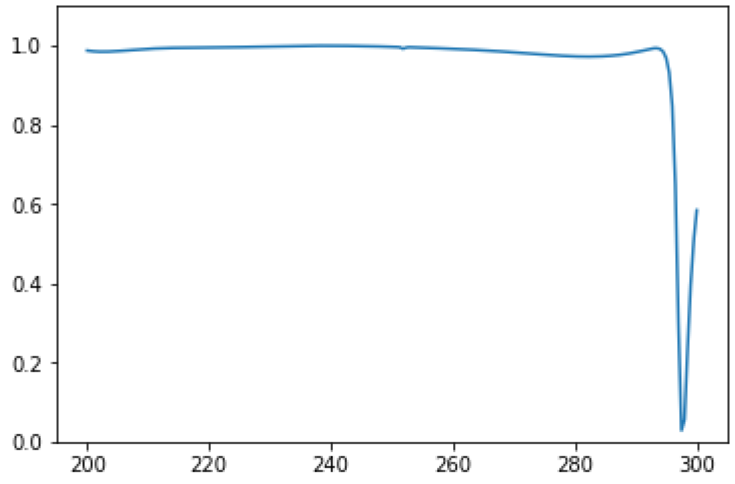
```
Test 21
True spectrum:
```



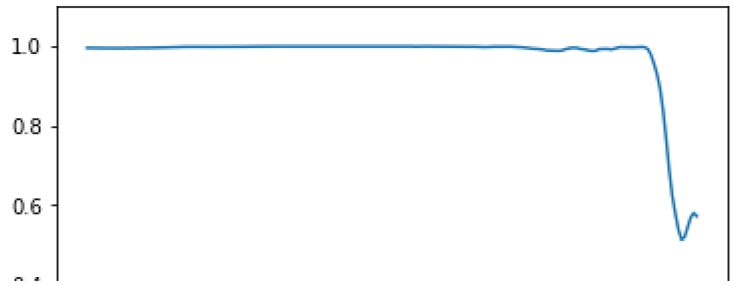
```
Predicted spectrum:
```



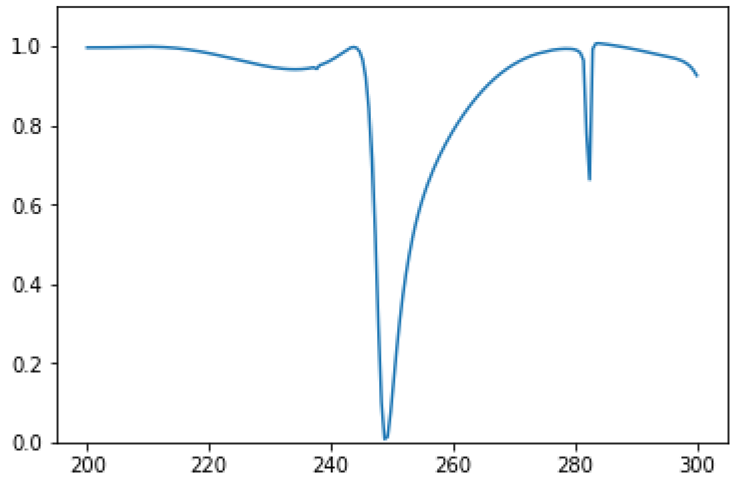
```
Test 22
True spectrum:
```



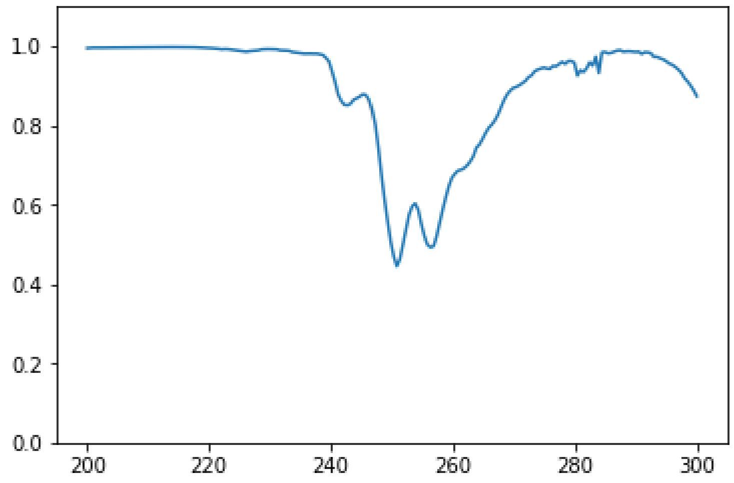
```
Predicted spectrum:
```



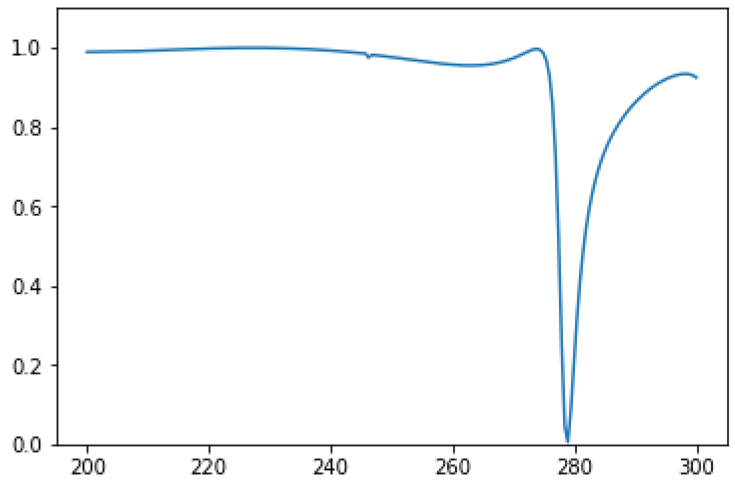
```
Test 23
True spectrum:
```



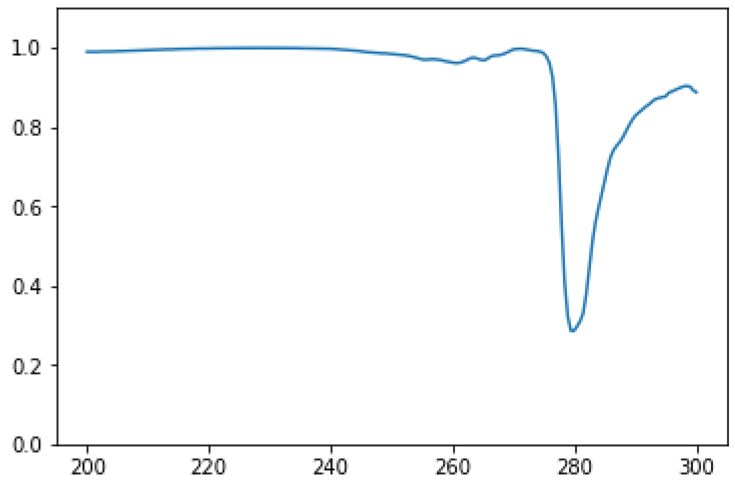
```
Predicted spectrum:
```



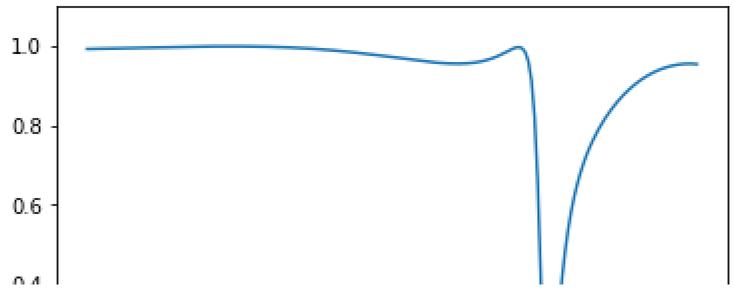
```
Test 24
True spectrum:
```



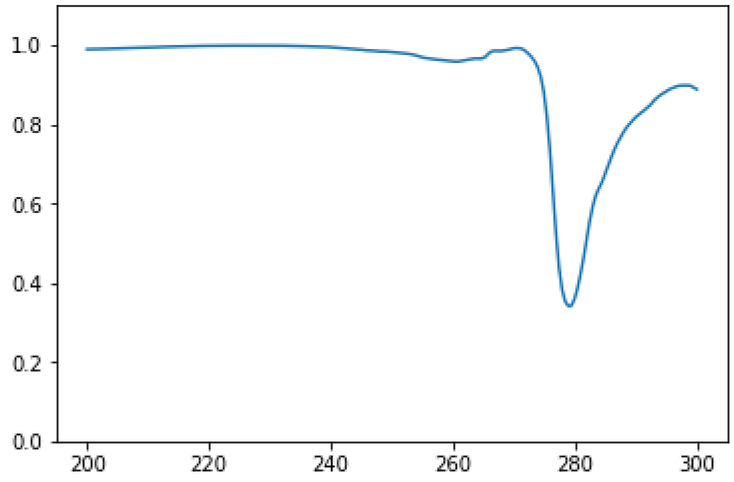
```
Predicted spectrum:
```



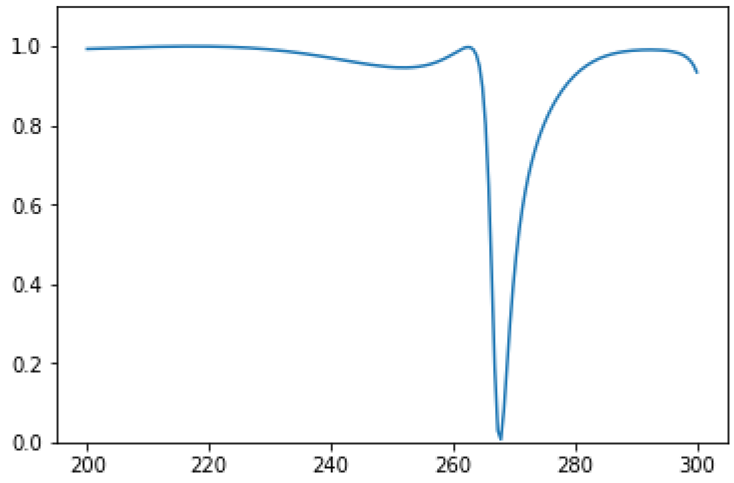
```
Test 25
True spectrum:
```



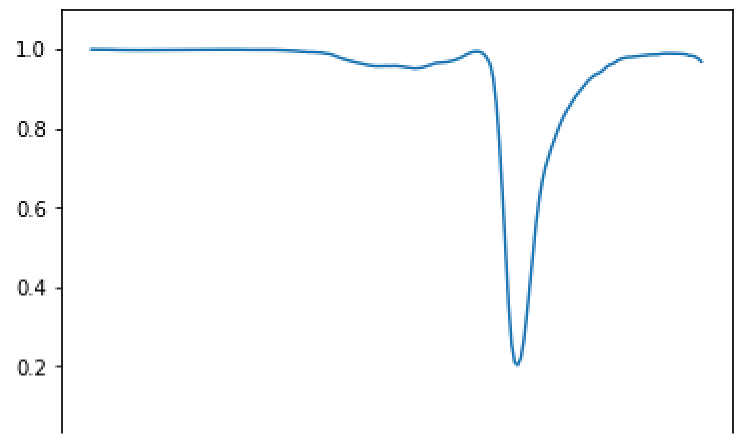
```
Predicted spectrum:
```



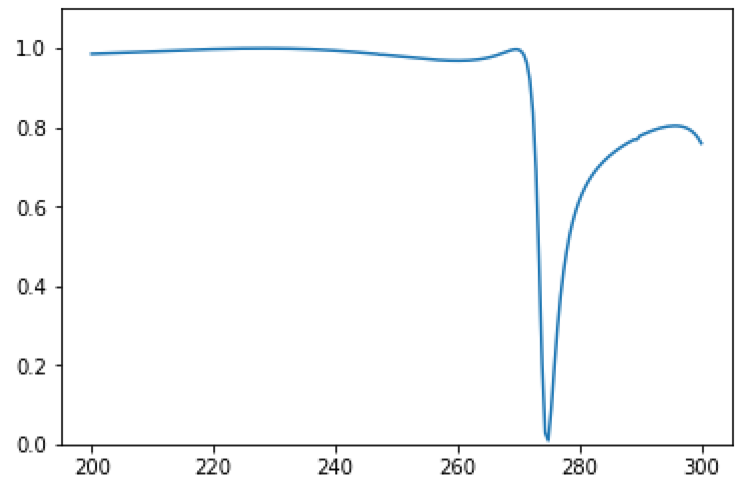
```
Test 26
True spectrum:
```



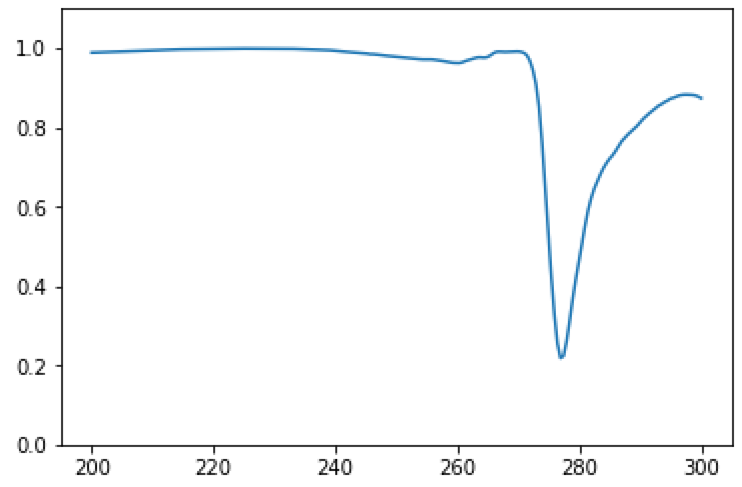
```
Predicted spectrum:
```



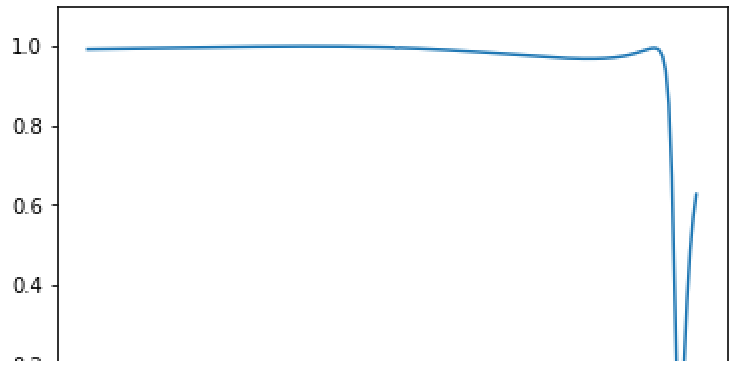
```
Test 27
True spectrum:
```



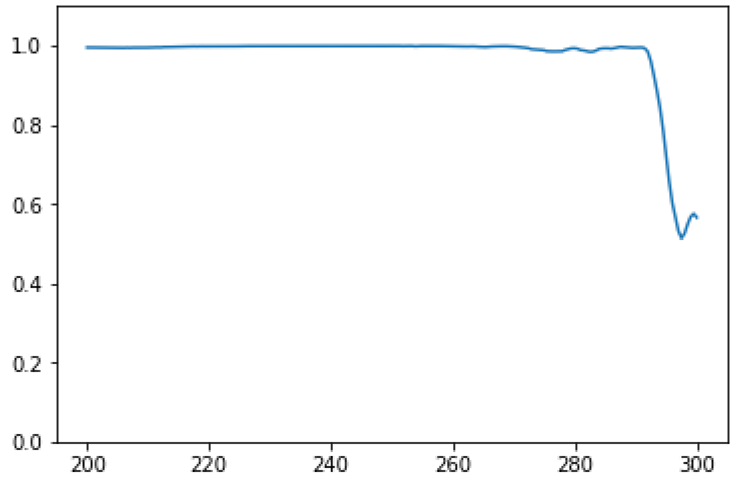
```
Predicted spectrum:
```



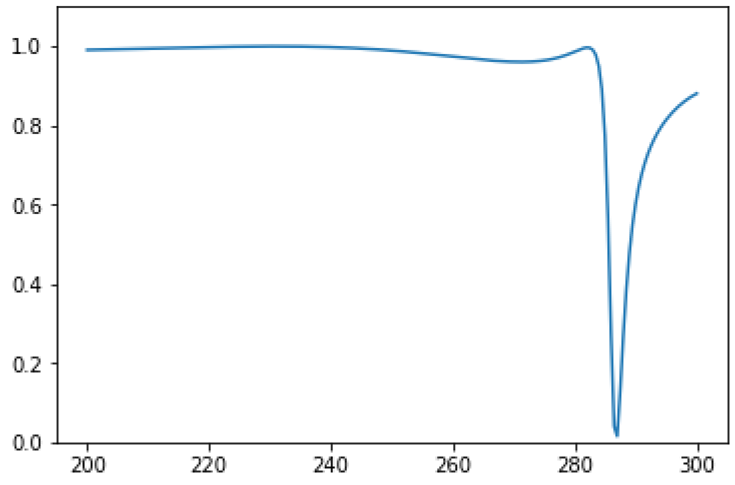
```
Test 28
True spectrum:
```



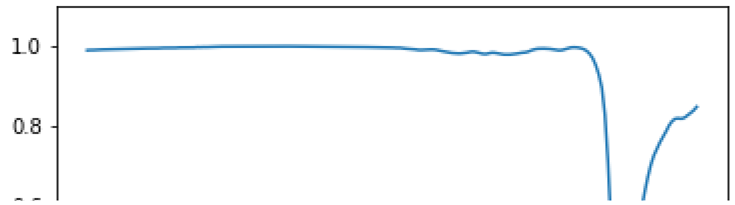
```
Predicted spectrum:
```



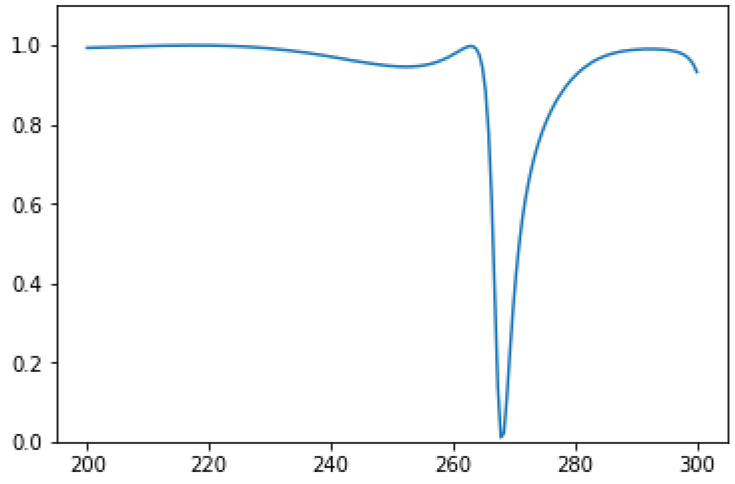
```
Test 29
True spectrum:
```



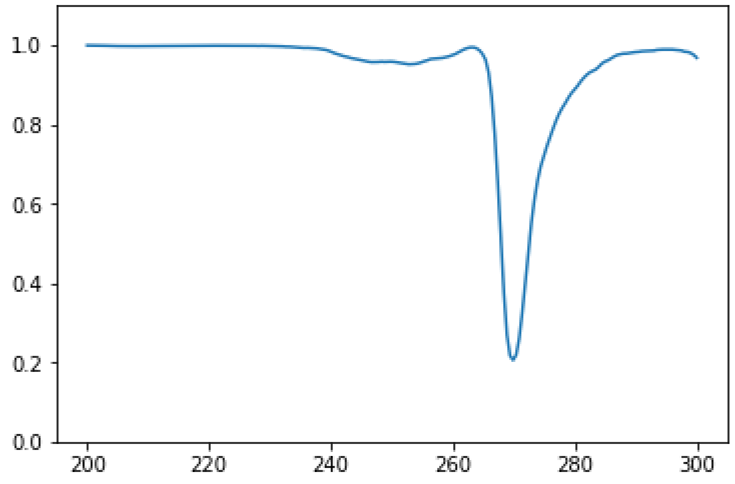
```
Predicted spectrum:
```



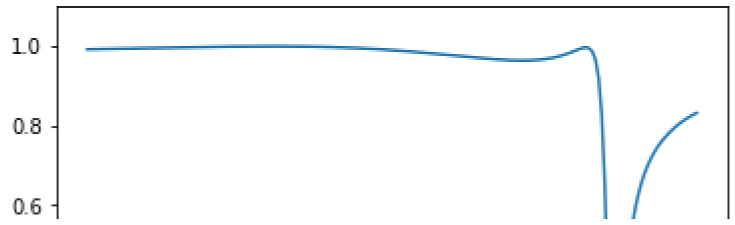
```
Test 30
True spectrum:
```



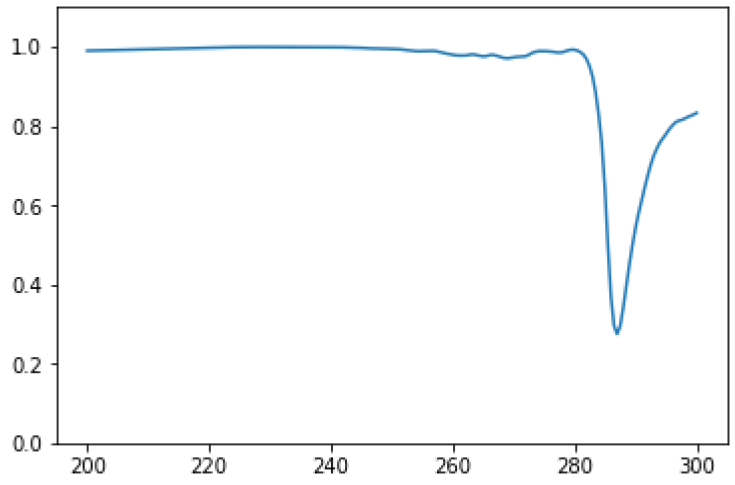
```
Predicted spectrum:
```



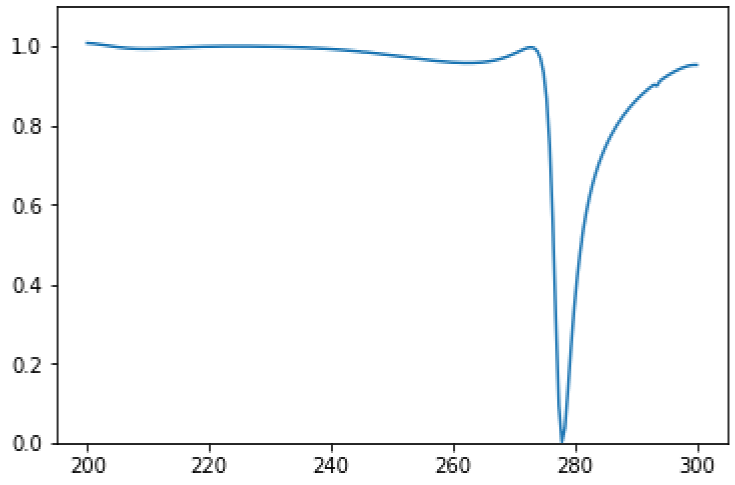
```
Test 31
True spectrum:
```



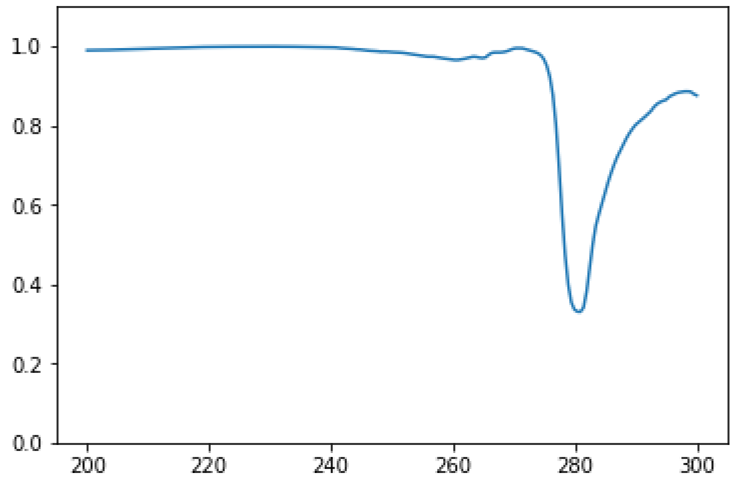
```
Predicted spectrum:
```



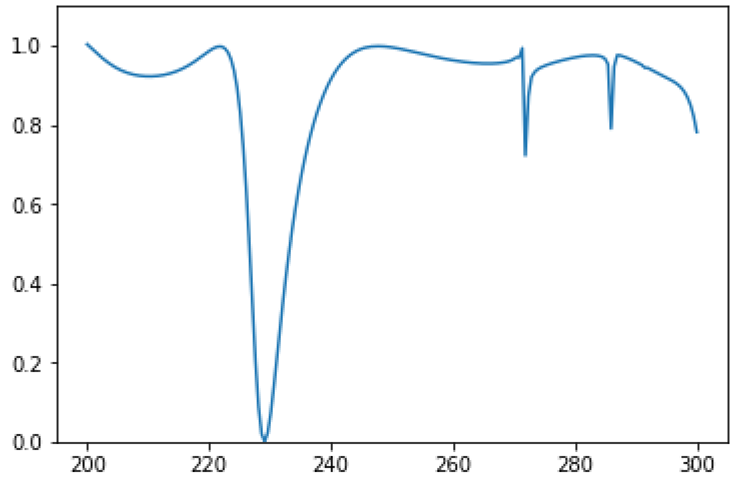
```
Test 32
True spectrum:
```



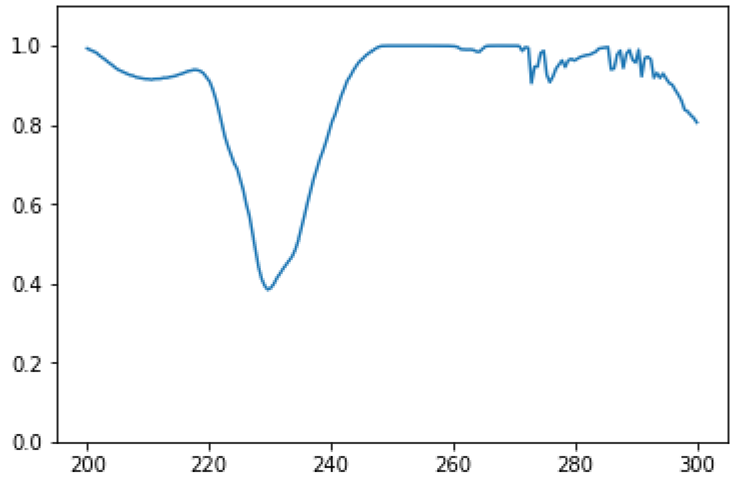
```
Predicted spectrum:
```



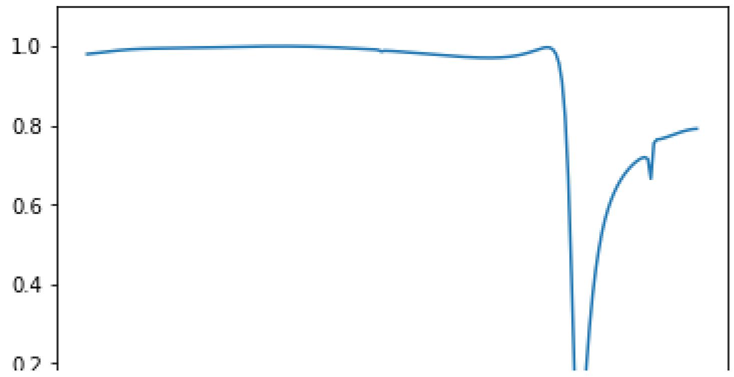
```
Test 33
True spectrum:
```



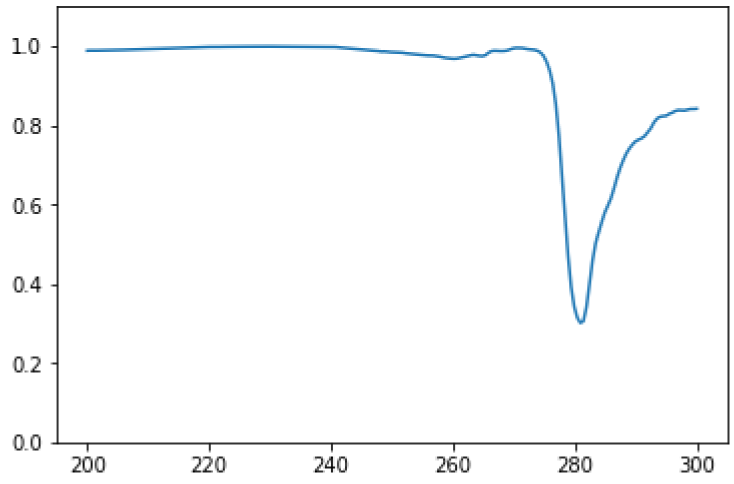
```
Predicted spectrum:
```



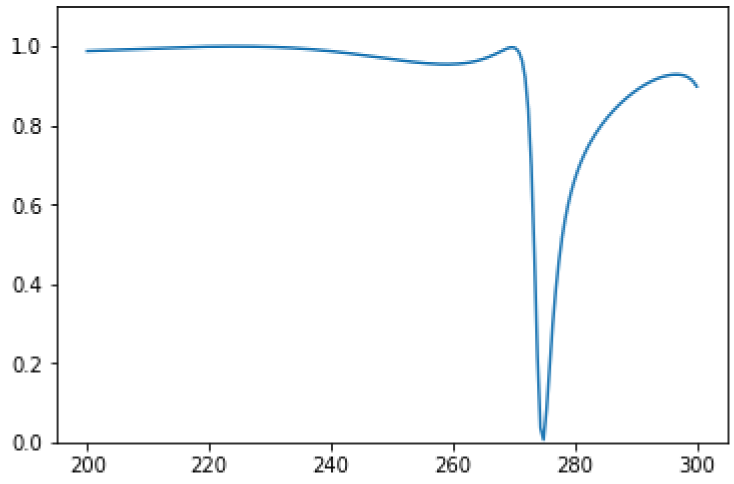
```
Test 34
True spectrum:
```



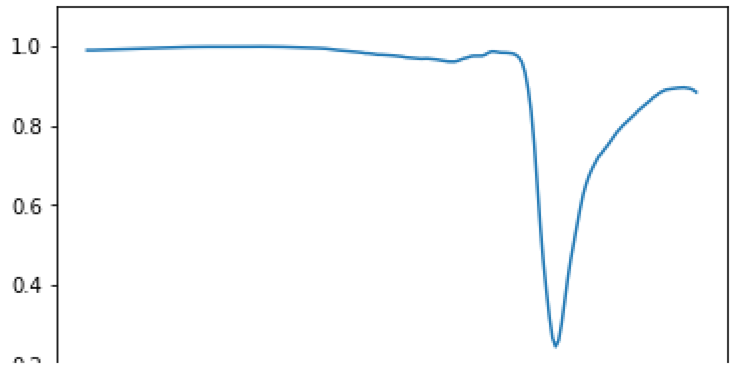
```
Predicted spectrum:
```



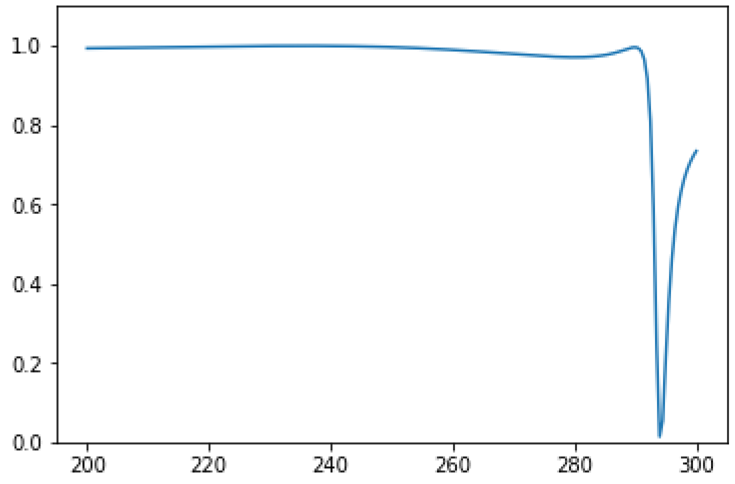
```
Test 35
True spectrum:
```



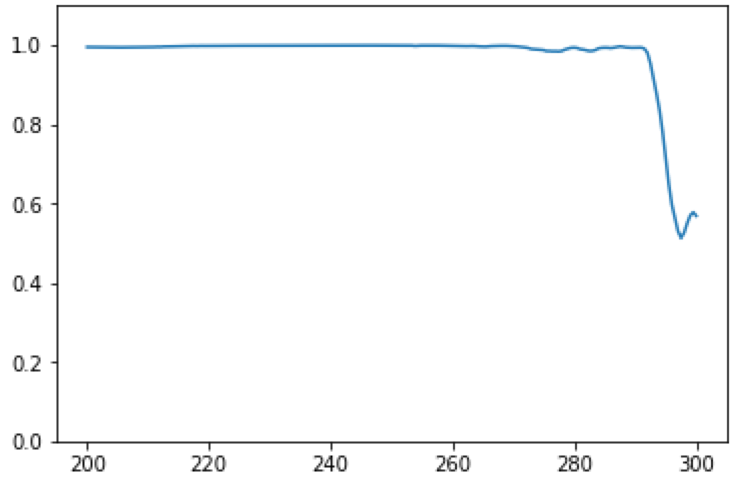
```
Predicted spectrum:
```



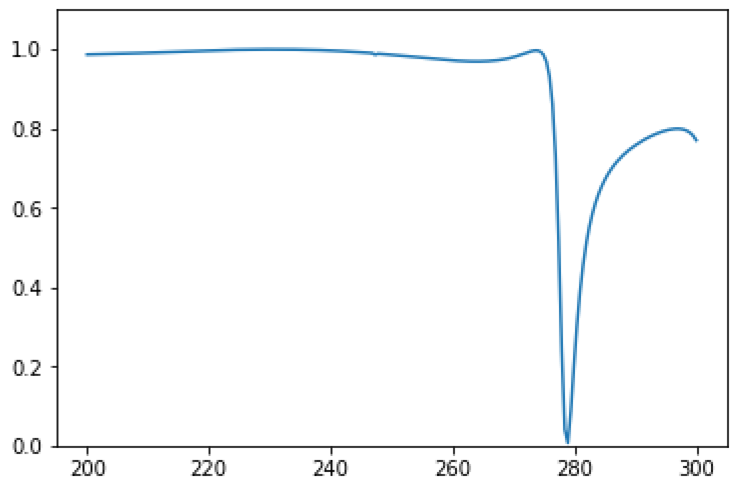
```
Test 36
True spectrum:
```



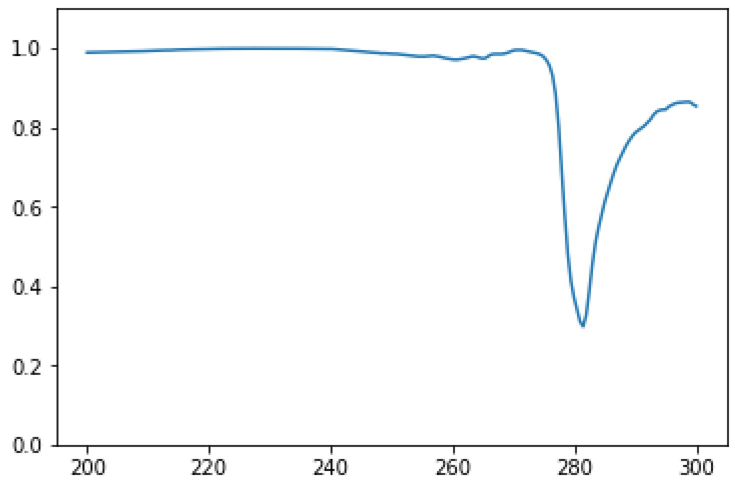
```
Predicted spectrum:
```



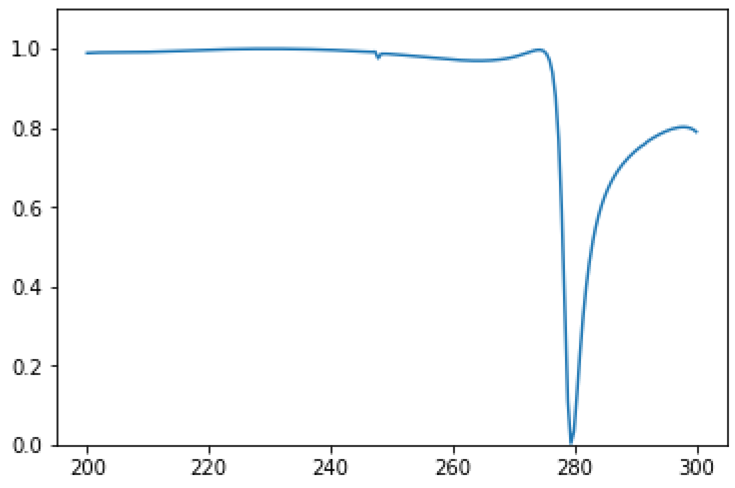
```
Test 37
True spectrum:
```



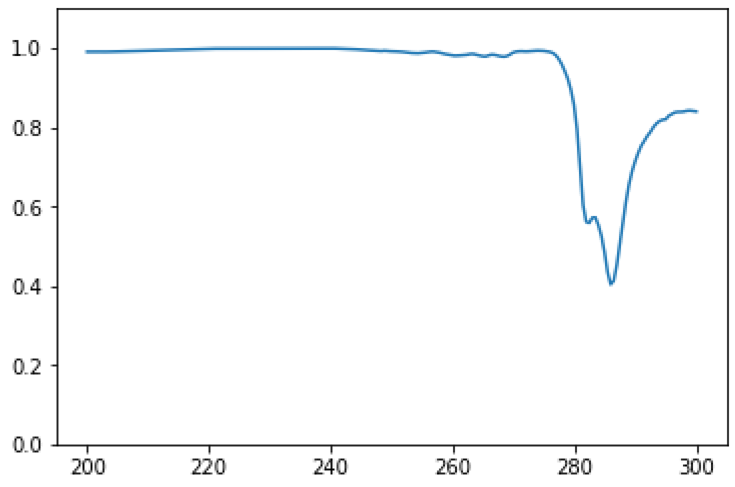
```
Predicted spectrum:
```



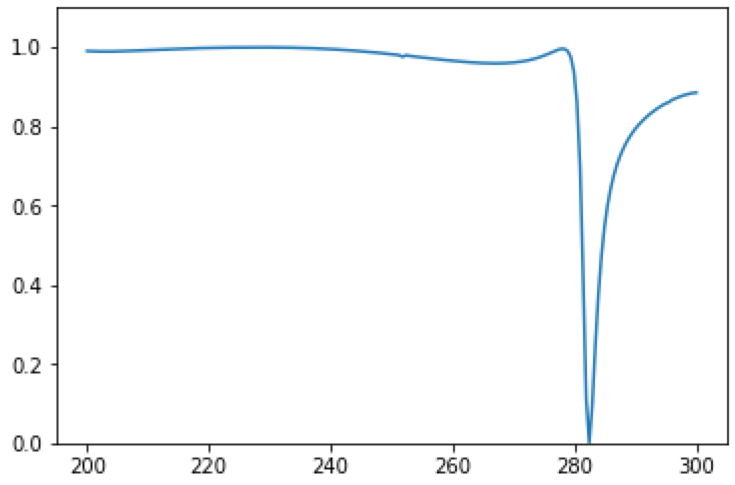
```
Test 38
True spectrum:
```



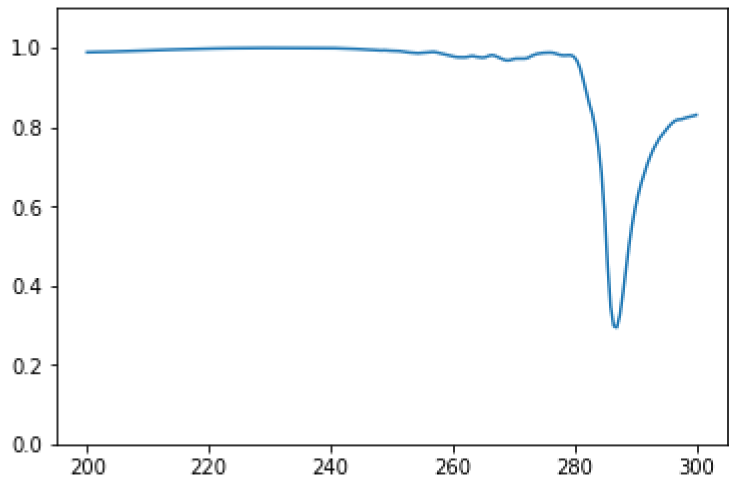
```
Predicted spectrum:
```



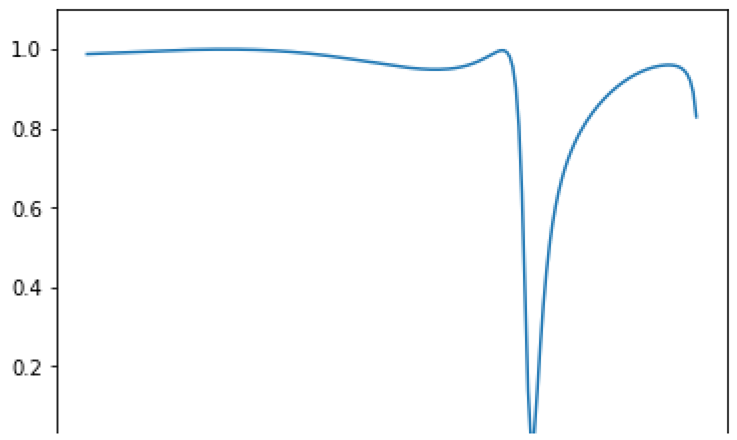
```
Test 39
True spectrum:
```



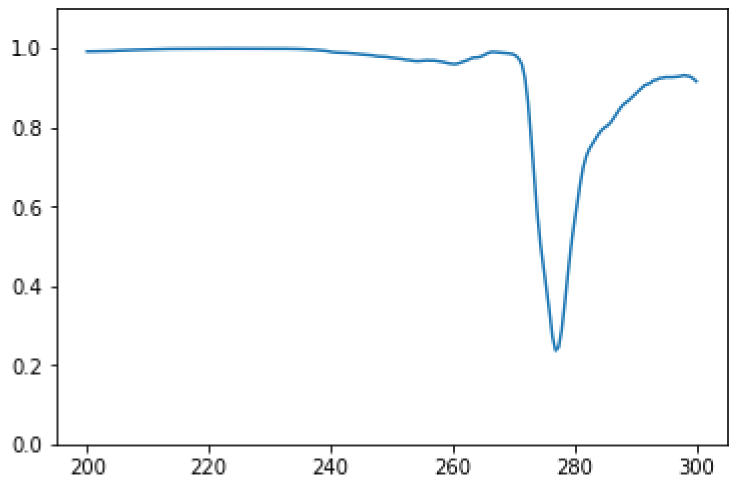
```
Predicted spectrum:
```



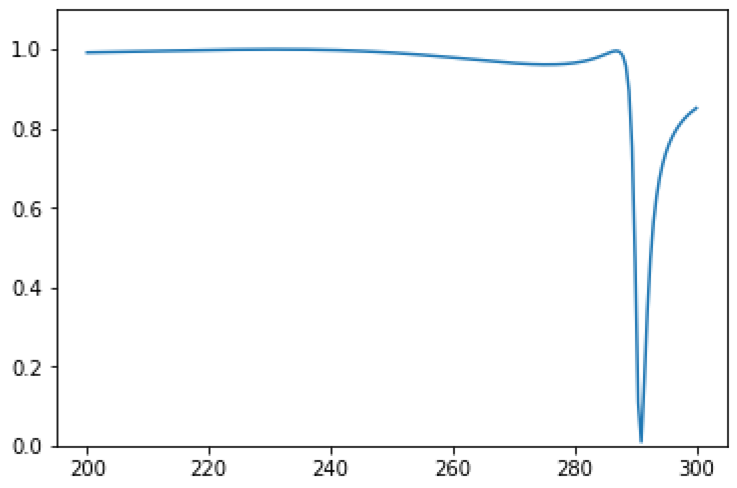
```
Test 40
True spectrum:
```



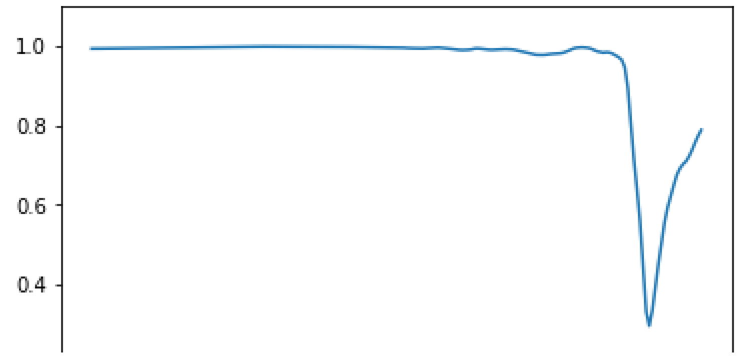
```
Predicted spectrum:
```



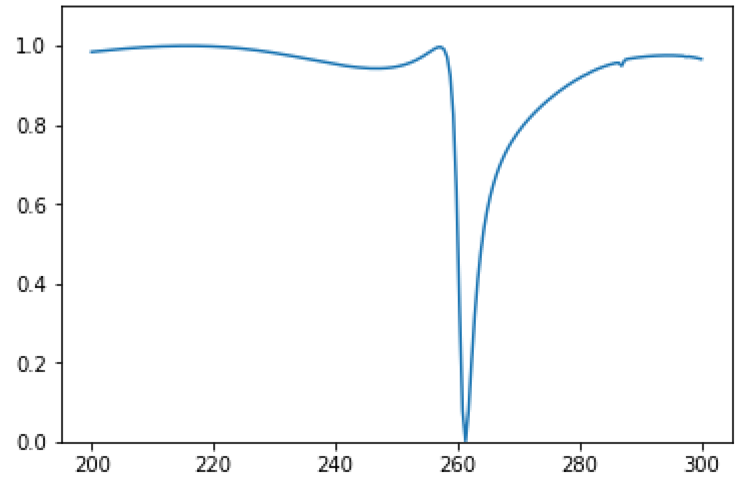
```
Test 41
True spectrum:
```



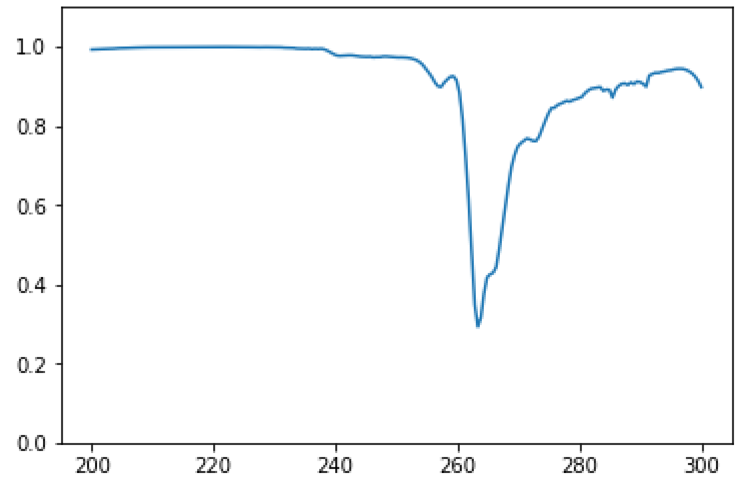
```
Predicted spectrum:
```



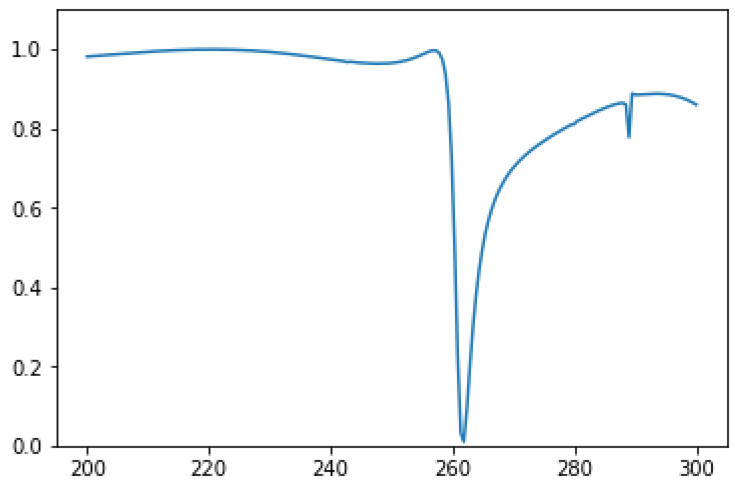
```
Test 42
True spectrum:
```



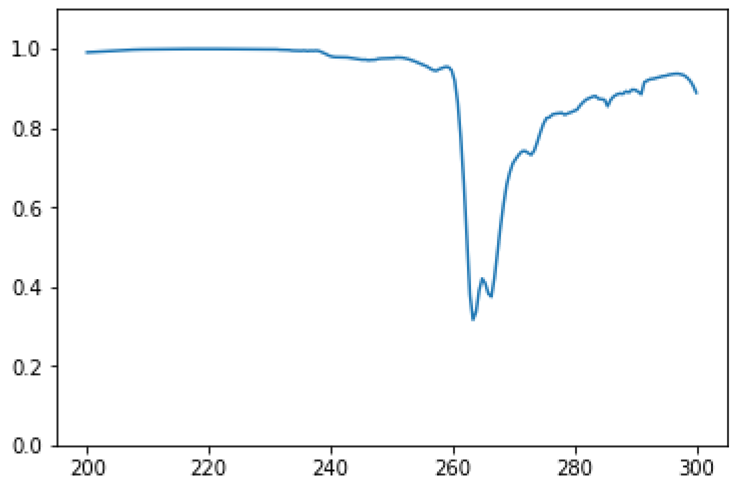
```
Predicted spectrum:
```



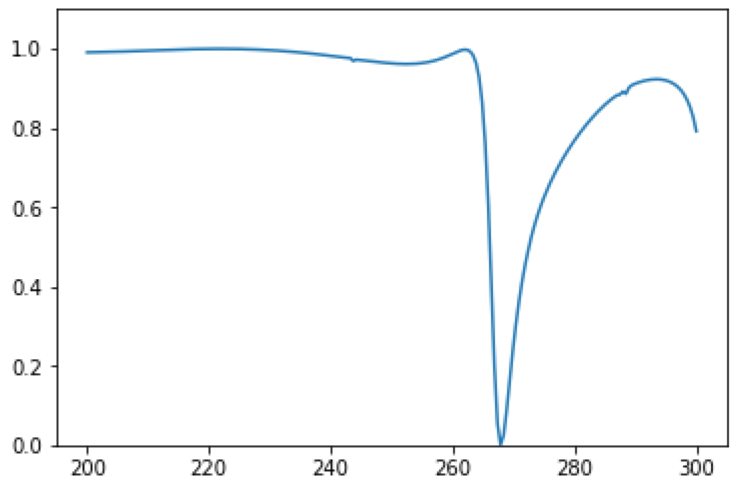
```
Test 43
True spectrum:
```



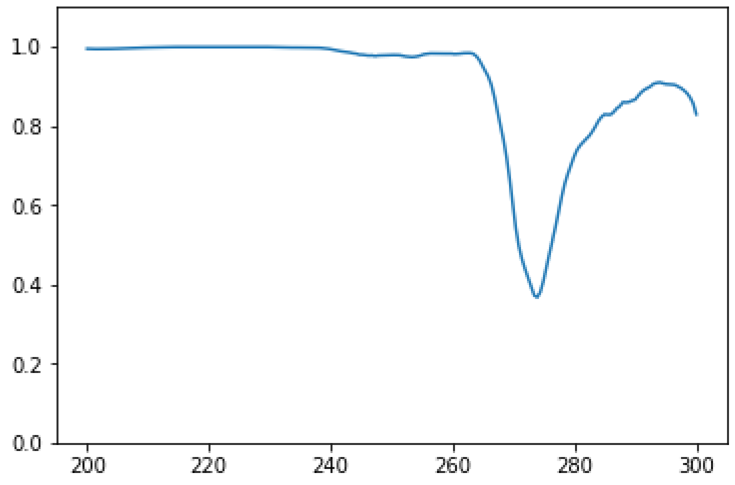
```
Predicted spectrum:
```



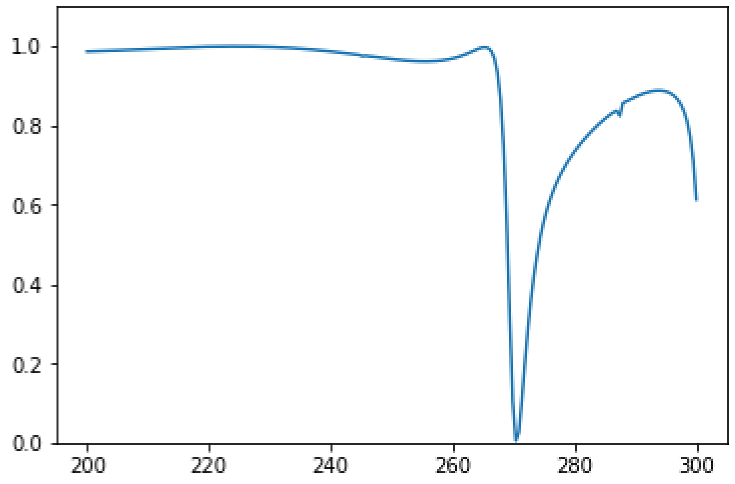
```
Test 44
True spectrum:
```



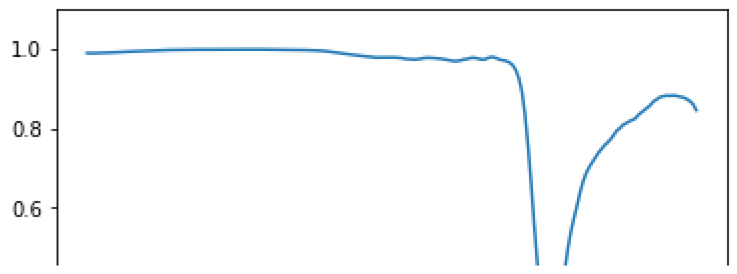
```
Predicted spectrum:
```



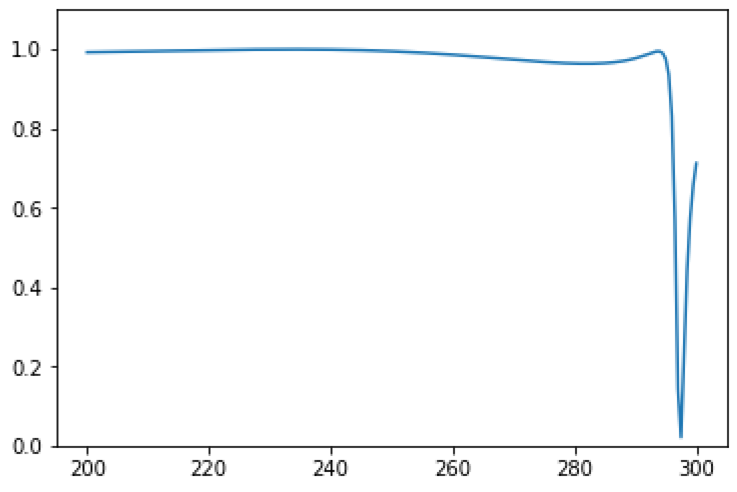
```
Test 45
True spectrum:
```



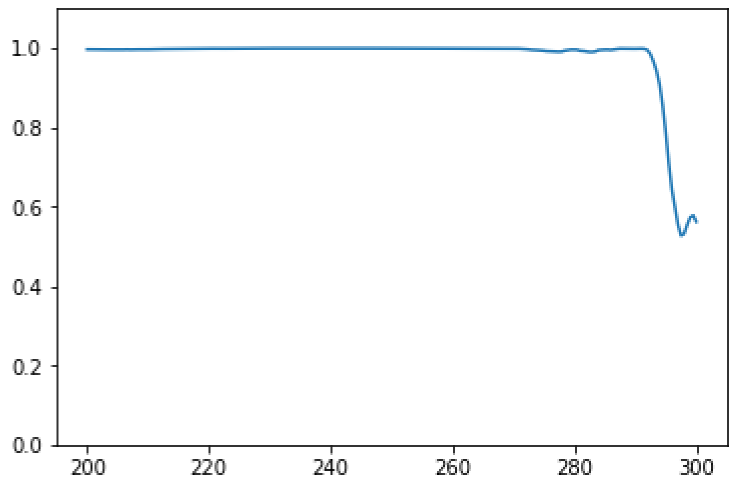
```
Predicted spectrum:
```



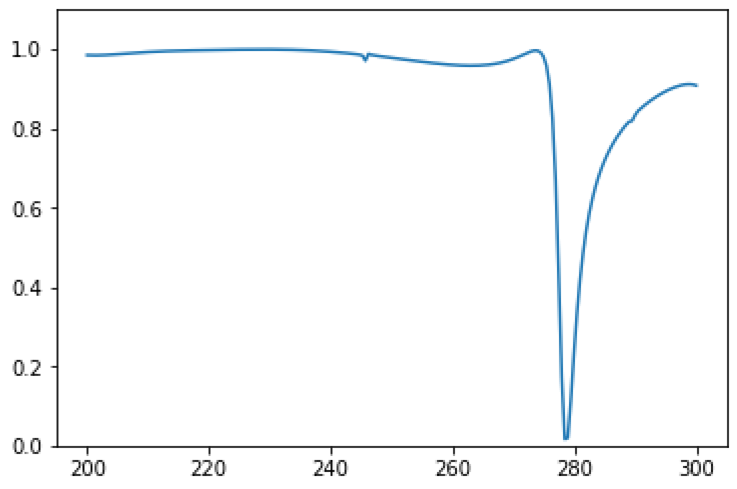
```
Test 46
True spectrum:
```



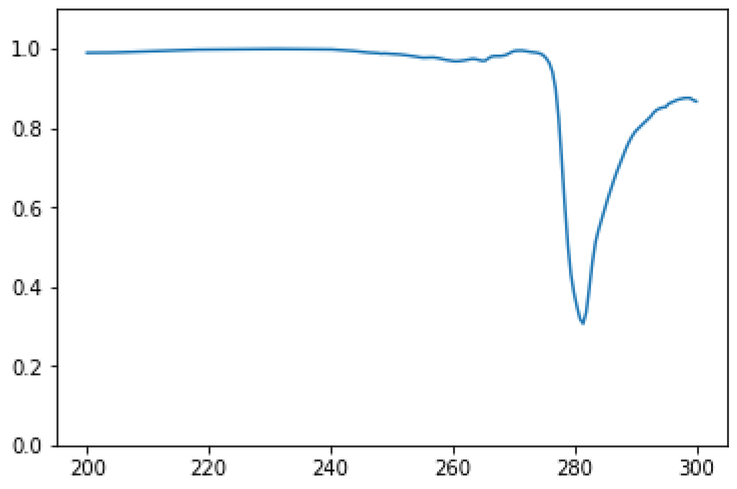
```
Predicted spectrum:
```



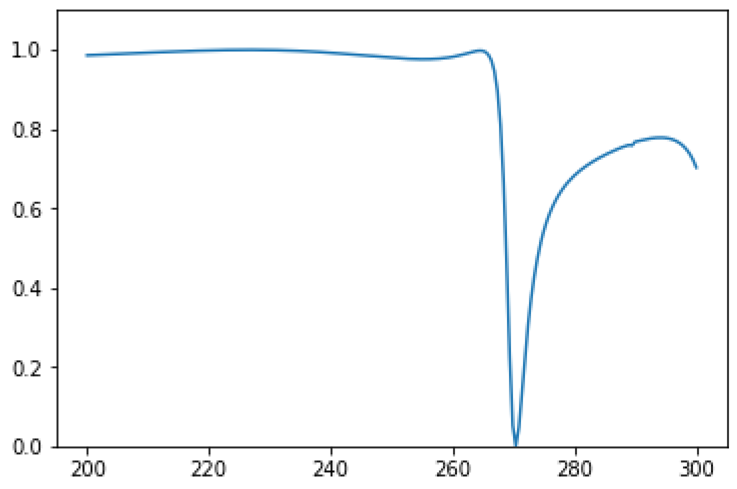
```
Test 47
True spectrum:
```



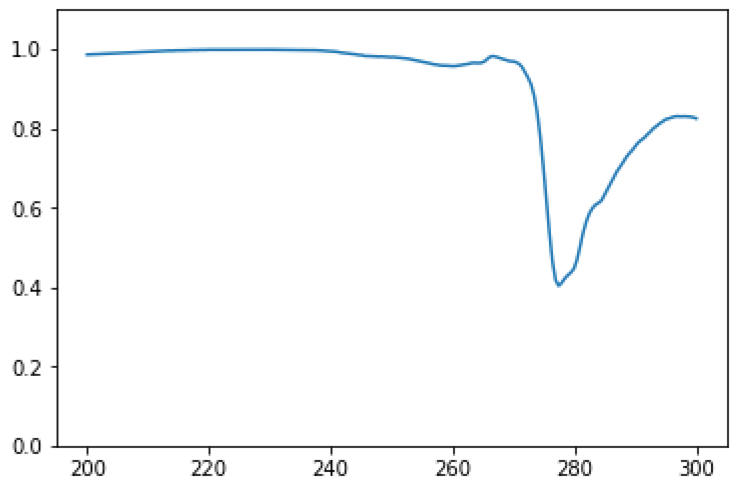
```
Predicted spectrum:
```



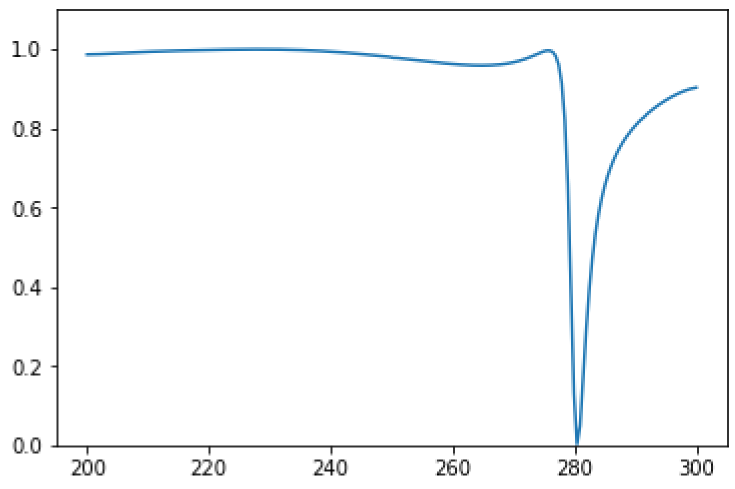
```
Test 48
True spectrum:
```



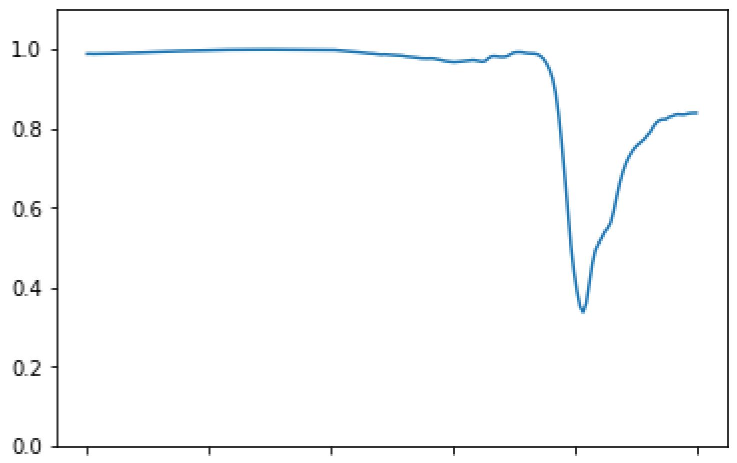
```
Predicted spectrum:
```



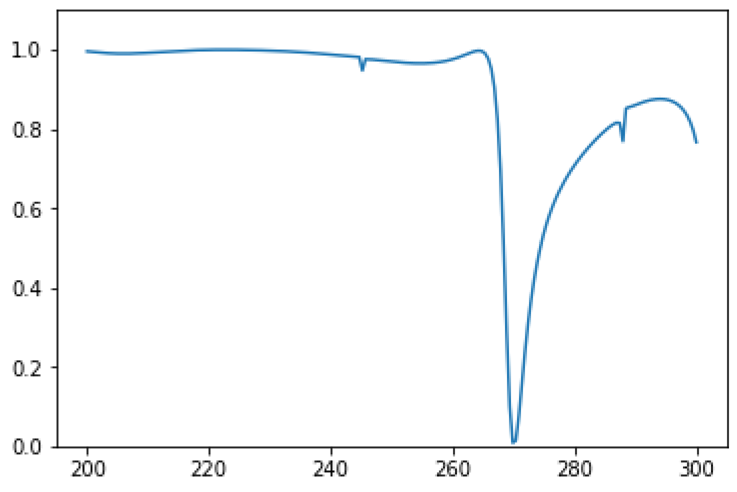
```
Test 49
True spectrum:
```



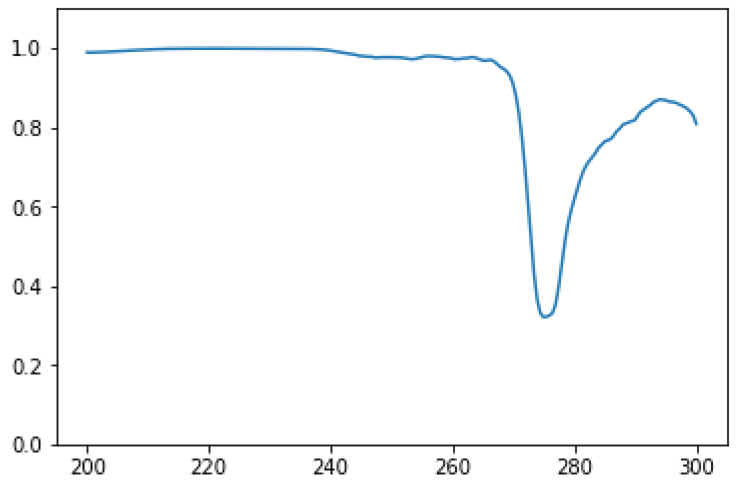
```
Predicted spectrum:
```



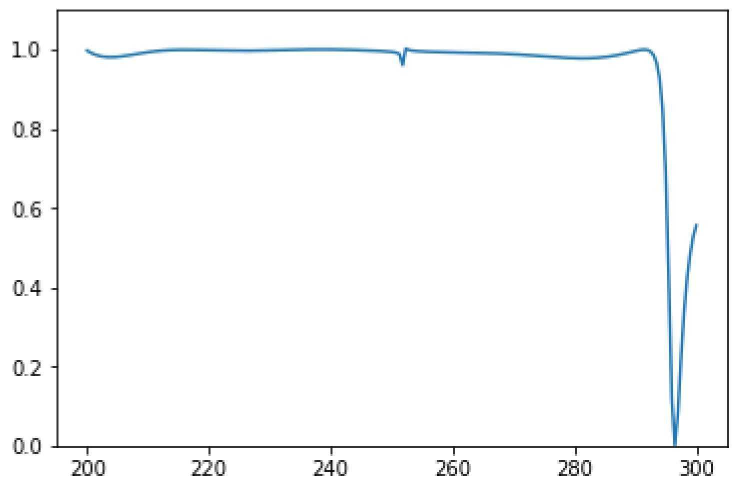
```
Test 50
True spectrum:
```



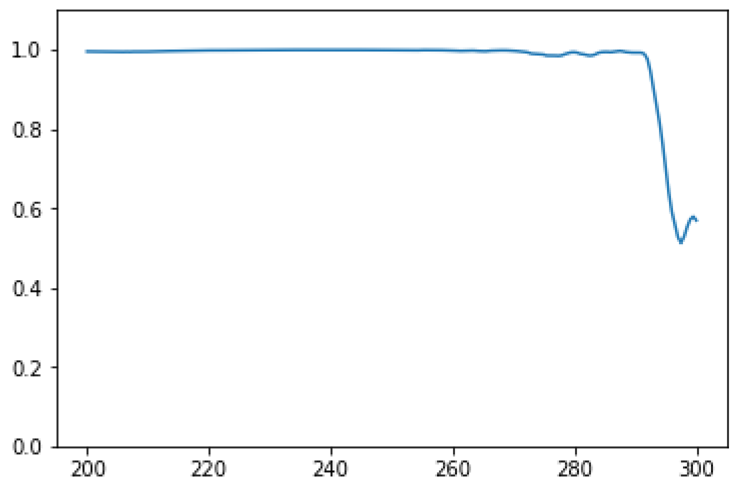
```
Predicted spectrum:
```



```
Test 51
True spectrum:
```

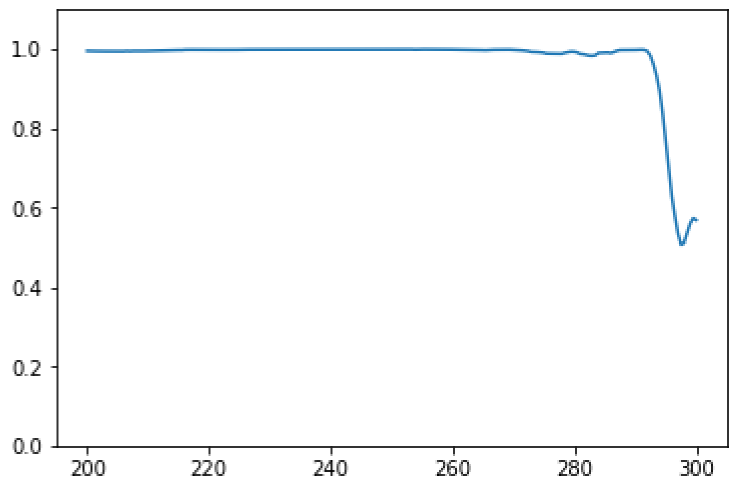


```
Predicted spectrum:
```

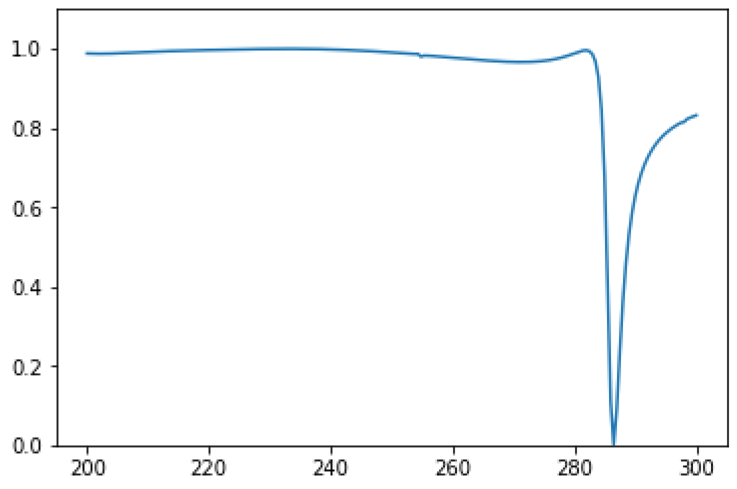


```
Test 52
True spectrum:
```

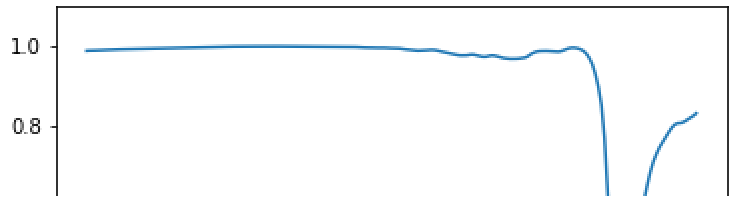
```
Predicted spectrum:
```



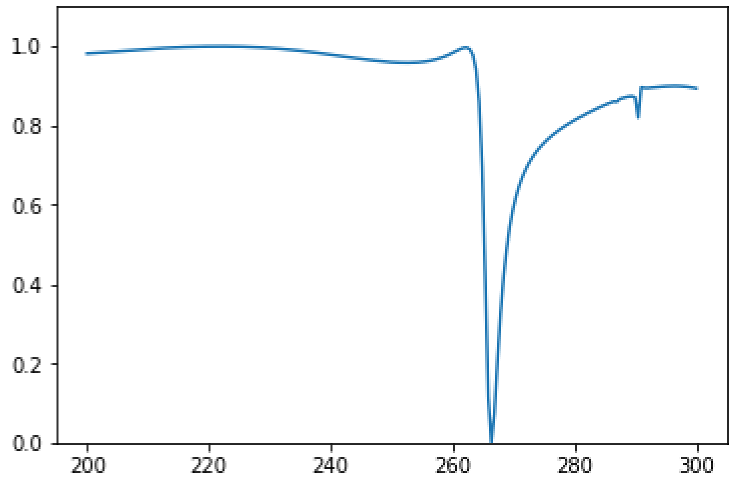
```
Test 53
True spectrum:
```



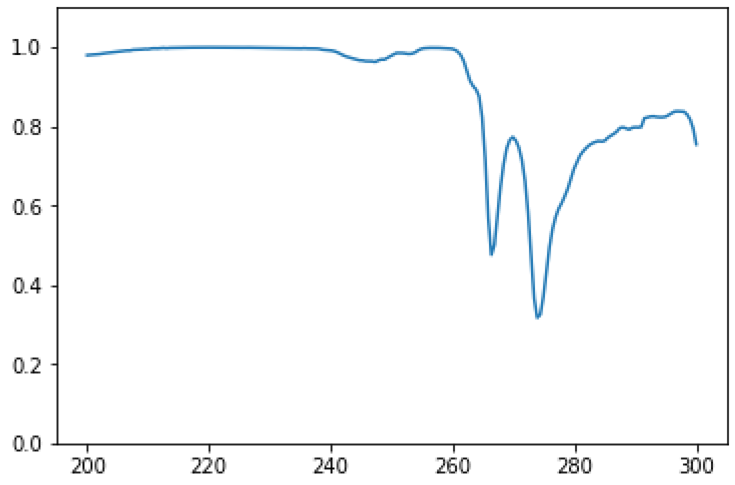
```
Predicted spectrum:
```



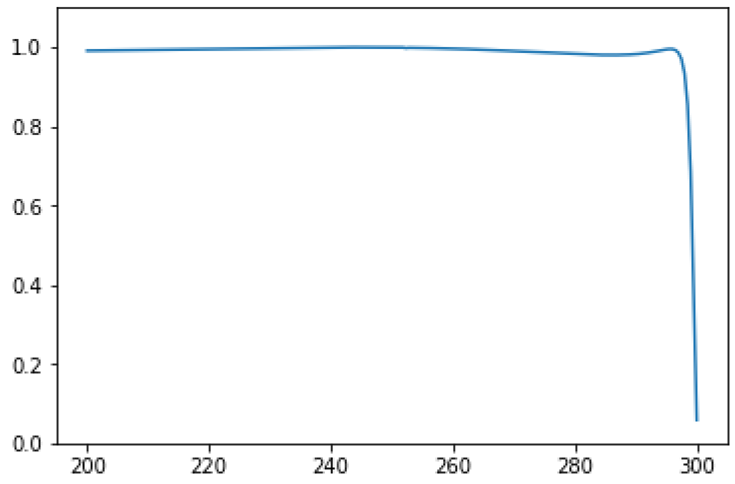
```
Test 54
True spectrum:
```



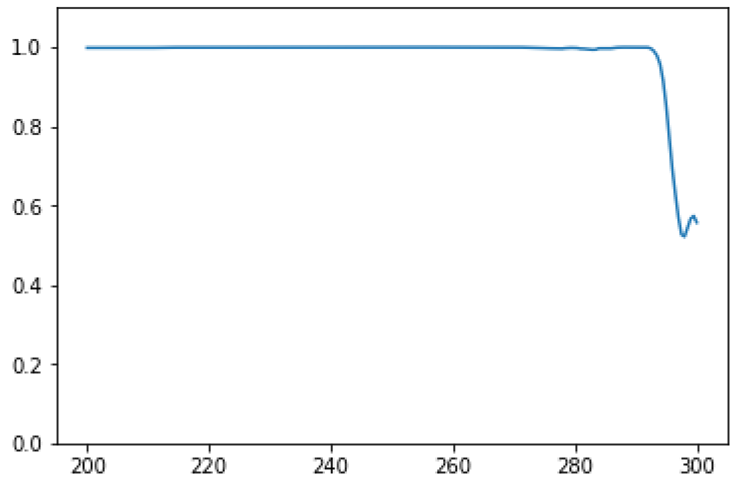
```
Predicted spectrum:
```



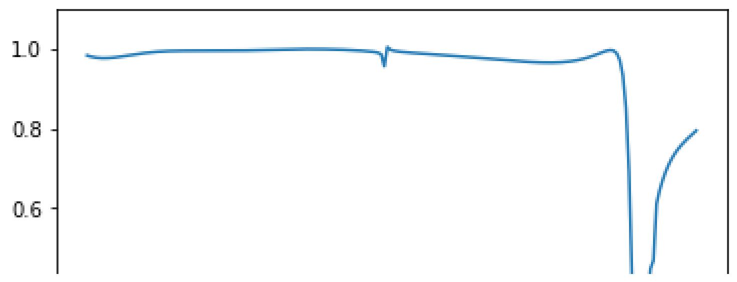
```
Test 55
True spectrum:
```



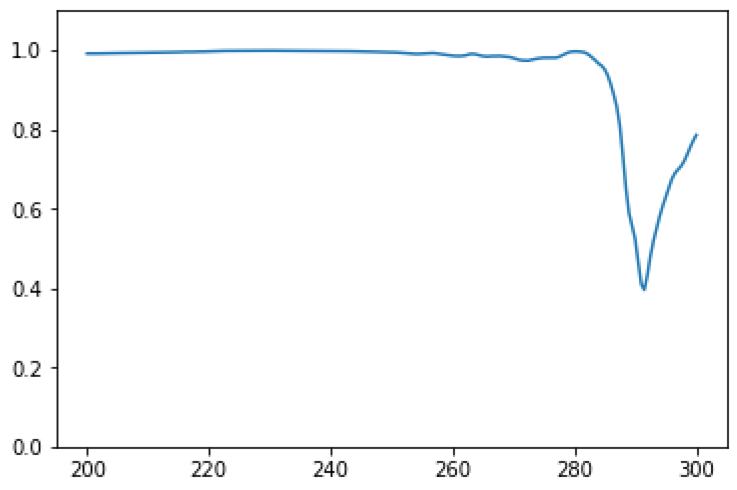
```
Predicted spectrum:
```



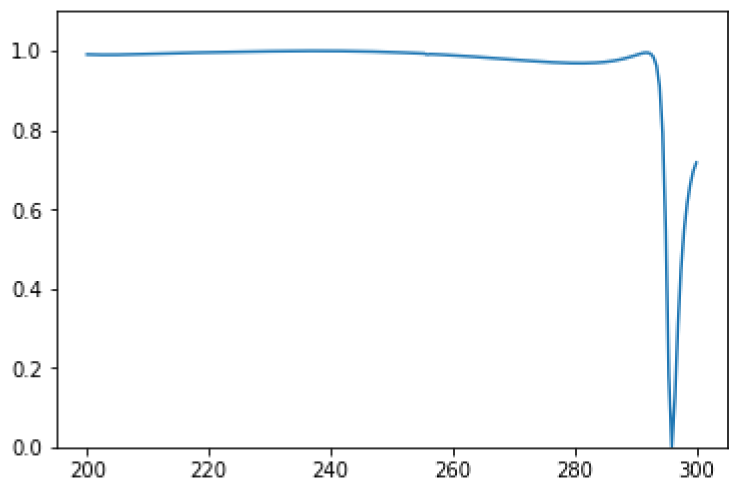
```
Test 56
True spectrum:
```



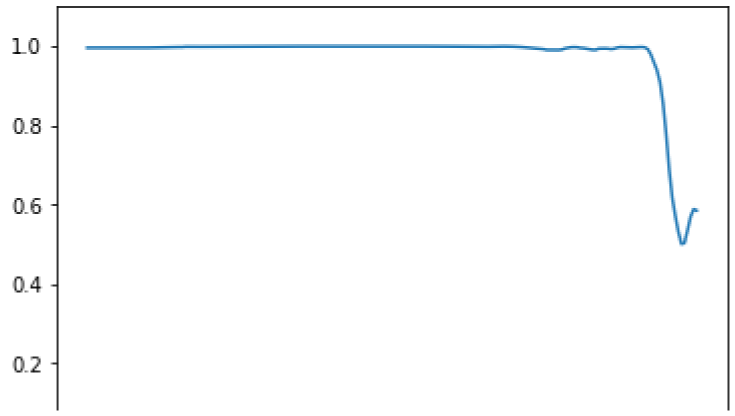
```
Predicted spectrum:
```



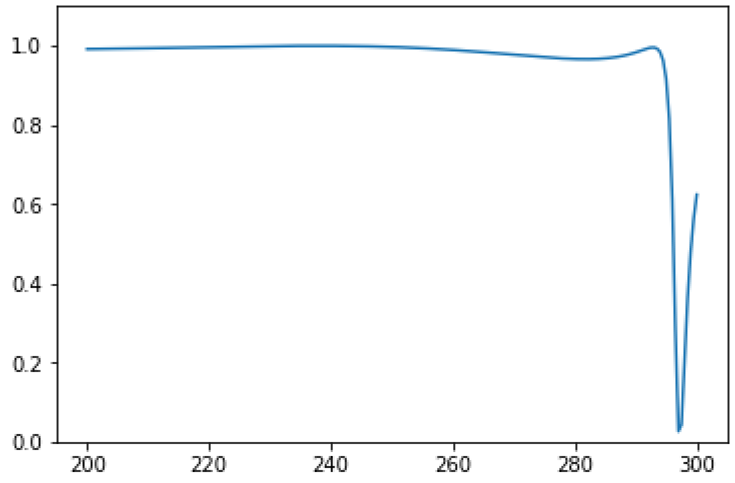
```
Test 57
True spectrum:
```



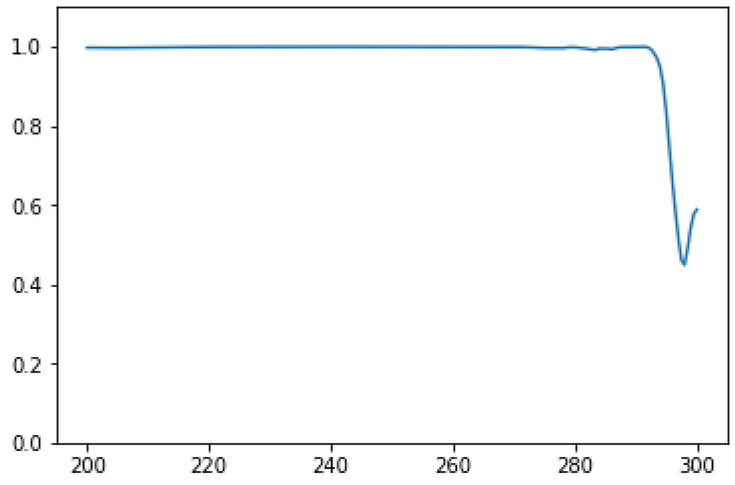
```
Predicted spectrum:
```



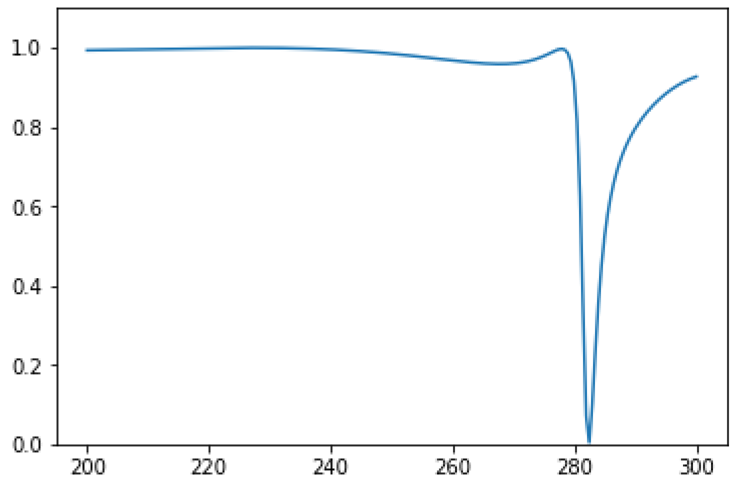
```
Test 58
True spectrum:
```



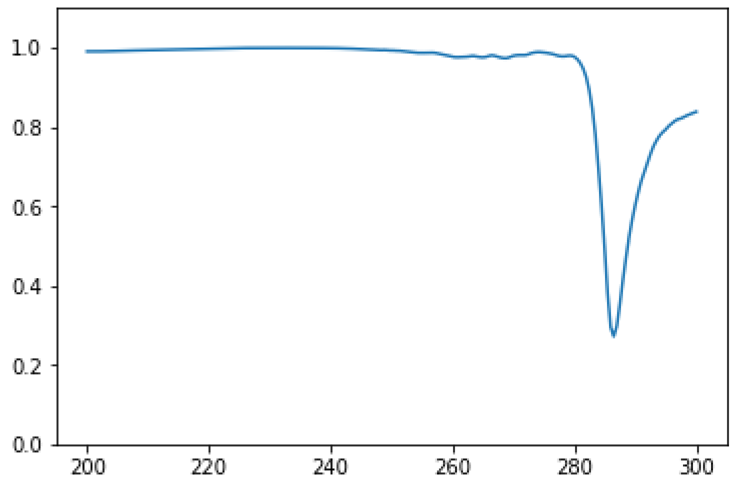
```
Predicted spectrum:
```



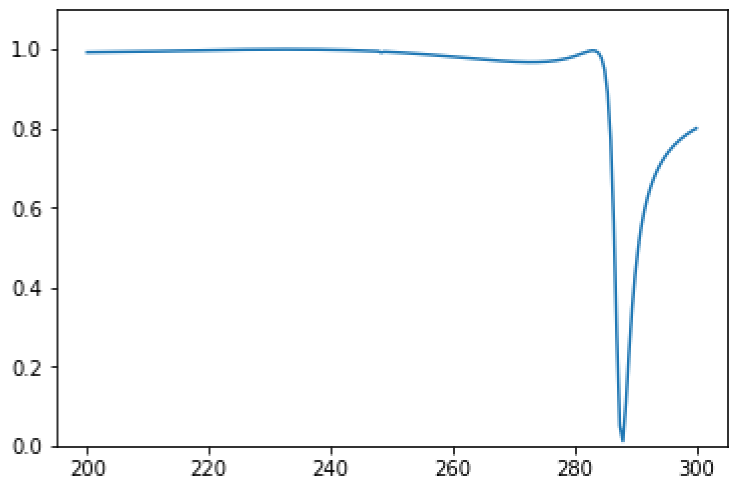
```
Test 59
True spectrum:
```



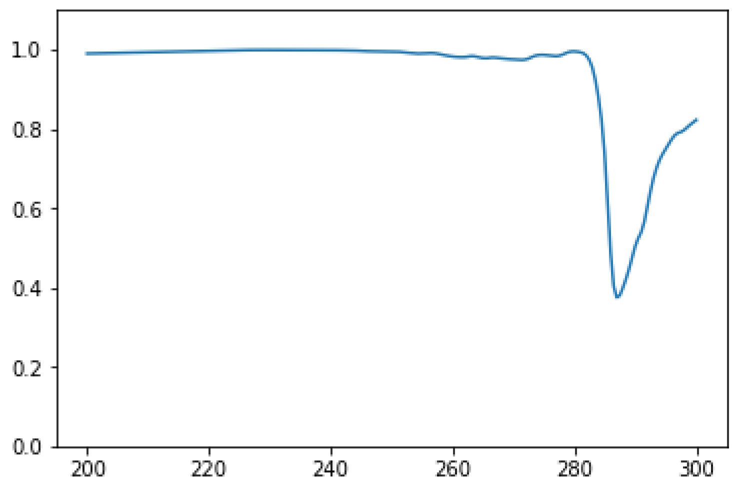
```
Predicted spectrum:
```



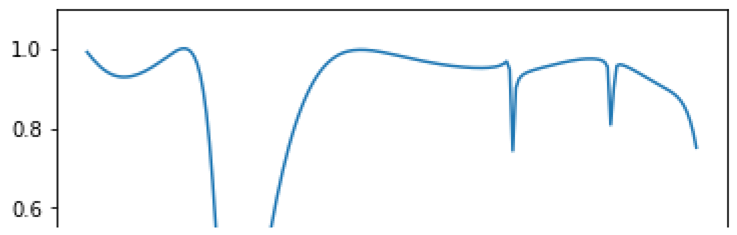
```
Test 60
True spectrum:
```



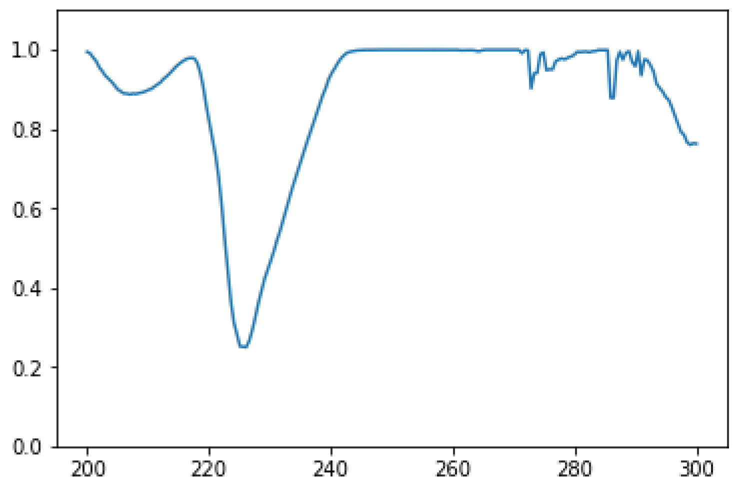
```
Predicted spectrum:
```



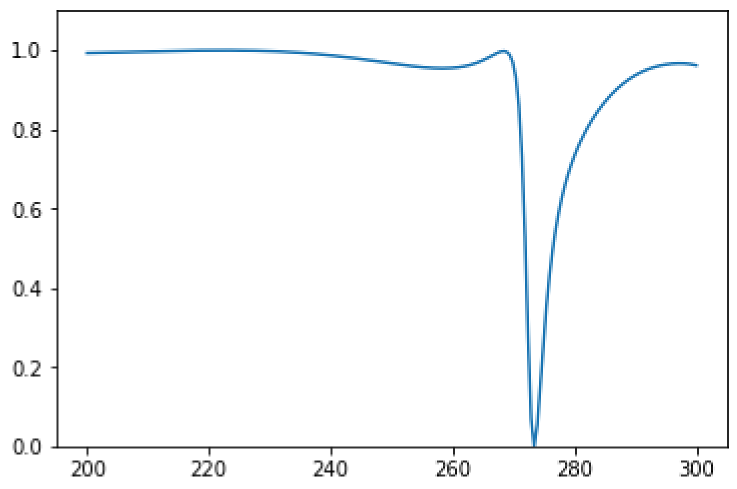
```
Test 61
True spectrum:
```



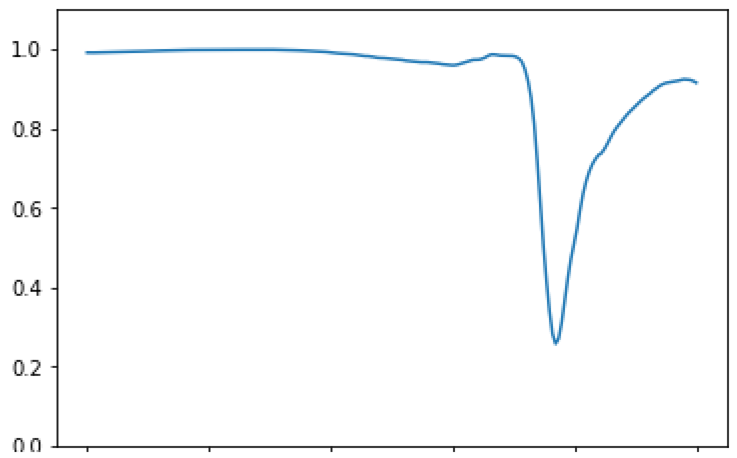
```
Predicted spectrum:
```



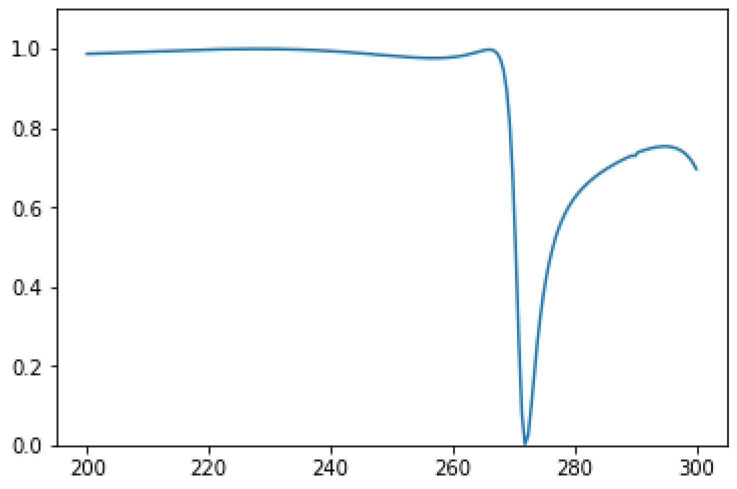
```
Test 62
True spectrum:
```



```
Predicted spectrum:
```

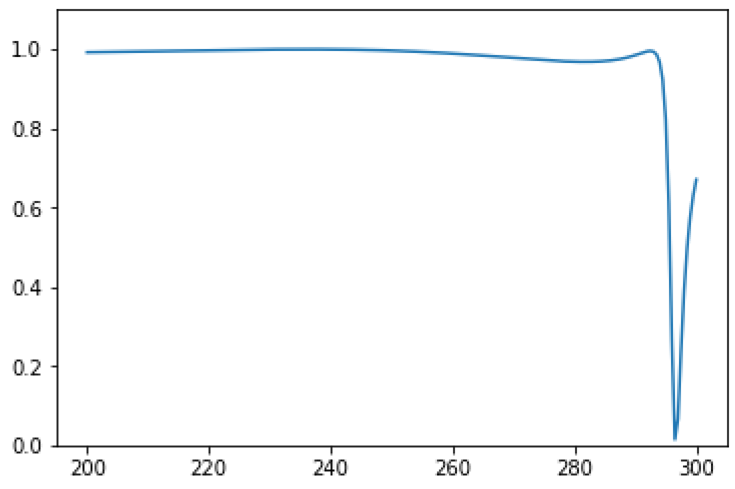


```
Test 63
True spectrum:
```

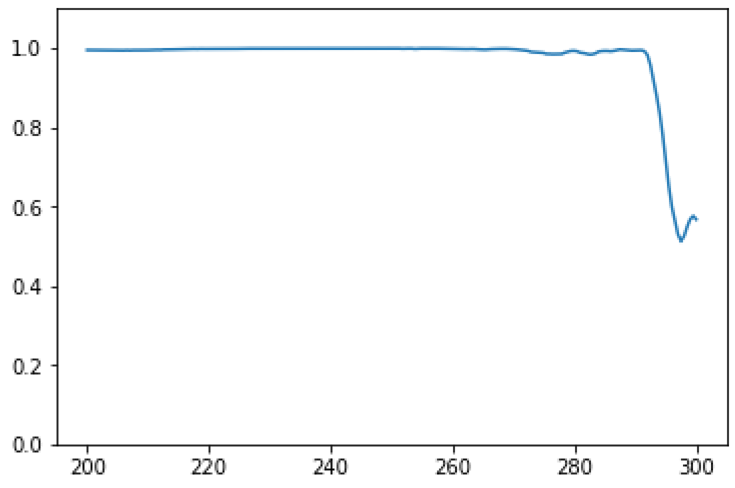


```
Predicted spectrum:
```

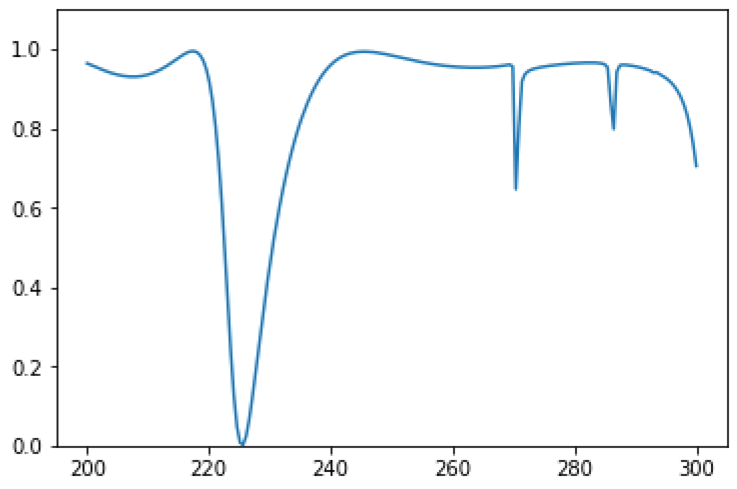
```
Test 64
True spectrum:
```



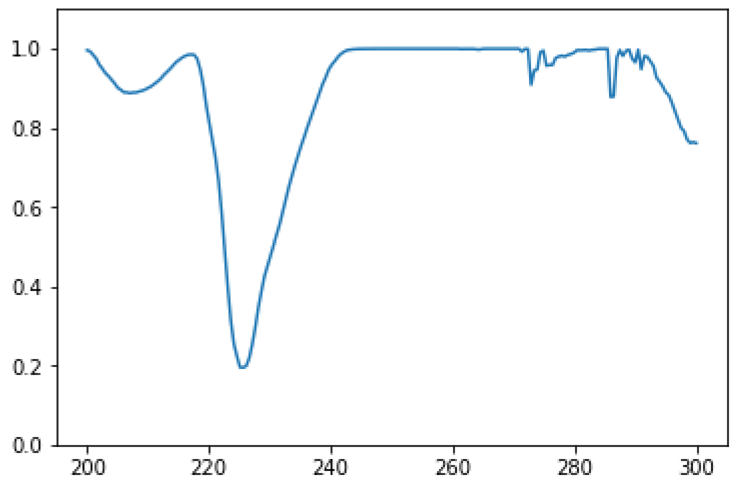
```
Predicted spectrum:
```



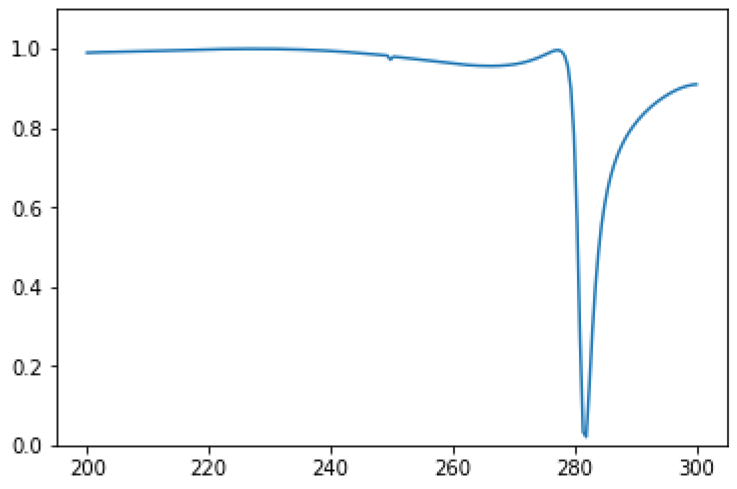
```
Test 65
True spectrum:
```



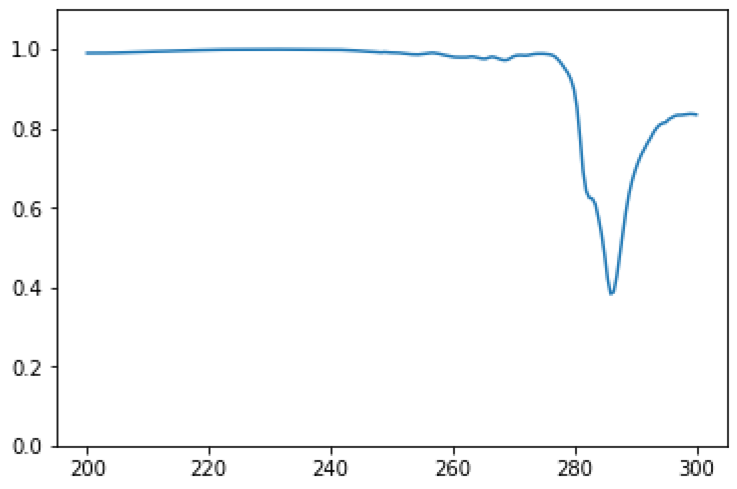
```
Predicted spectrum:
```



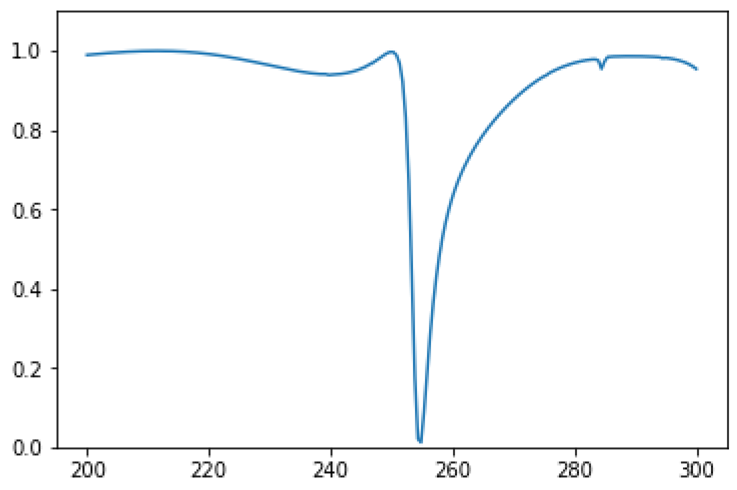
```
Test 66
True spectrum:
```



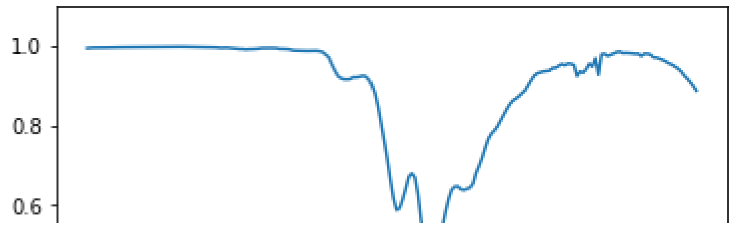
```
Predicted spectrum:
```



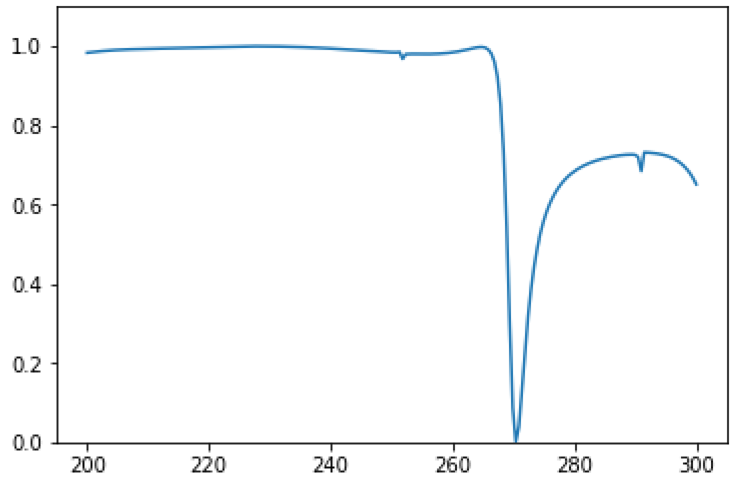
```
Test 67
True spectrum:
```



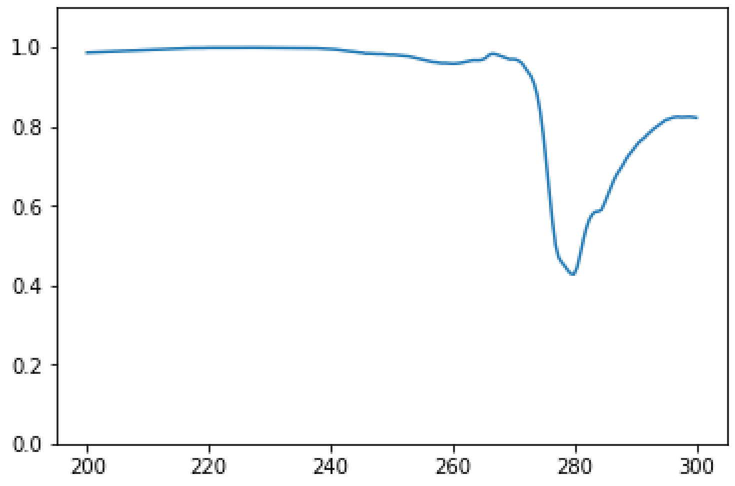
```
Predicted spectrum:
```



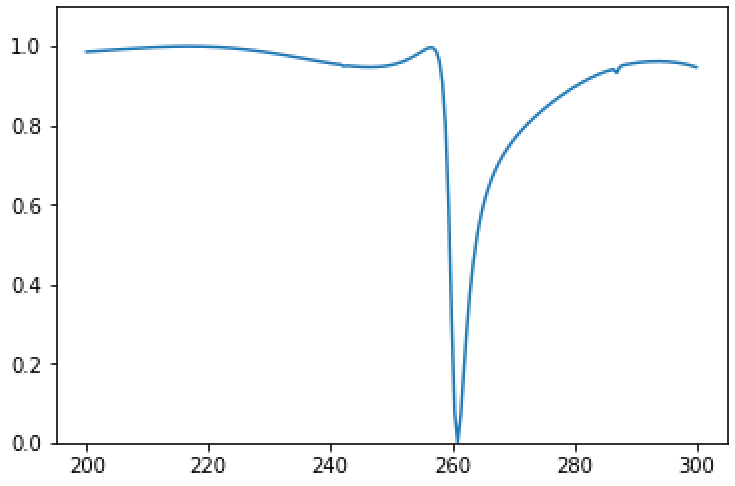
```
Test 68
True spectrum:
```



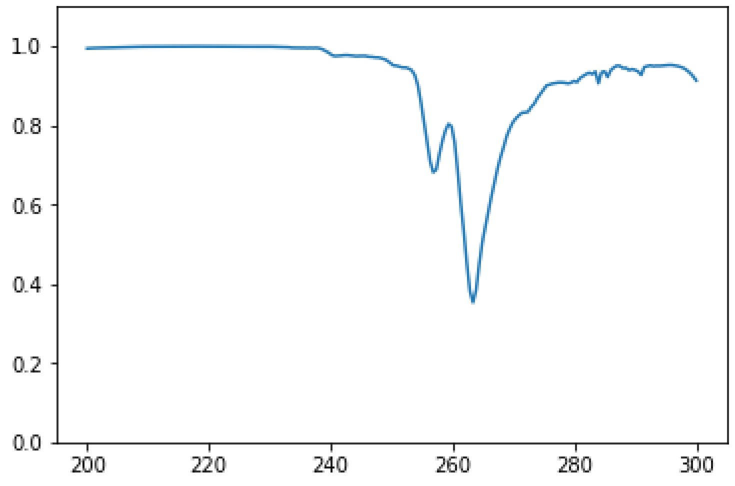
```
Predicted spectrum:
```



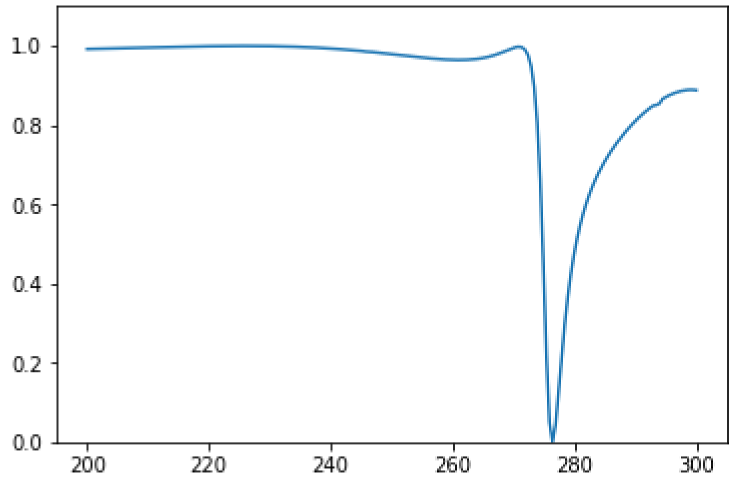
```
Test 69
True spectrum:
```



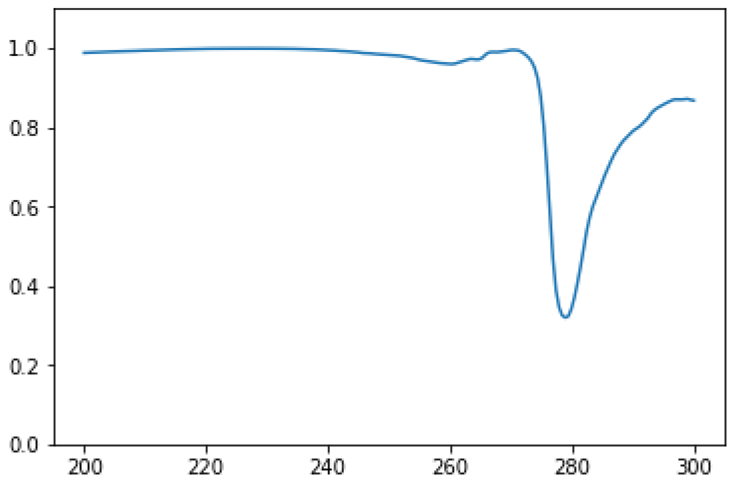
```
Predicted spectrum:
```



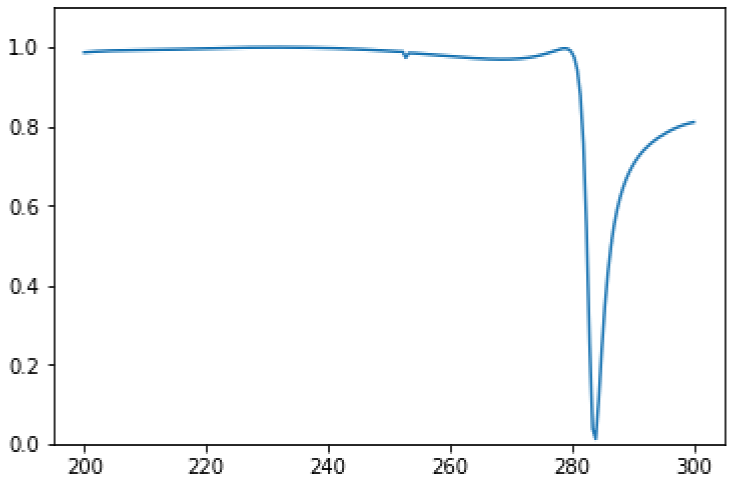
```
Test 70
True spectrum:
```



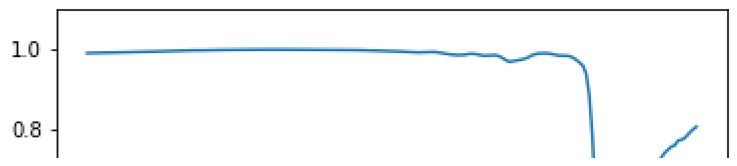
```
Predicted spectrum:
```



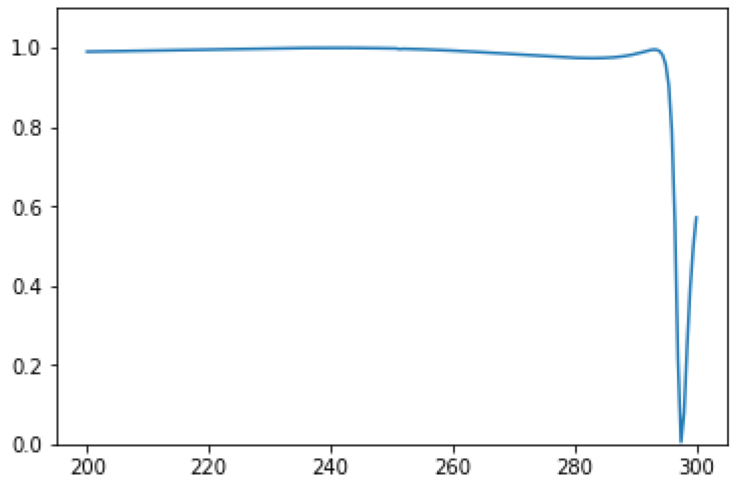
```
Test 71
True spectrum:
```



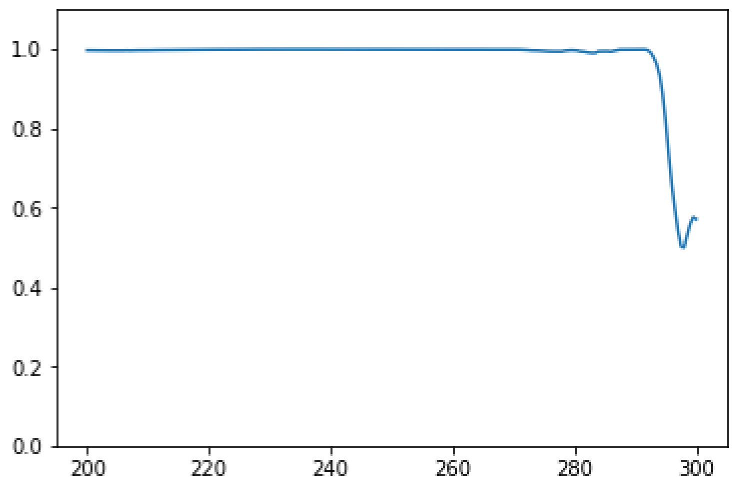
```
Predicted spectrum:
```



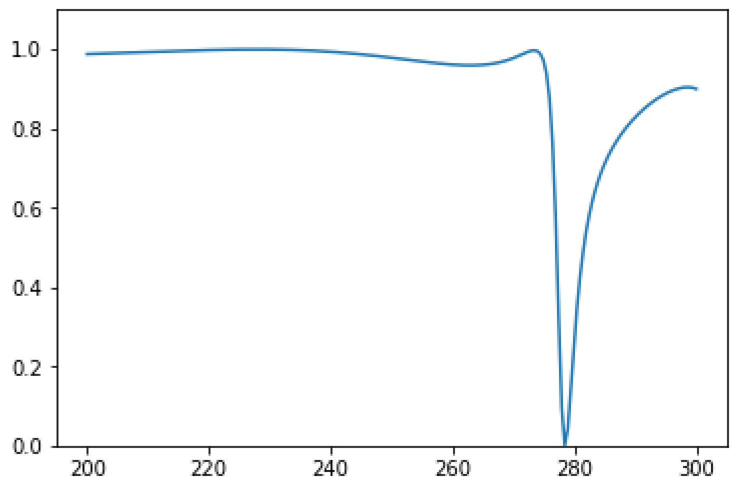
```
Test 72
True spectrum:
```



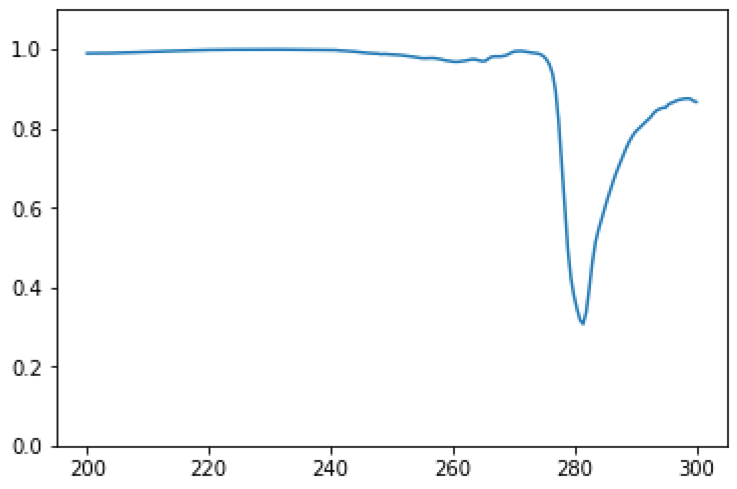
```
Predicted spectrum:
```



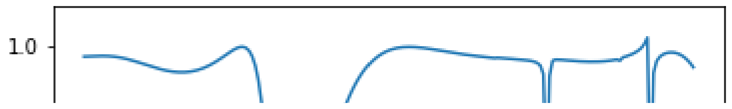
```
Test 73
True spectrum:
```



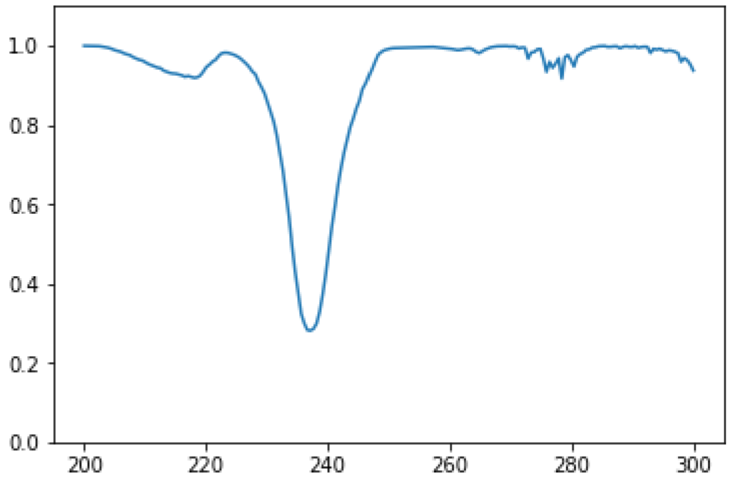
```
Predicted spectrum:
```



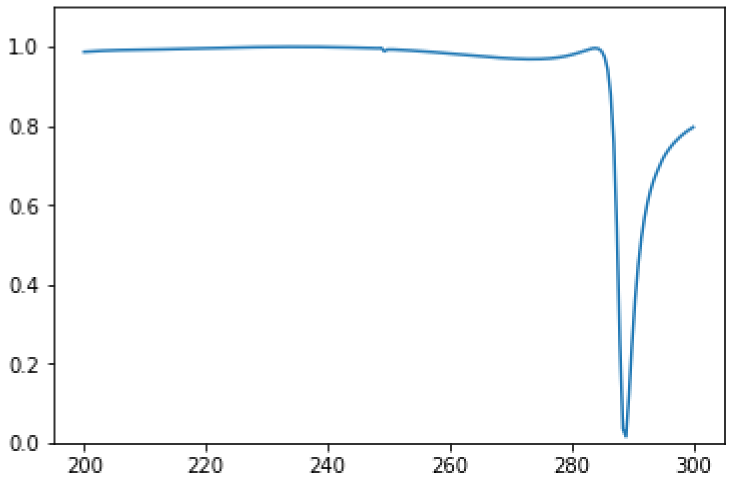
```
Test 74
True spectrum:
```



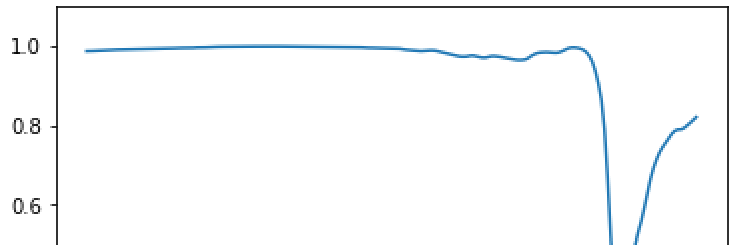
```
Predicted spectrum:
```



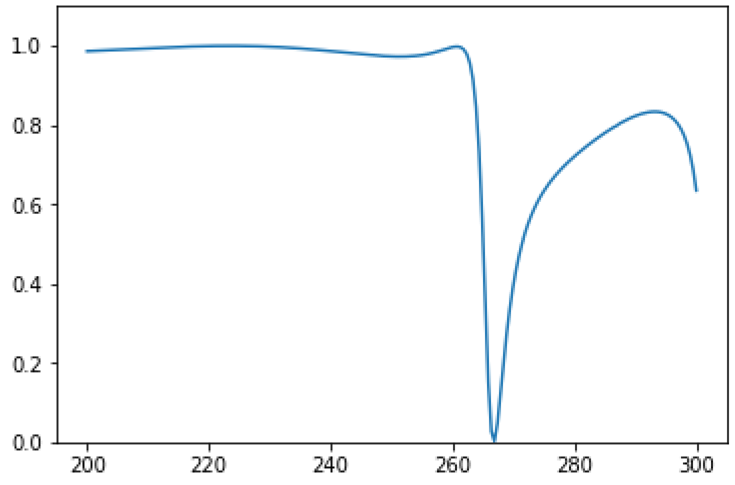
```
Test 75
True spectrum:
```



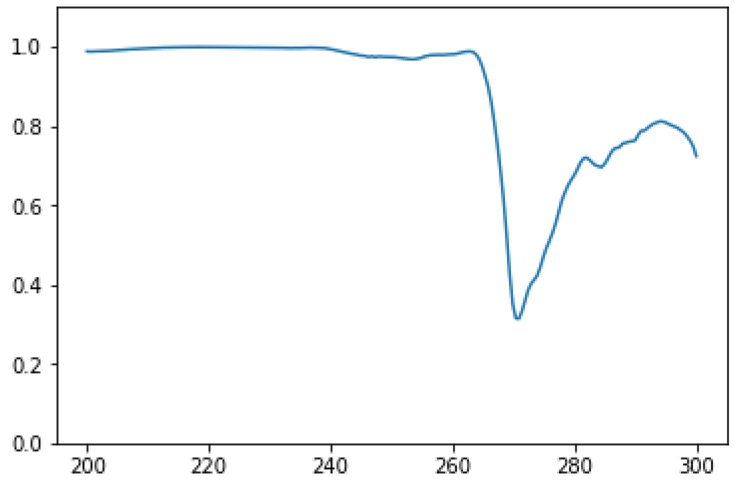
```
Predicted spectrum:
```



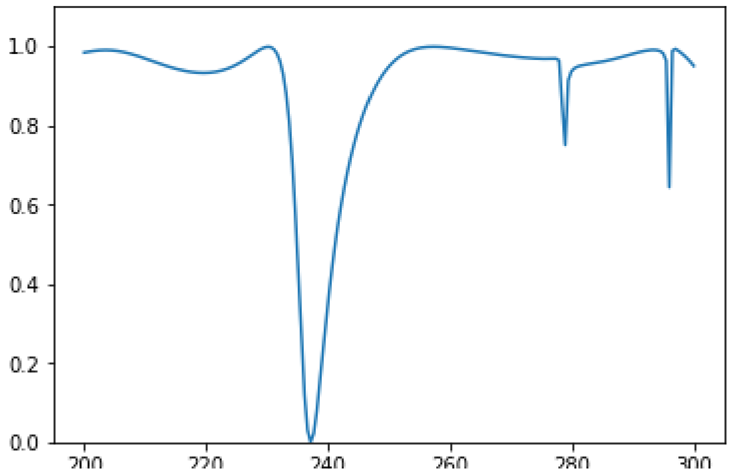
```
Test 76
True spectrum:
```



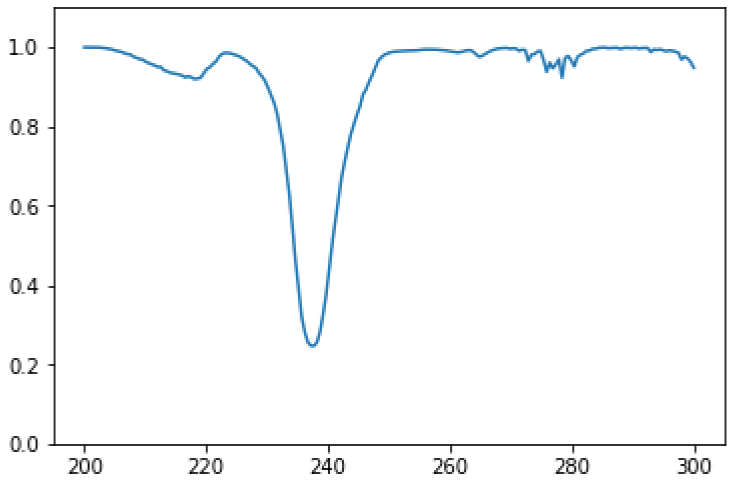
```
Predicted spectrum:
```



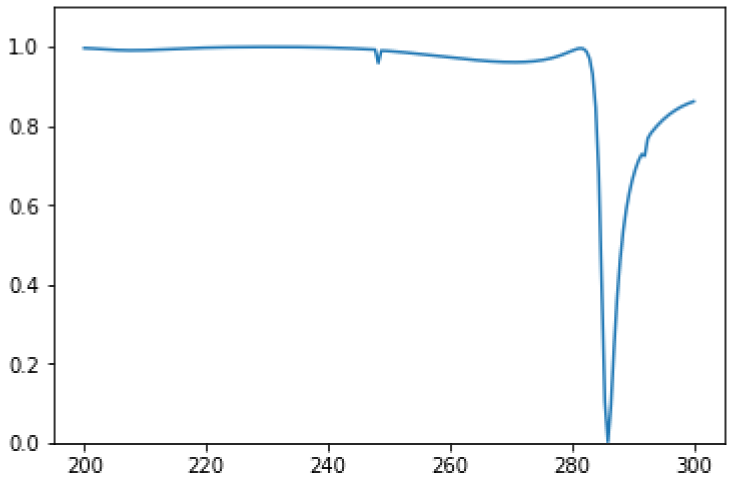
```
Test 77
True spectrum:
```



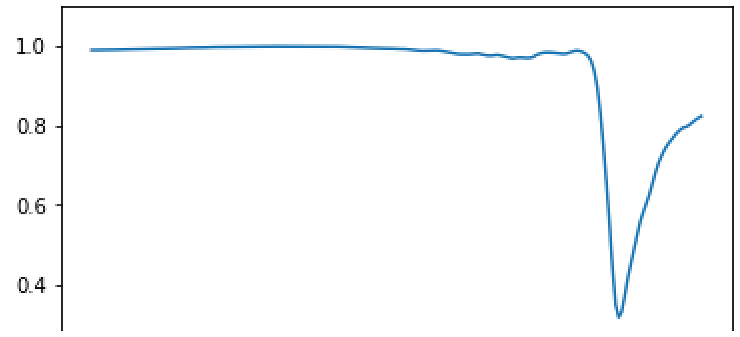
```
Predicted spectrum:
```



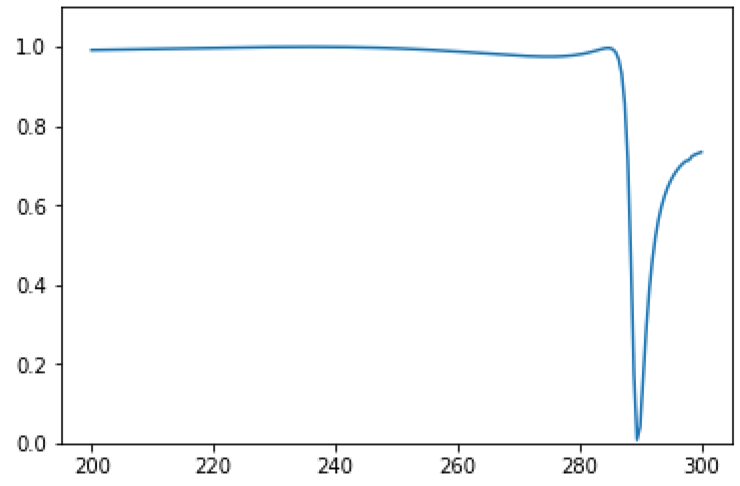
```
Test 78
True spectrum:
```



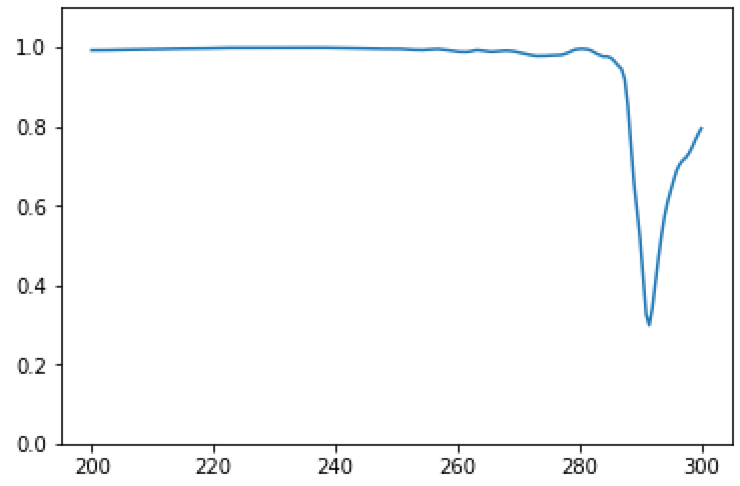
```
Predicted spectrum:
```



```
Test 79
True spectrum:
```

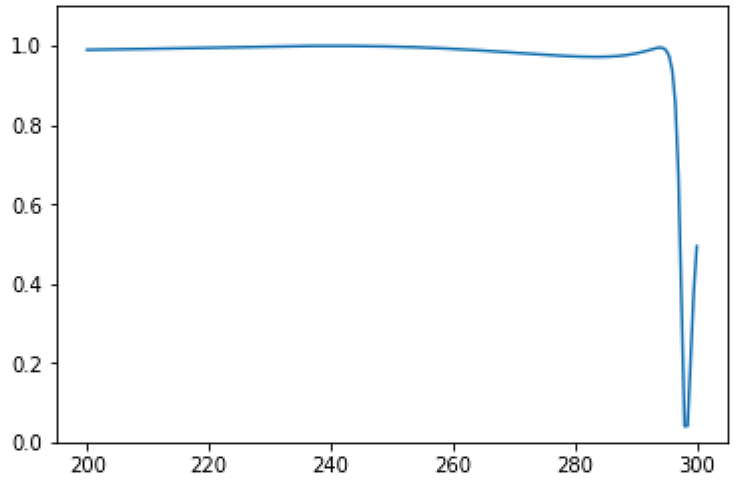


```
Predicted spectrum:
```

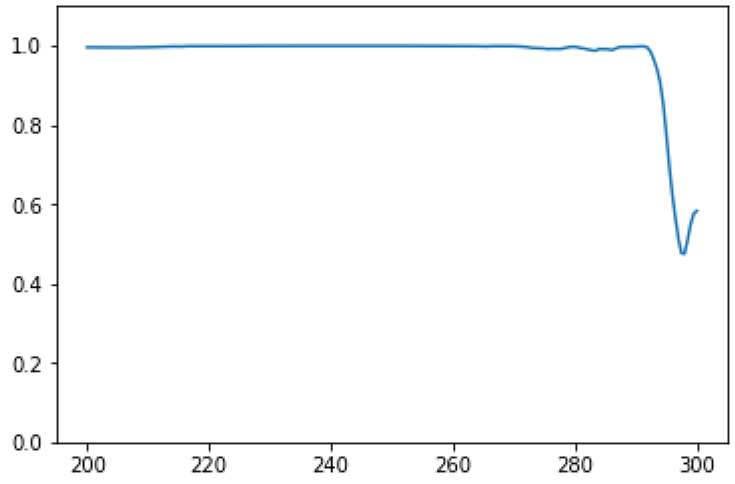


```
Test 80
```

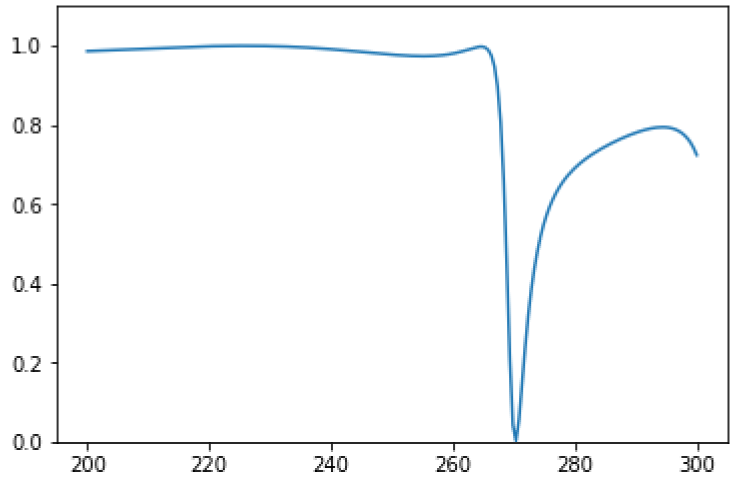
```
True spectrum:
```



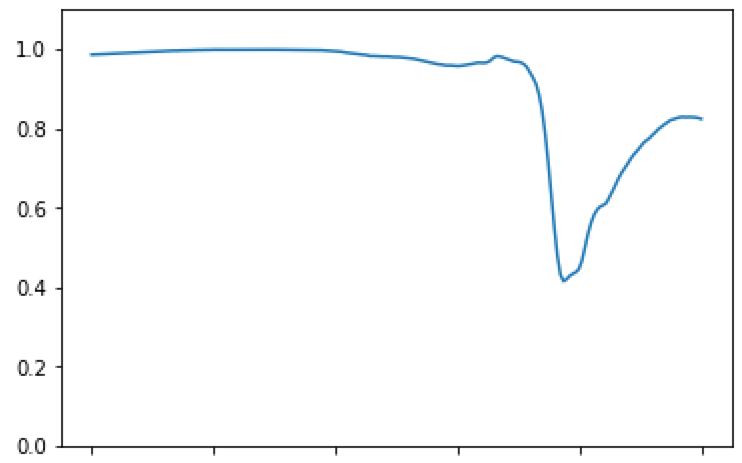
```
Predicted spectrum:
```



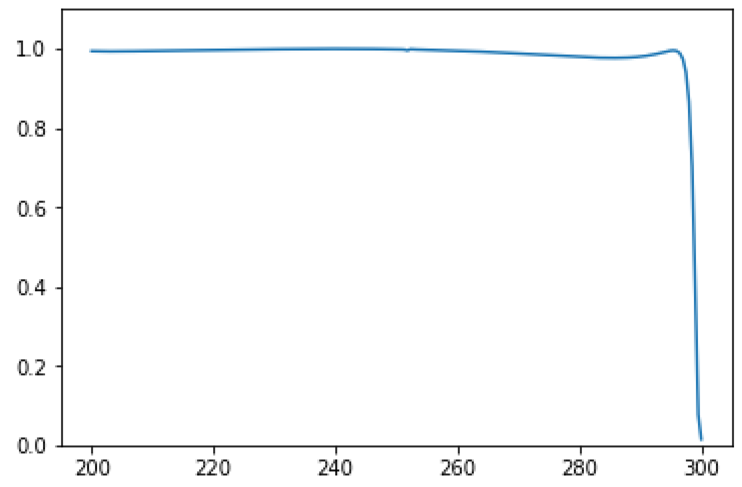
```
Test 81
True spectrum:
```



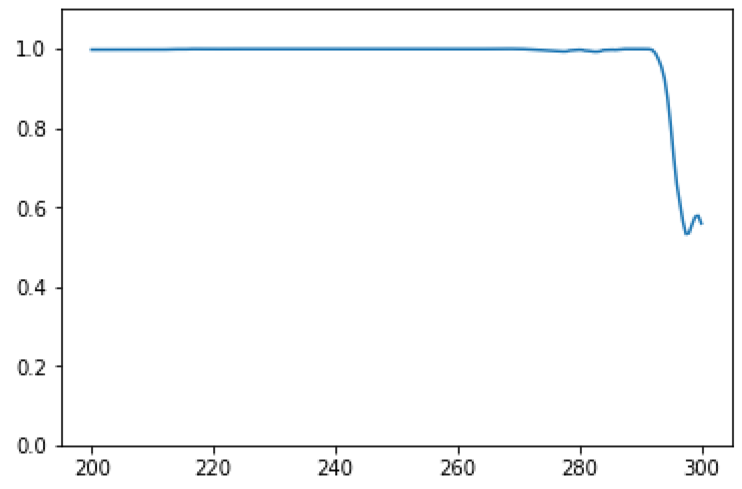
```
Predicted spectrum:
```



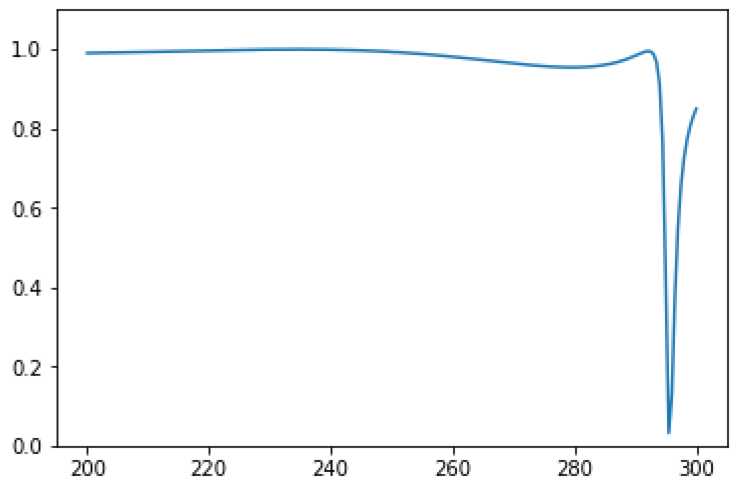
```
Test 82
True spectrum:
```



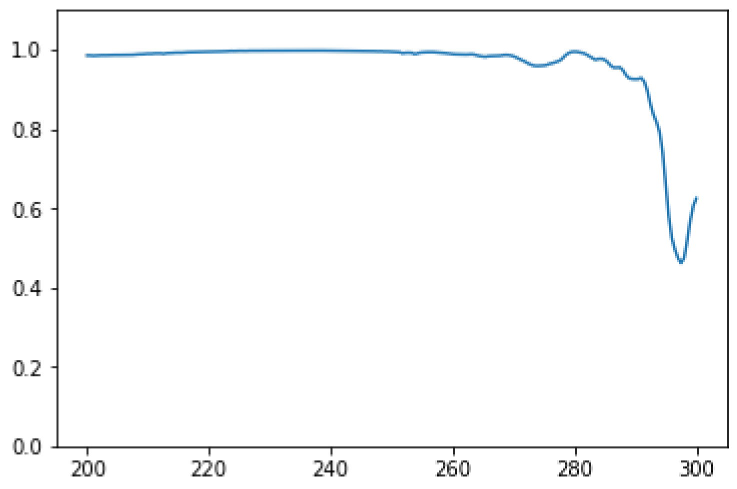
```
Predicted spectrum:
```



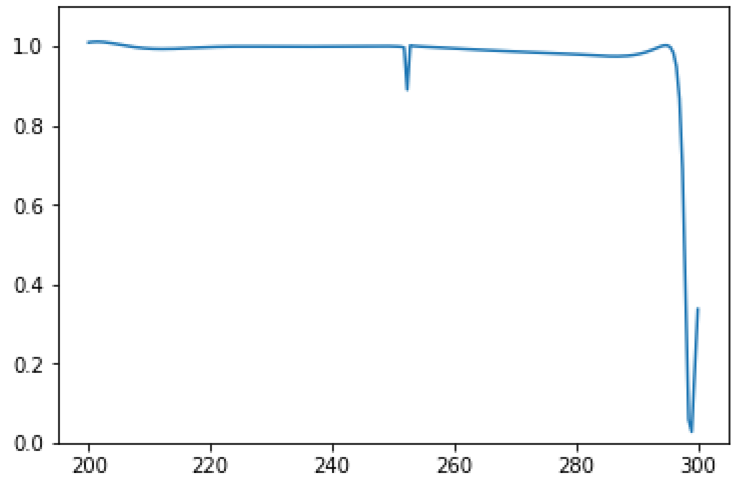
```
Test 83
True spectrum:
```



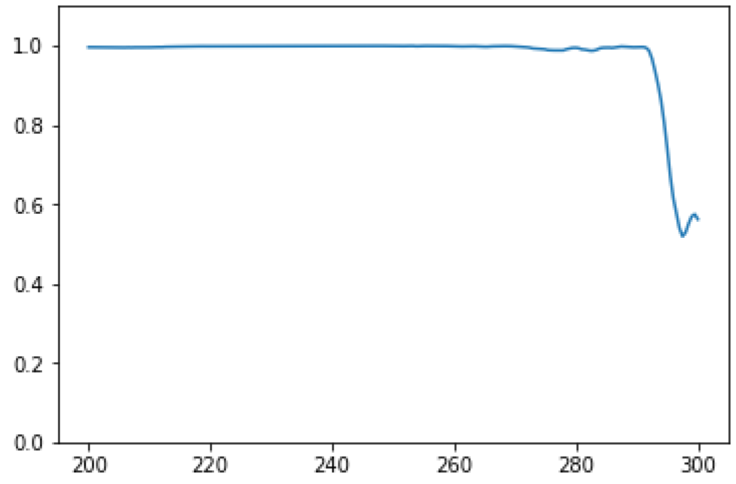
```
Predicted spectrum:
```



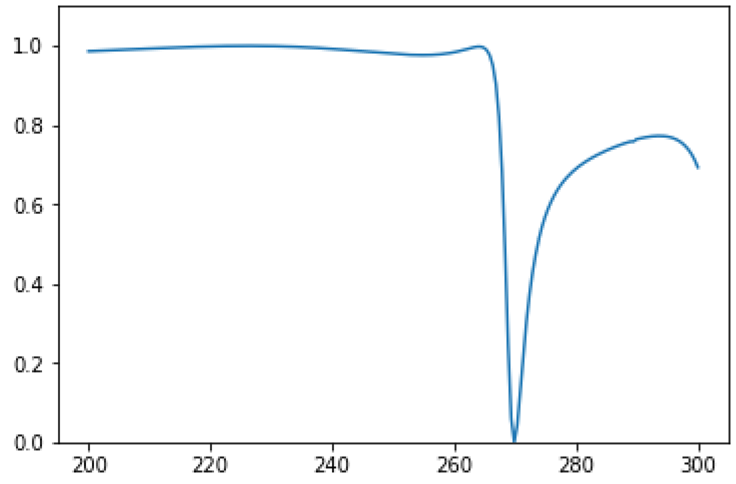
```
Test 84
True spectrum:
```



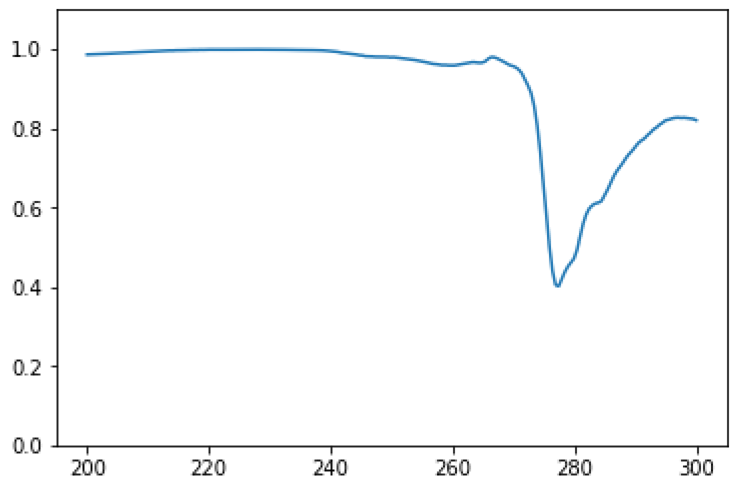
```
Predicted spectrum:
```



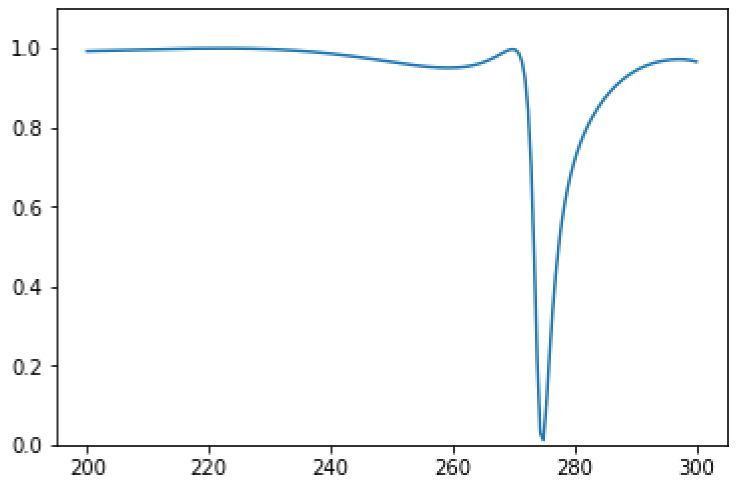
```
Test 85
True spectrum:
```



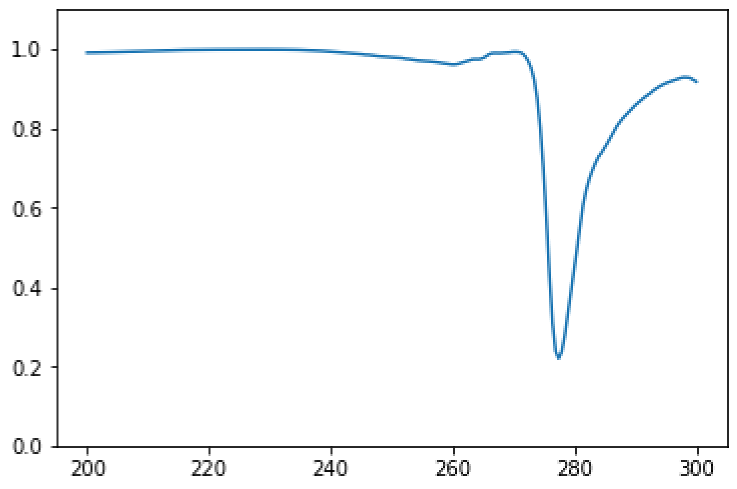
```
Predicted spectrum:
```



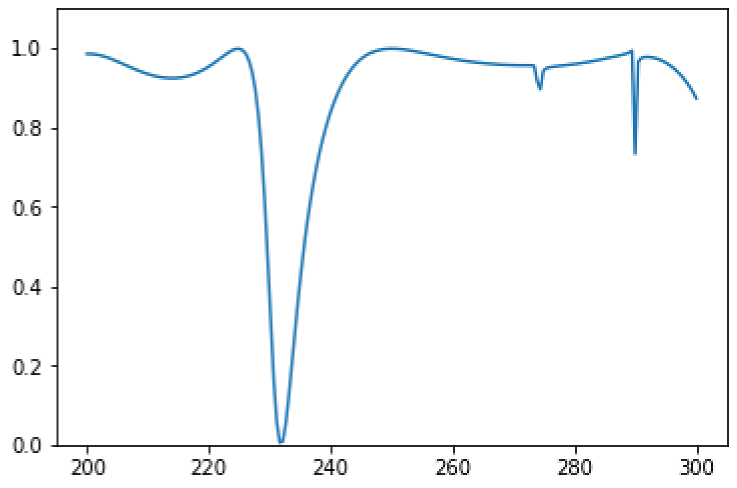
```
Test 86
True spectrum:
```



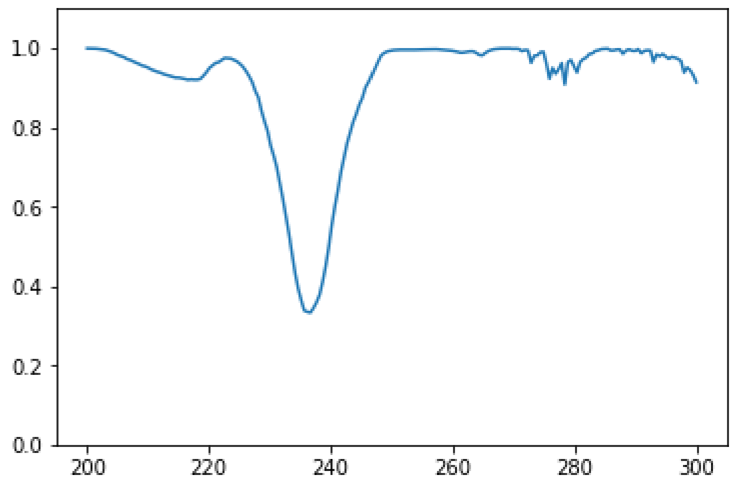
```
Predicted spectrum:
```



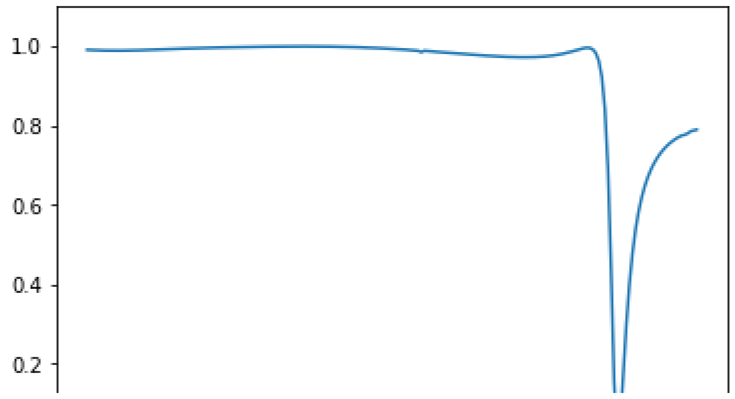
```
Test 87
True spectrum:
```



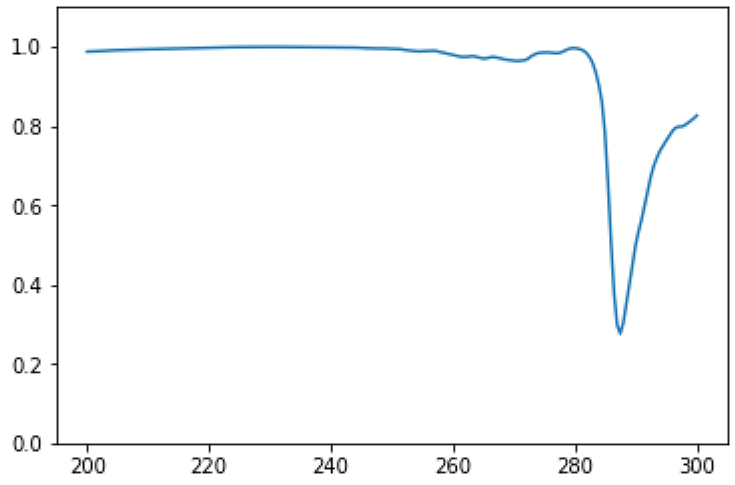
```
Predicted spectrum:
```



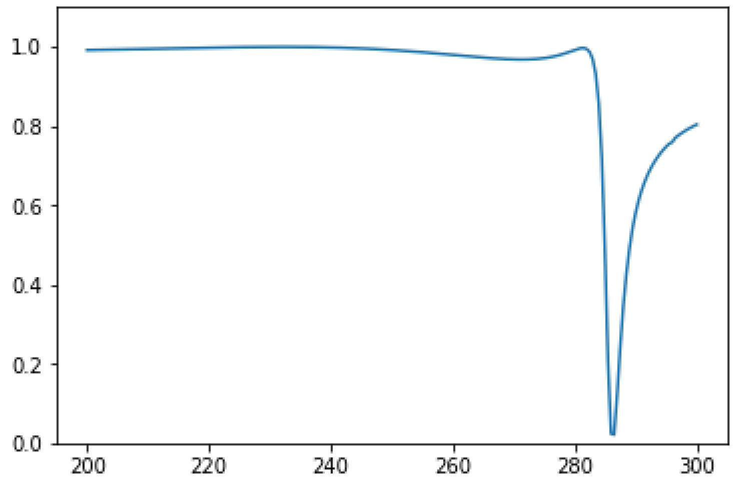
```
Test 88
True spectrum:
```



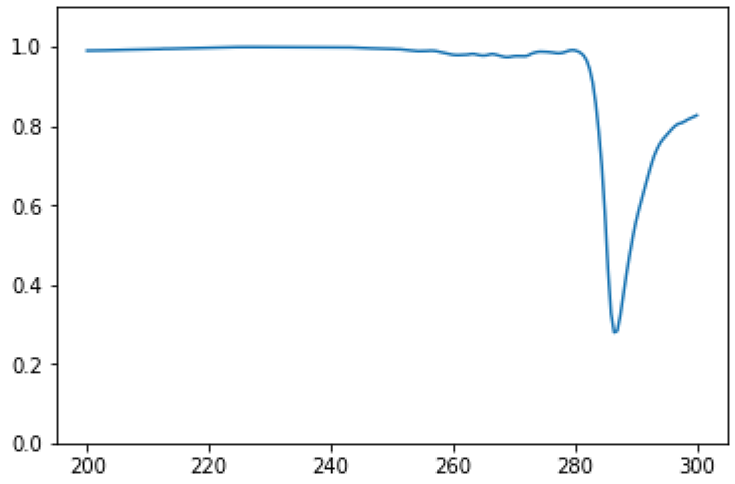
```
Predicted spectrum:
```



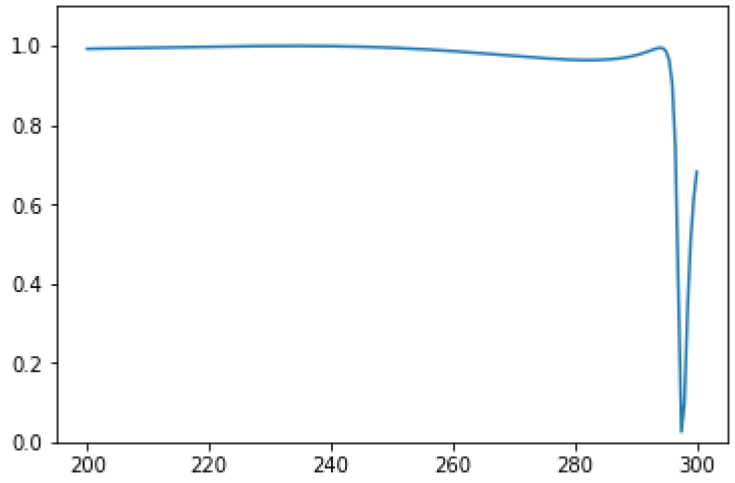
```
Test 89
True spectrum:
```



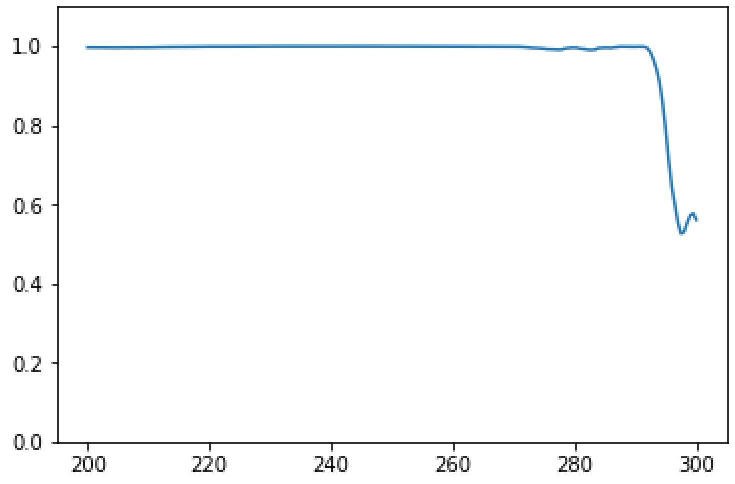
```
Predicted spectrum:
```



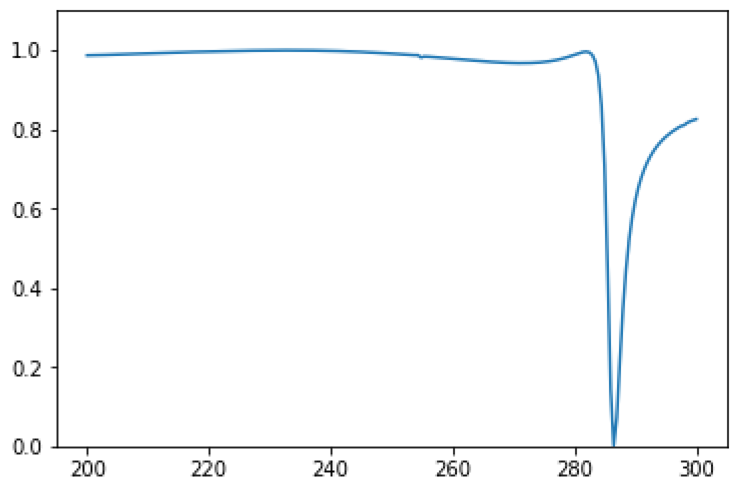
```
Test 90
True spectrum:
```



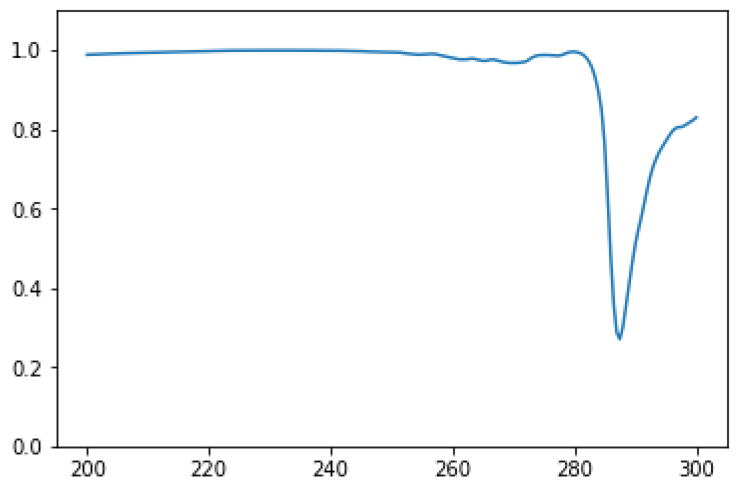
```
Predicted spectrum:
```



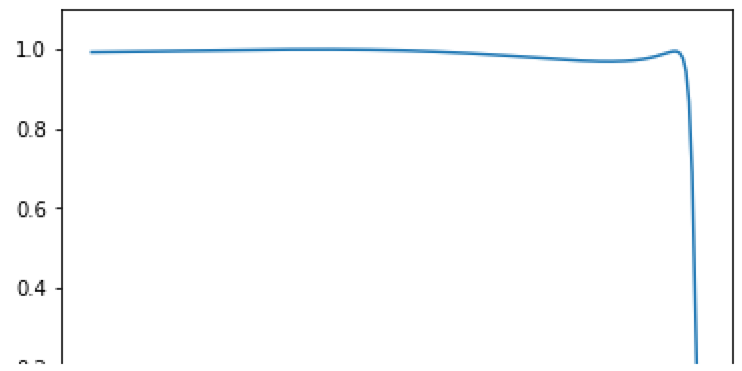
```
Test 91
True spectrum:
```



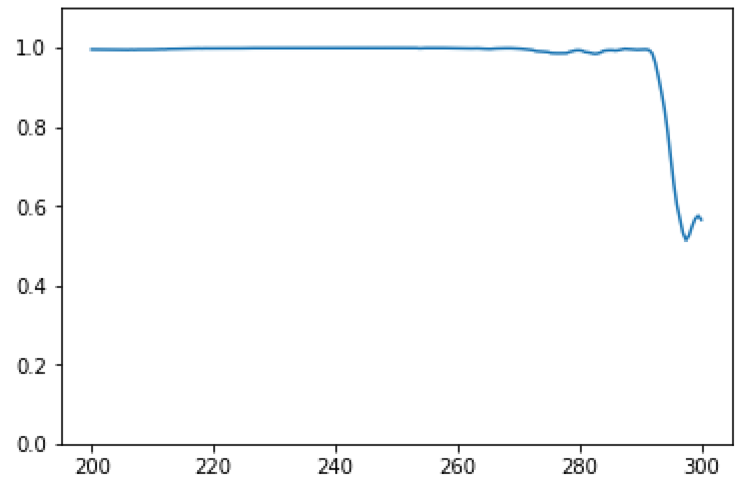
```
Predicted spectrum:
```



```
Test 92
True spectrum:
```



```
Predicted spectrum:
```



In [18]:
```
for i,layer in enumerate(simulator.layers):
    print(i,layer.name)
```

```
0 dense_19
1 dropout_16
2 dense_20
3 dropout_17
4 dense_21
5 dropout_18
6 dense_22
7 dropout_19
8 dense_23
9 dropout_20
10 dense_24
```

In [19]:
```
for i,layer in enumerate(simulator.layers):
    print(i,layer.name,layer.trainable)
```

```
0 dense_19 True
1 dropout_16 True
2 dense_20 True
3 dropout_17 True
4 dense_21 True
5 dropout_18 True
6 dense_22 True
7 dropout_19 True
8 dense_23 True
9 dropout_20 True
10 dense_24 True
```

In [20]:
```
for layer in simulator.layers:
    layer.trainable=False
```

In [47]:
```
tandem = Sequential()
tandem.add(Dense(512, activation='relu', input_dim=out_dim))
tandem.add(Dropout(0.5))
tandem.add(Dense(256, activation='relu'))
tandem.add(Dropout(0.5))
# tandem.add(Dense(200, activation='relu', input_dim=out_dim))
# tandem.add(Dropout(0.5))
# tandem.add(Dense(200, activation='relu'))
# tandem.add(Dropout(0.5))
# tandem.add(Dense(100, activation='relu'))
# tandem.add(Dropout(0.5))
# tandem.add(Dense(40, activation='relu'))
# tandem.add(Dropout(0.2))
tandem.add(Dense(6, activation='sigmoid'))
for layer in simulator.layers:
    tandem.add(layer)
tandem.compile(loss=keras.losses.mean_squared_error,
                 optimizer=keras.optimizers.Adam(lr = 0.001))
```

In [48]:
```
for i,layer in enumerate(tandem.layers):
    print(i,layer.name,layer.trainable)
```

```
0 dense_39 True
1 dropout_31 True
2 dense_40 True
3 dropout_32 True
4 dense_41 True
5 dense_19 False
6 dropout_16 False
7 dense_20 False
8 dropout_17 False
9 dense_21 False
10 dropout_18 False
11 dense_22 False
12 dropout_19 False
13 dense_23 False
14 dropout_20 False
15 dense_24 False
```

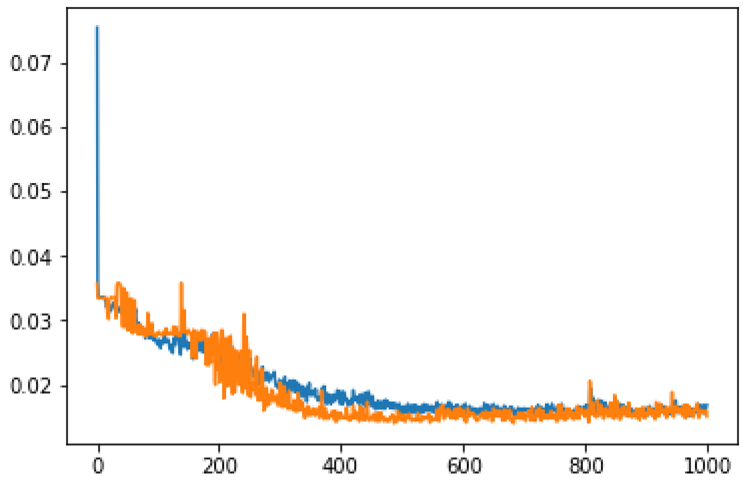
In [49]:
```
tandem.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
dense_39 (Dense)             (None, 512)               102912    
_________________________________________________________________
dropout_31 (Dropout)         (None, 512)               0         
_________________________________________________________________
dense_40 (Dense)             (None, 256)               131328    
_________________________________________________________________
dropout_32 (Dropout)         (None, 256)               0         
_________________________________________________________________
dense_41 (Dense)             (None, 6)                 1542      
_________________________________________________________________
dense_19 (Dense)             (None, 20)                140       
_________________________________________________________________
dropout_16 (Dropout)         (None, 20)                0         
_________________________________________________________________
dense_20 (Dense)             (None, 500)               10500     
_________________________________________________________________
dropout_17 (Dropout)         (None, 500)               0         
_________________________________________________________________
dense_21 (Dense)             (None, 500)               250500    
_________________________________________________________________
dropout_18 (Dropout)         (None, 500)               0         
_________________________________________________________________
dense_22 (Dense)             (None, 200)               100200    
_________________________________________________________________
dropout_19 (Dropout)         (None, 200)               0         
_________________________________________________________________
dense_23 (Dense)             (None, 200)               40200     
_________________________________________________________________
dropout_20 (Dropout)         (None, 200)               0         
_________________________________________________________________
dense_24 (Dense)             (None, 200)               40200     
=================================================================
Total params: 677,522
Trainable params: 235,782
Non-trainable params: 441,740
_________________________________________________________________
```

In [*]:

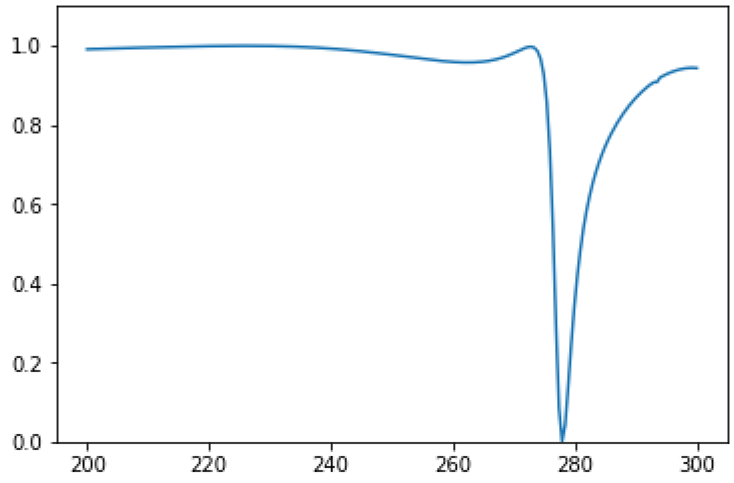
```
history = tandem.fit(train_Y, train_Y,
                    epochs=1000,
                    batch_size=100,
                    validation_data=(test_Y, test_Y),
                    verbose=2)

train_score = tandem.evaluate(train_Y, train_Y, batch_size=20)
test_score = tandem.evaluate(test_Y, test_Y, batch_size= 50)
print(train_score)
print(test_score)
plt.plot(history.history['loss'], label='train')
plt.plot(history.history['val_loss'], label='test')
```

```
 - 0s - loss: 0.0264 - val_loss: 0.0279
Epoch 237/1000
 - 0s - loss: 0.0268 - val_loss: 0.0285
Epoch 238/1000
 - 0s - loss: 0.0271 - val_loss: 0.0286
Epoch 239/1000
 - 0s - loss: 0.0263 - val_loss: 0.0284
Epoch 240/1000
 - 0s - loss: 0.0261 - val_loss: 0.0284
Epoch 241/1000
 - 0s - loss: 0.0255 - val_loss: 0.0289
Epoch 242/1000
 - 0s - loss: 0.0251 - val_loss: 0.0261
Epoch 243/1000
 - 0s - loss: 0.0255 - val_loss: 0.0255
Epoch 244/1000
 - 0s - loss: 0.0251 - val_loss: 0.0277
Epoch 245/1000
 - 0s - loss: 0.0247 - val_loss: 0.0285
Epoch 246/1000
```

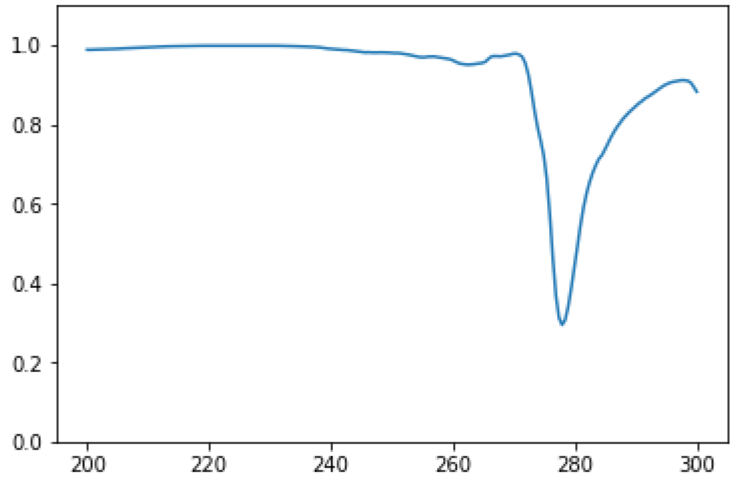


In [41]:
```
x = np.genfromtxt('meep_code/data/SP_xaxis.txt')
for i in range(len(test_Y)):
    print('Test '+str(i))
    print('True spectrum: ')
    plt.ylim(0, 1.1)
    plt.plot(x, test_Y[i])
    plt.show()
    print('Predicted spectrum: ')
    plt.ylim(0, 1.1)
    plt.plot(x, np.reshape(tandem.predict(np.reshape(test_Y[i], (1, 200))), (
    plt.show()
```

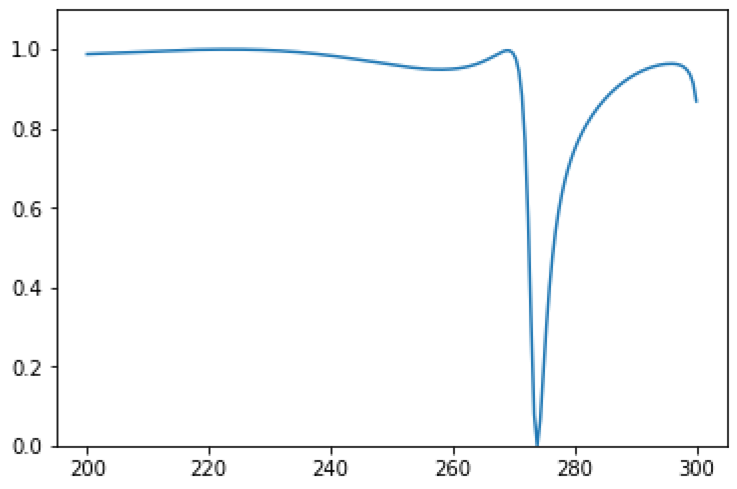
```
Test 0
True spectrum:
```



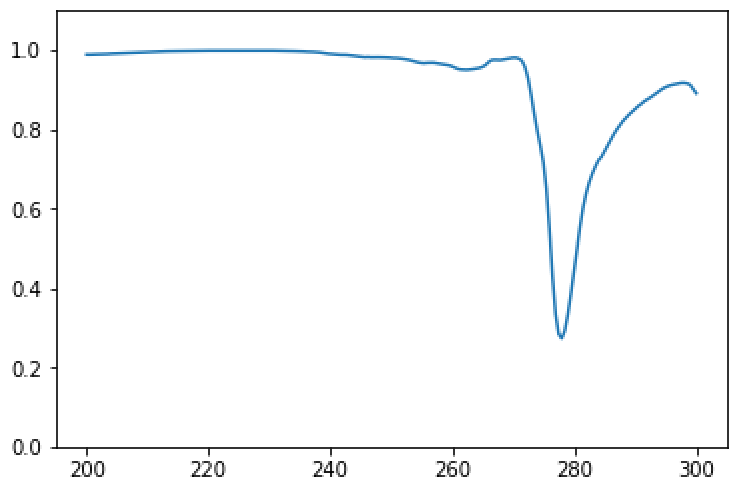
```
Predicted spectrum:
```



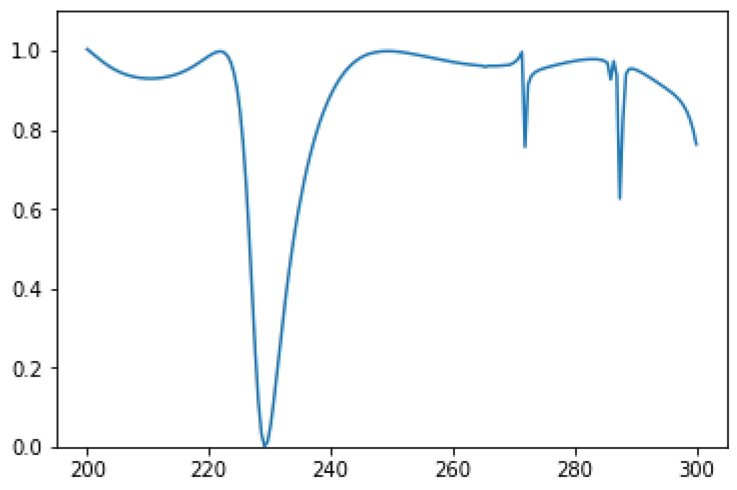
```
Test 1
True spectrum:
```



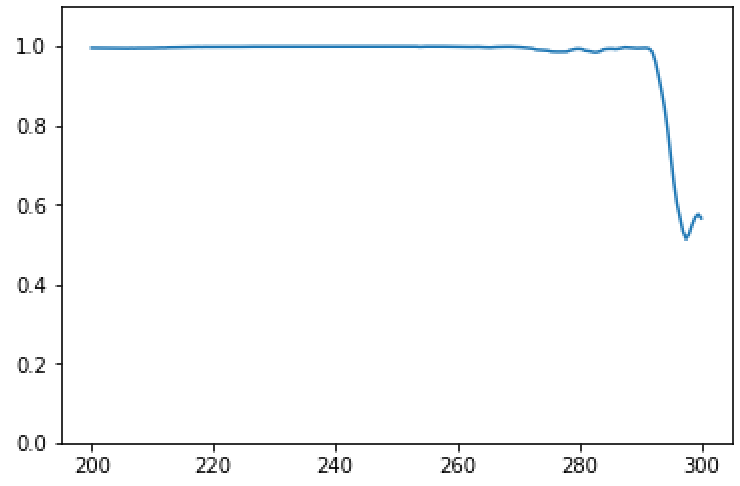
```
Predicted spectrum:
```



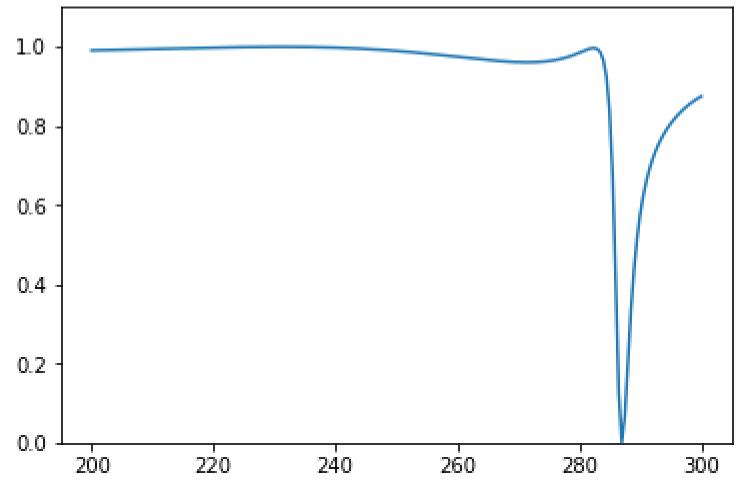
```
Test 2
True spectrum:
```



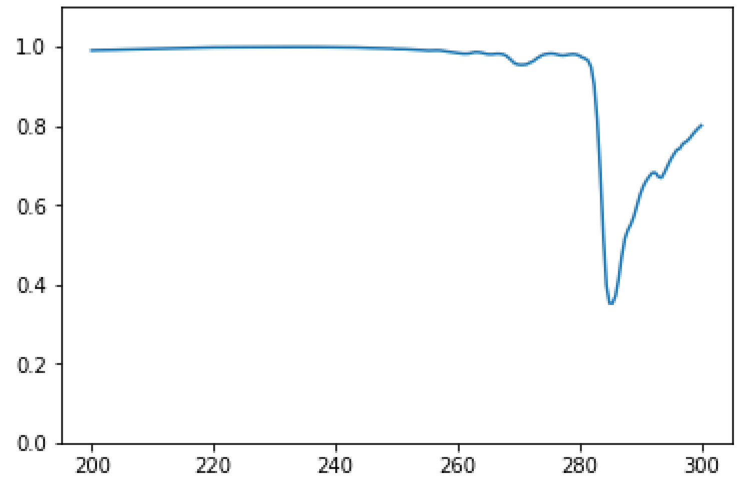
```
Predicted spectrum:
```



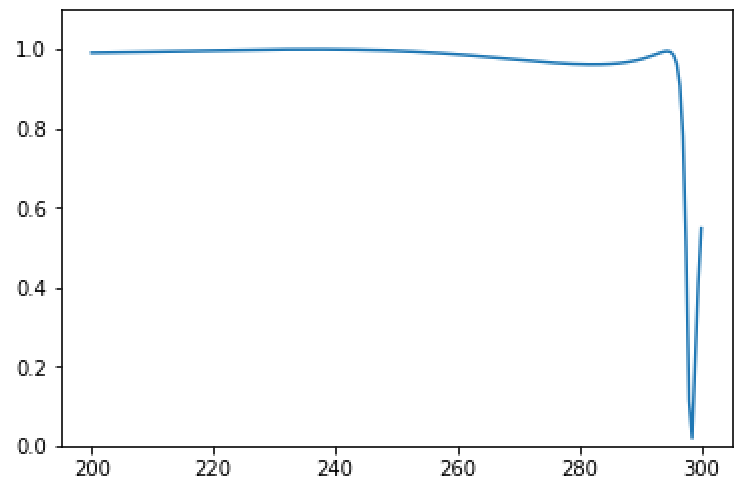
```
Test 3
True spectrum:
```



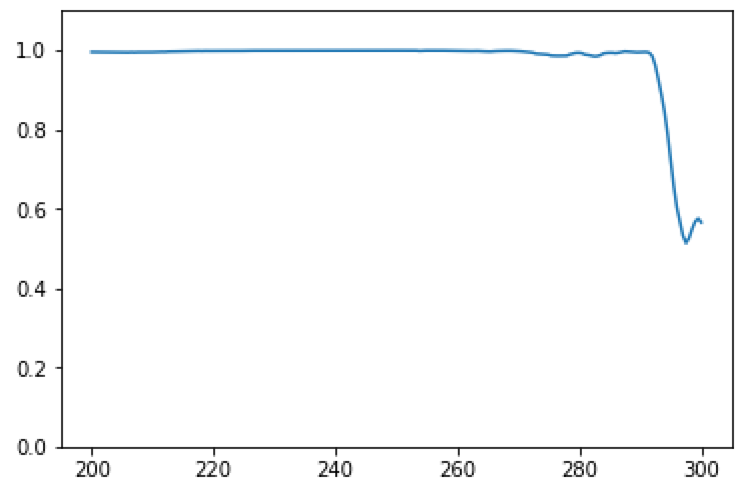
```
Predicted spectrum:
```



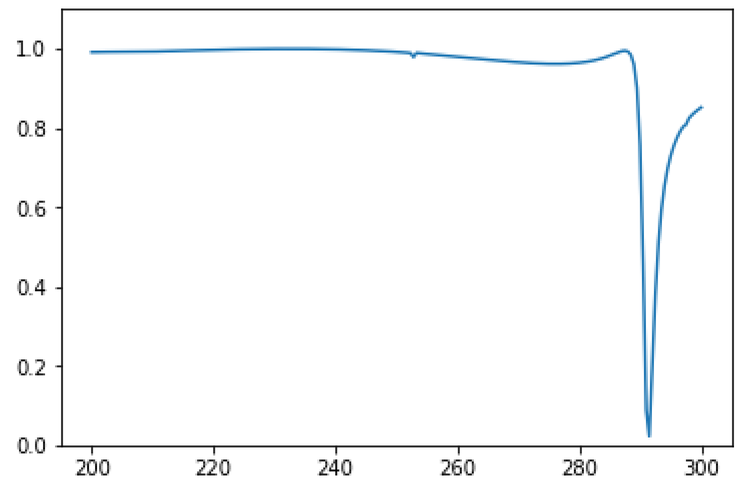
```
Test 4
True spectrum:
```



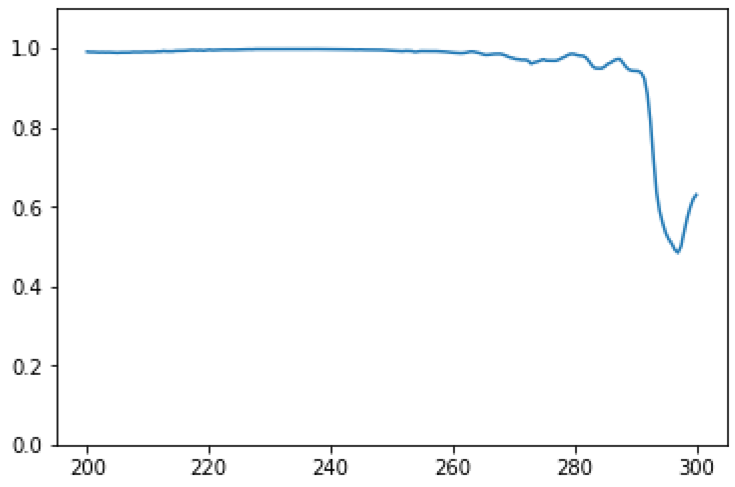
```
Predicted spectrum:
```



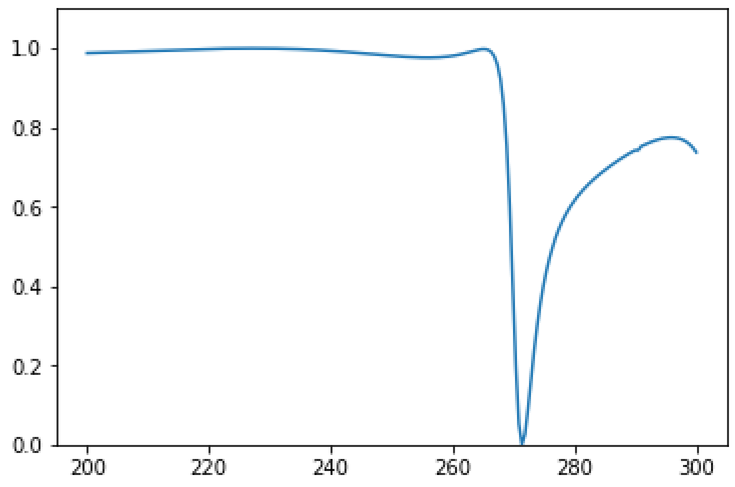
```
Test 5
True spectrum:
```



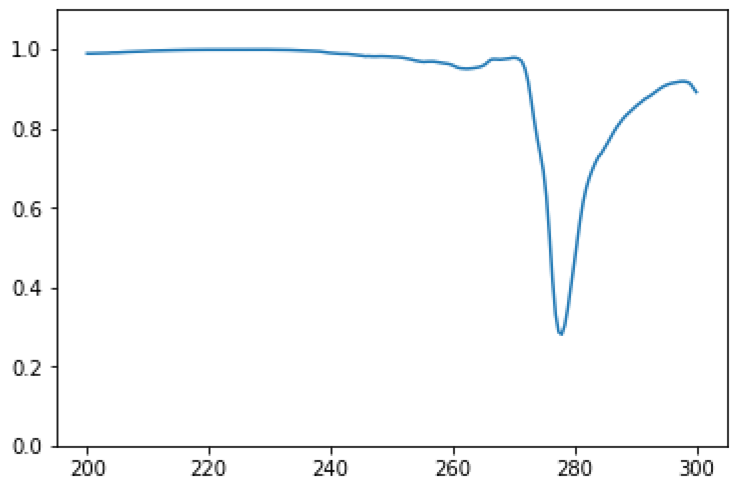
```
Predicted spectrum:
```



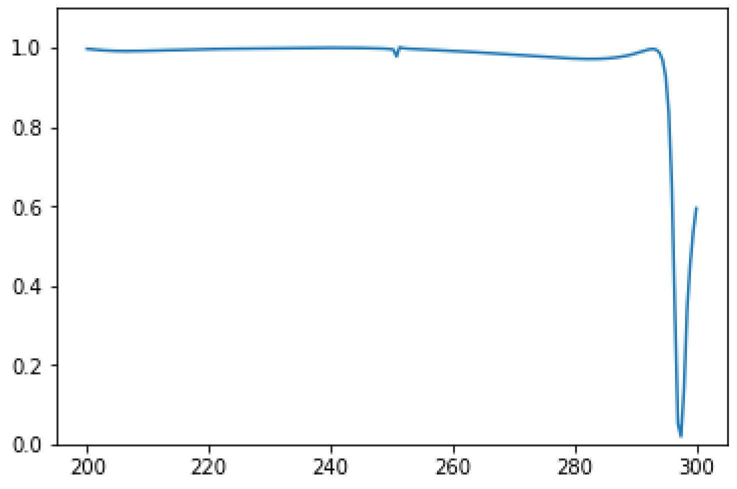
```
Test 6
True spectrum:
```



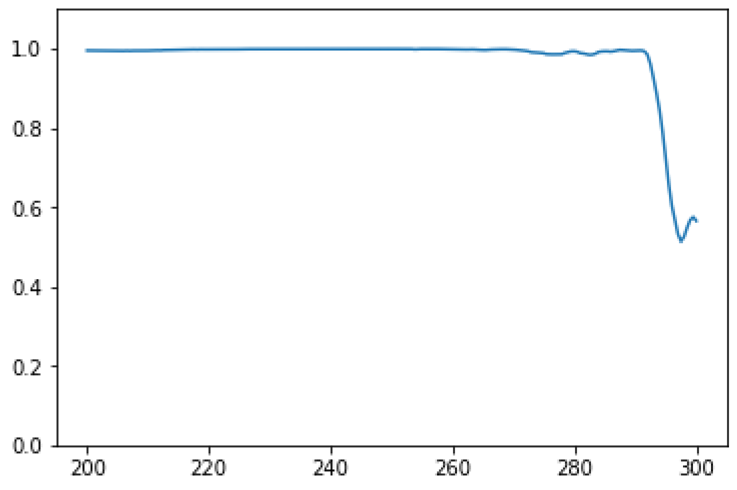
```
Predicted spectrum:
```



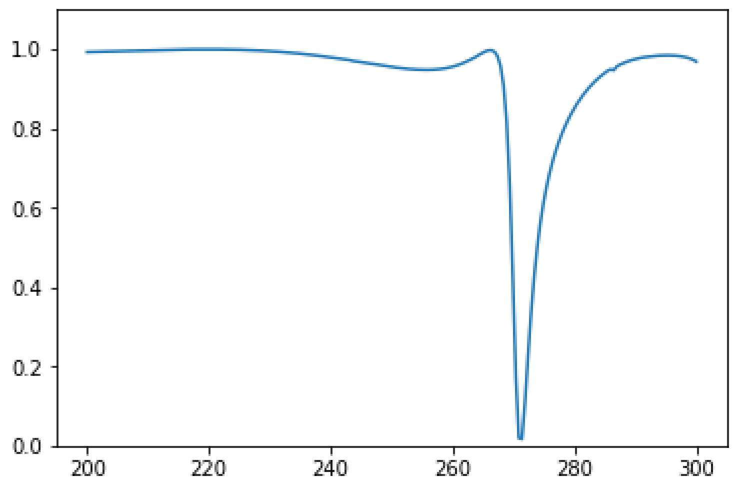
```
Test 7
True spectrum:
```



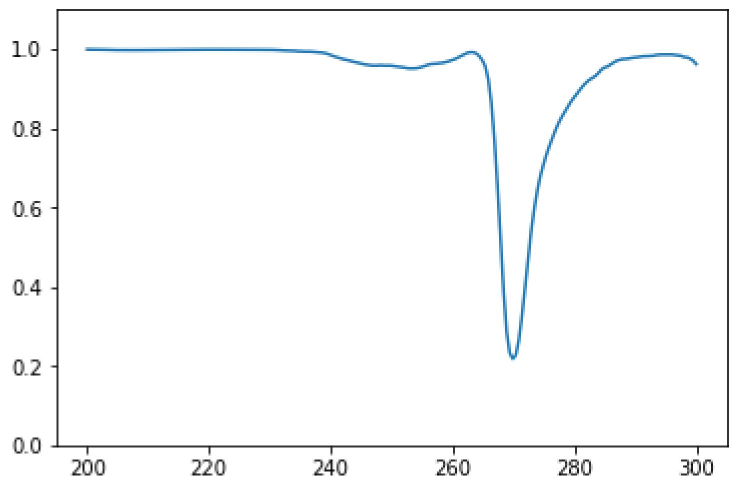
```
Predicted spectrum:
```



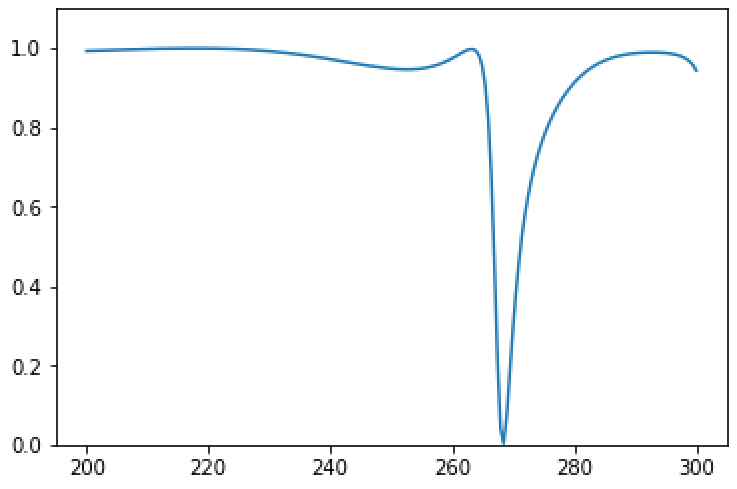
```
Test 8
True spectrum:
```



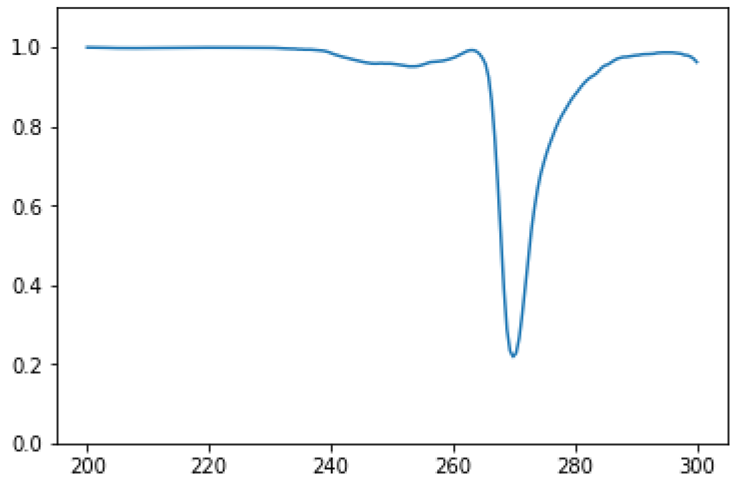
```
Predicted spectrum:
```



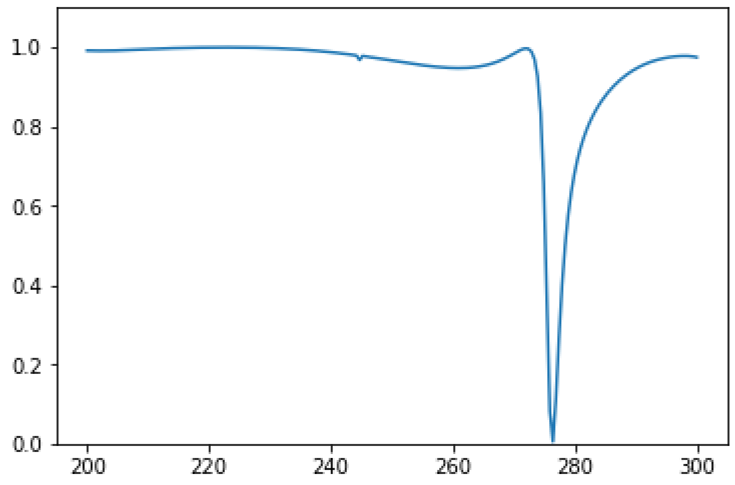
```
Test 9
True spectrum:
```



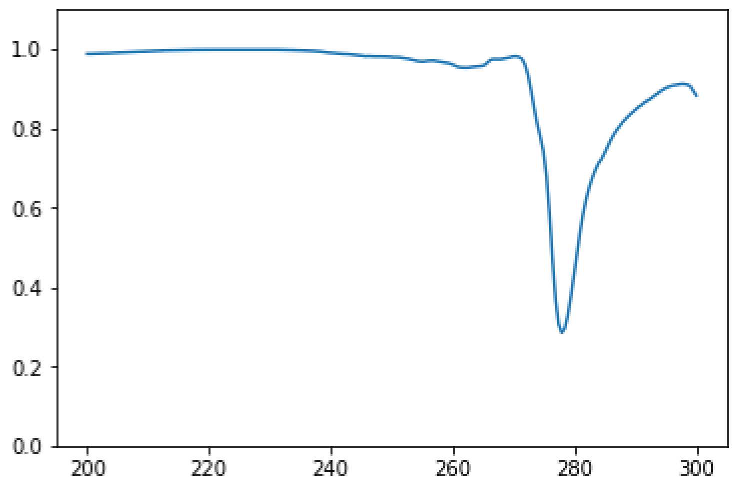
```
Predicted spectrum:
```



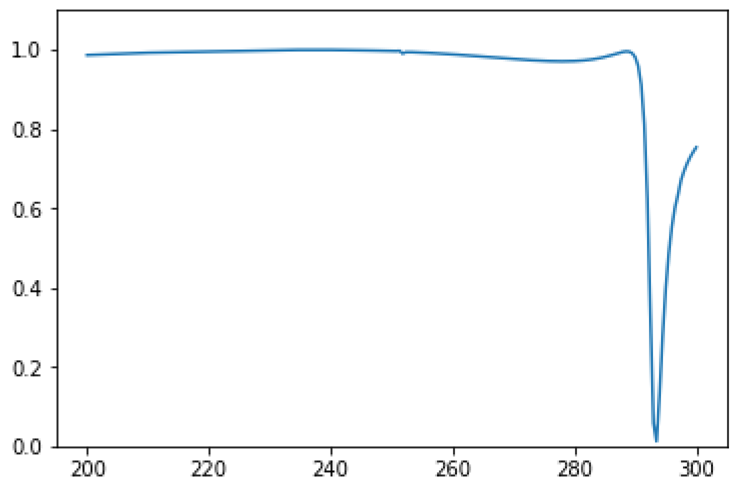
```
Test 10
True spectrum:
```



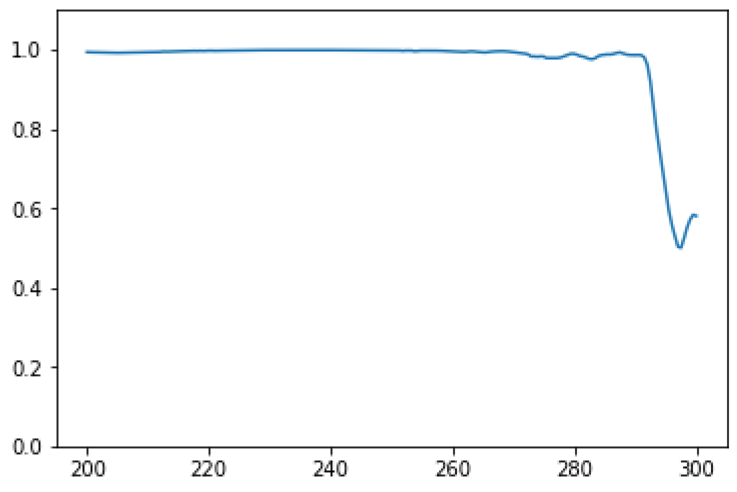
```
Predicted spectrum:
```



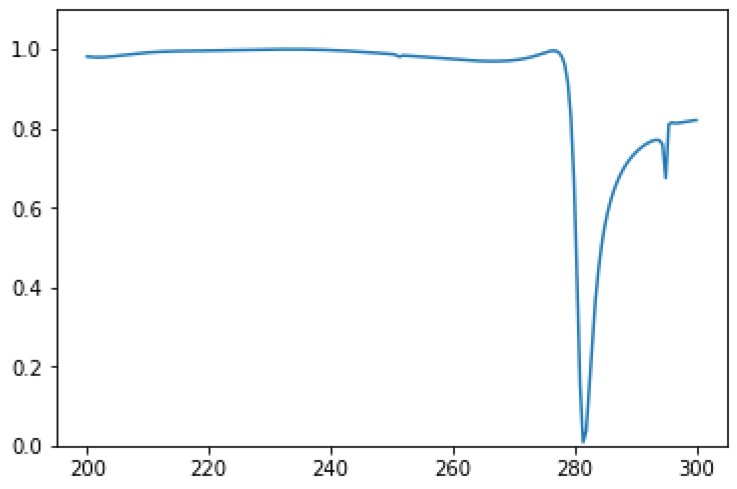
```
Test 11
True spectrum:
```



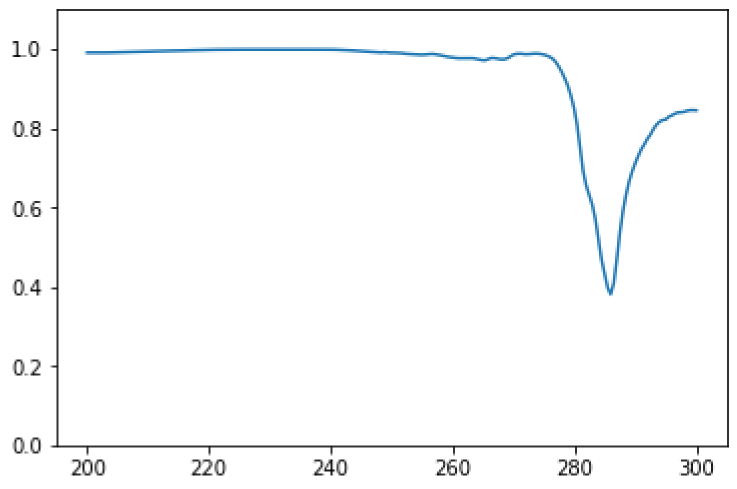
```
Predicted spectrum:
```



```
Test 12
True spectrum:
```

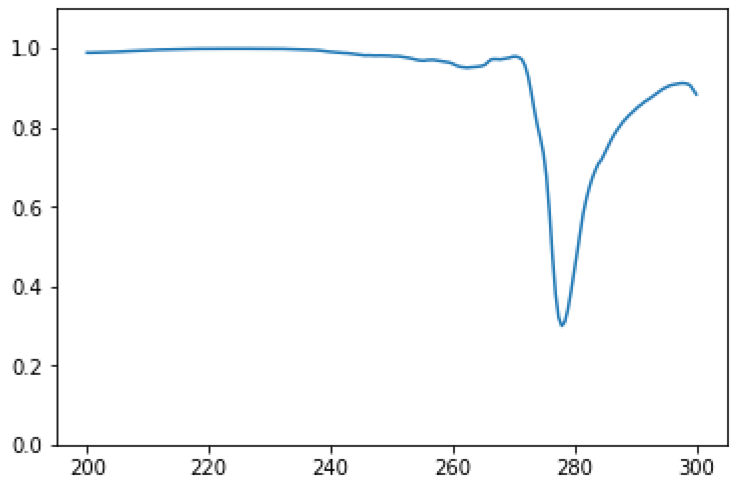


```
Predicted spectrum:
```

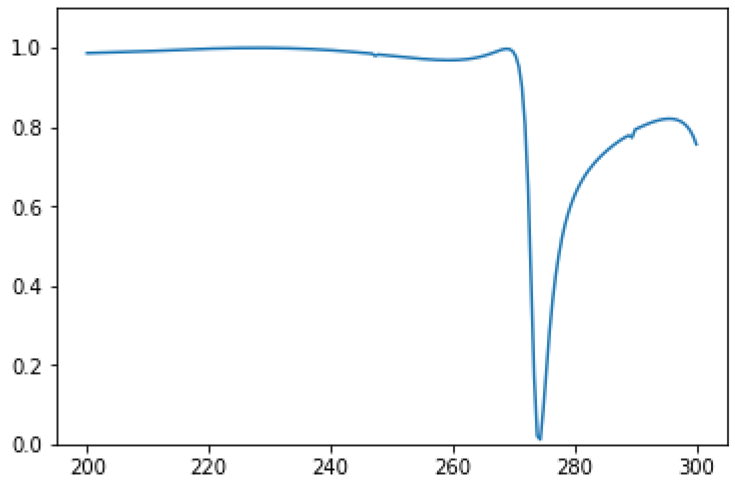


```
Test 13
True spectrum:
```

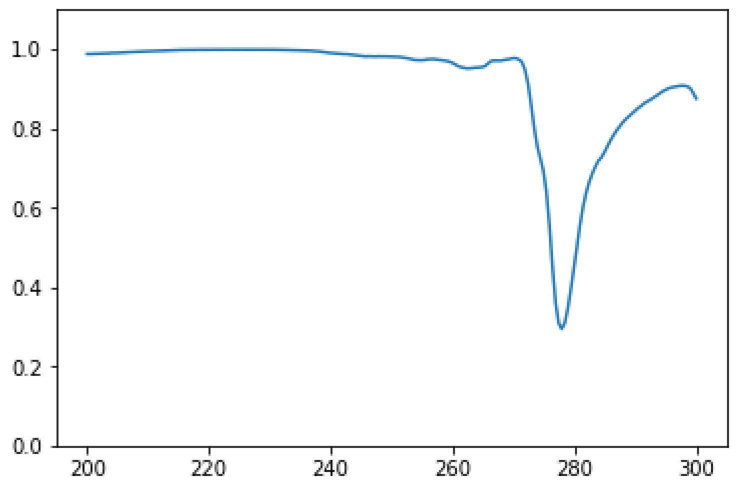
```
Predicted spectrum:
```



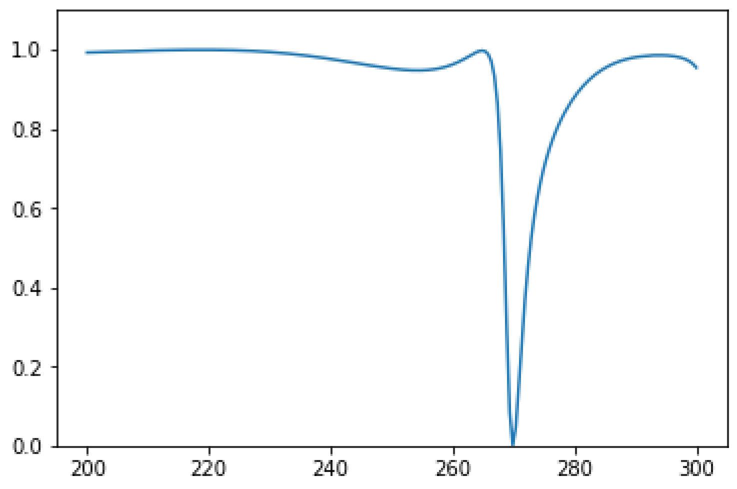
```
Test 14
True spectrum:
```



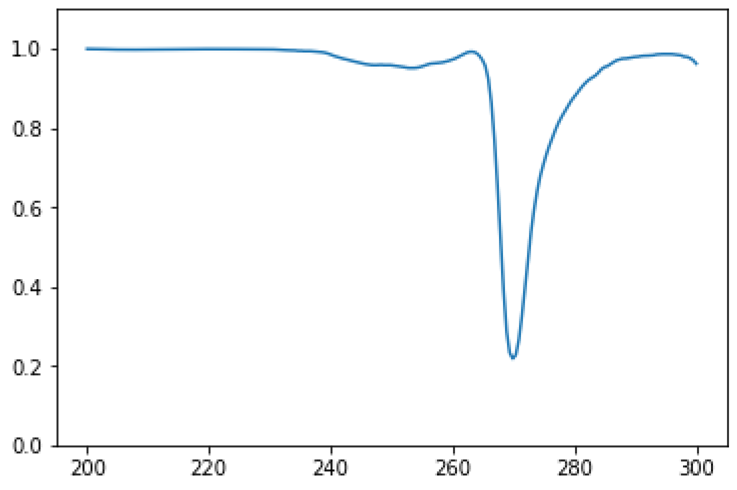
```
Predicted spectrum:
```



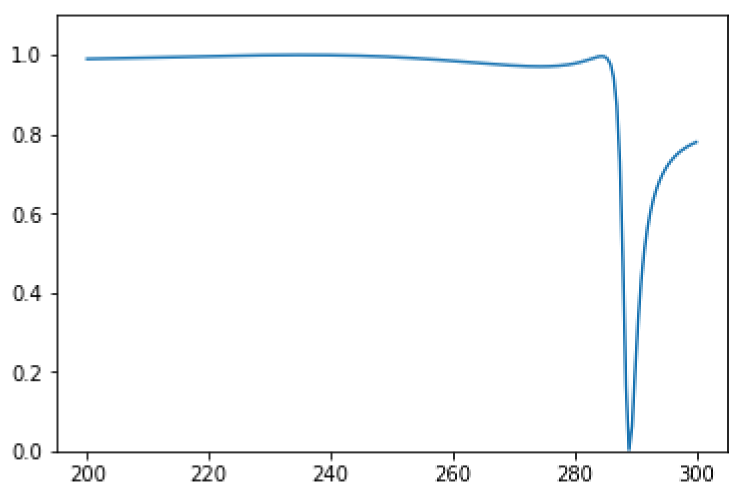
```
Test 15
True spectrum:
```



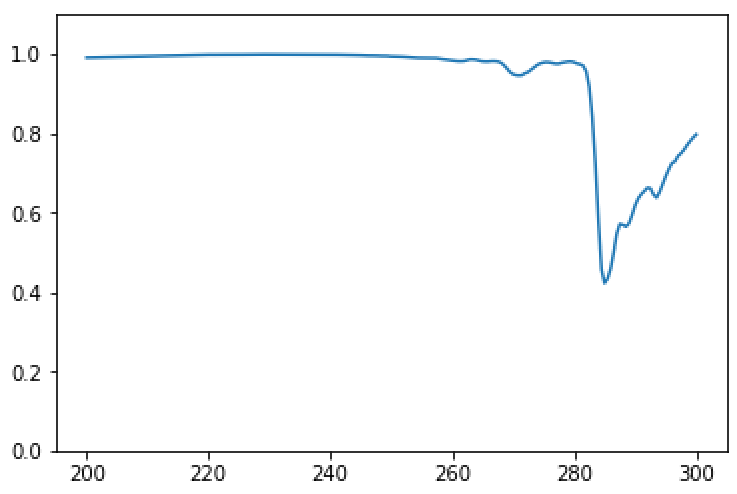
```
Predicted spectrum:
```



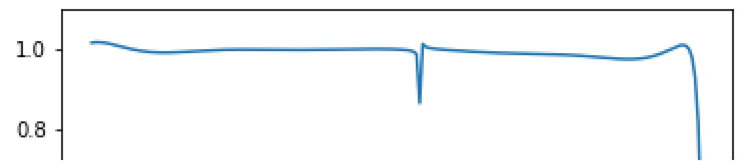
```
Test 16
True spectrum:
```



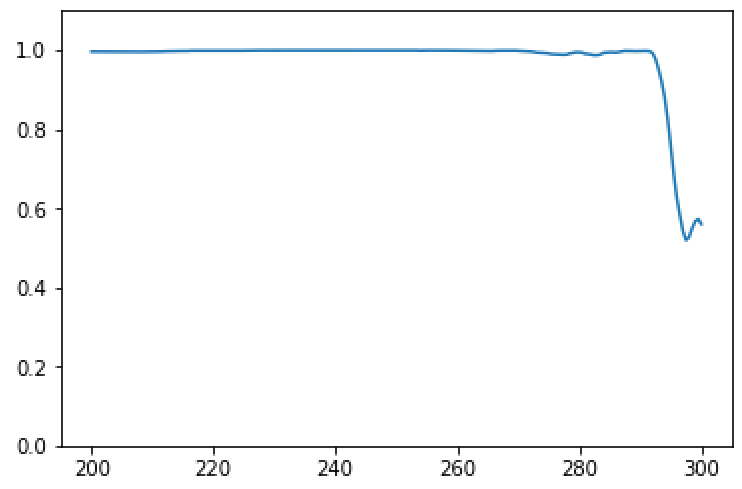
```
Predicted spectrum:
```



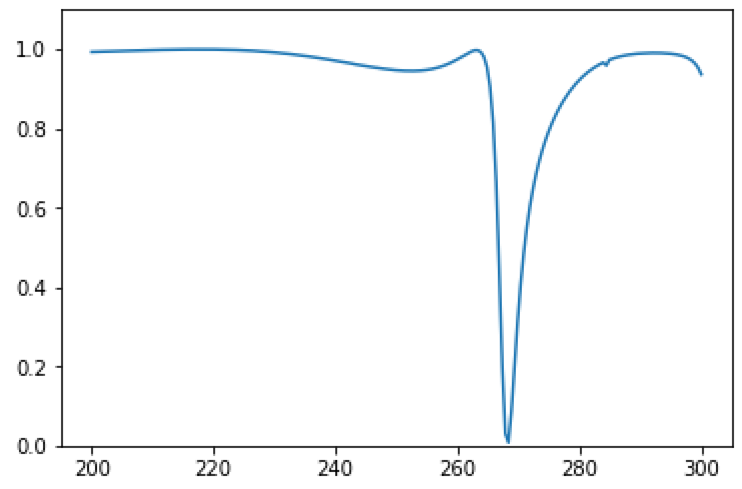
```
Test 17
True spectrum:
```



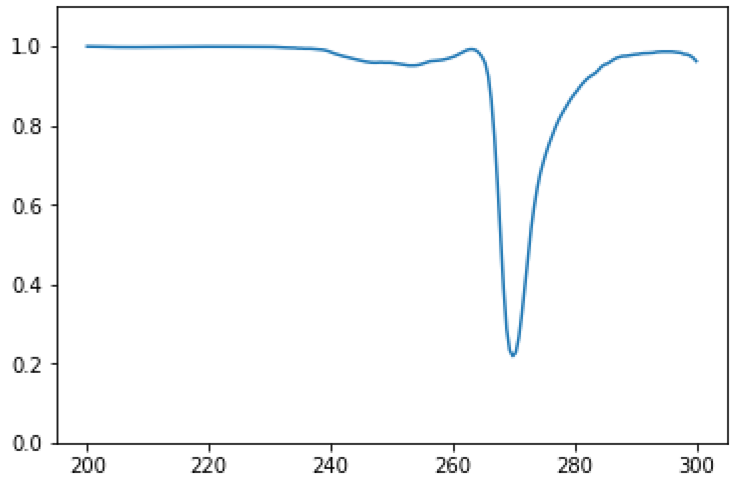
```
Predicted spectrum:
```



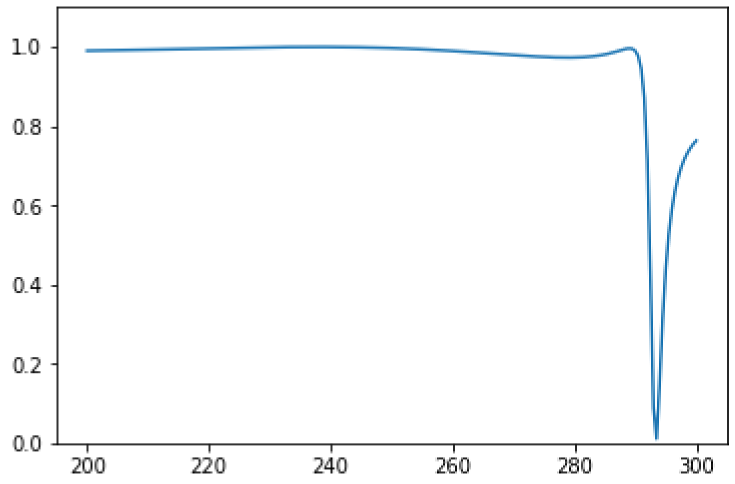
```
Test 18
True spectrum:
```



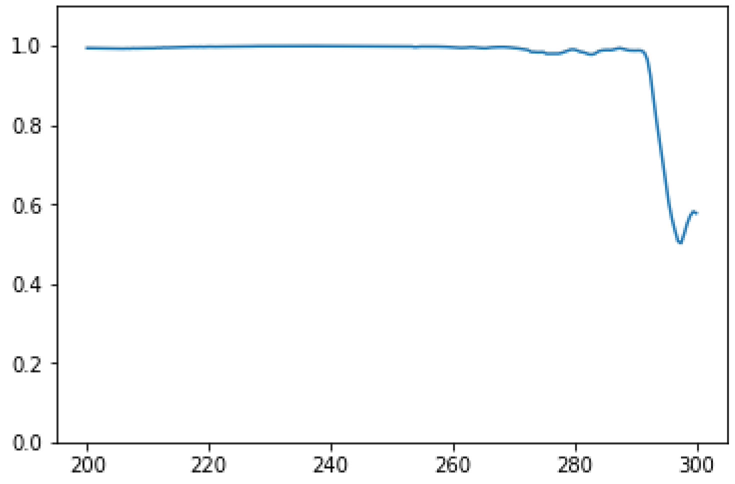
```
Predicted spectrum:
```



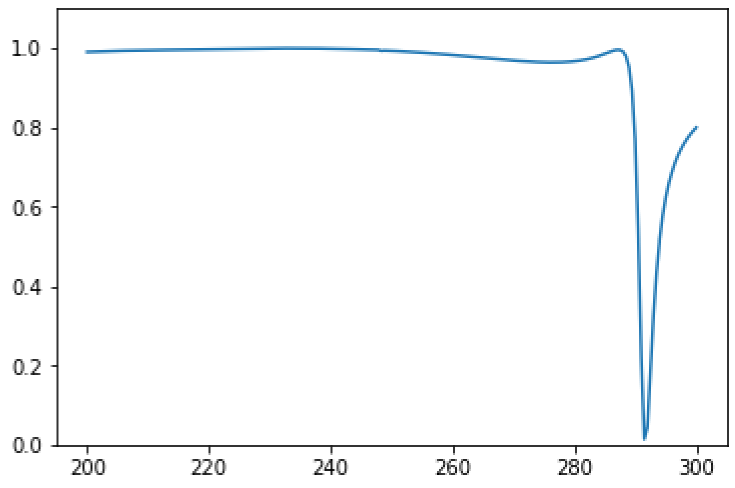
```
Test 19
True spectrum:
```



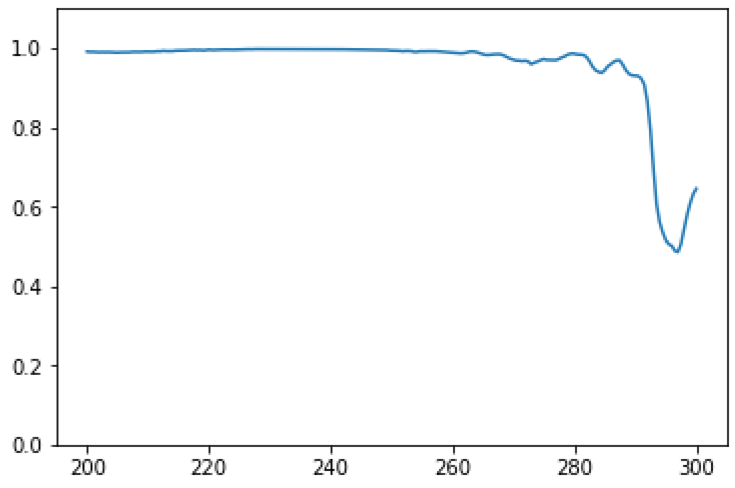
```
Predicted spectrum:
```



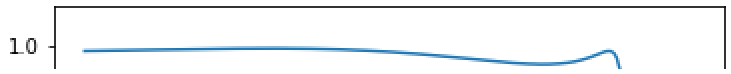
```
Test 20
True spectrum:
```



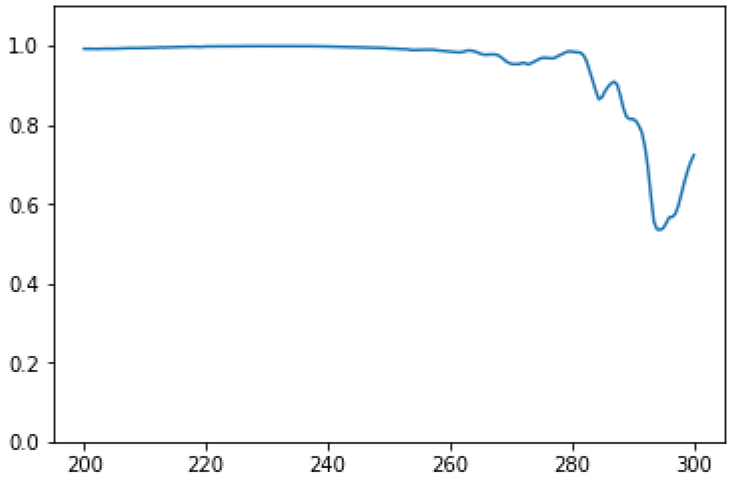
```
Predicted spectrum:
```



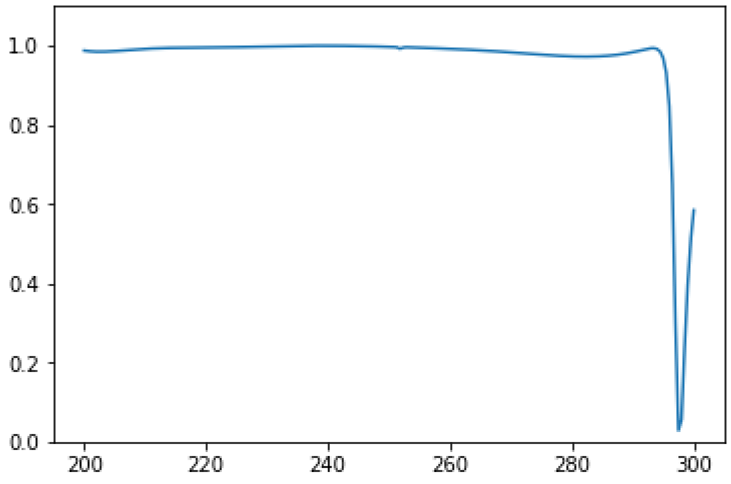
```
Test 21
True spectrum:
```



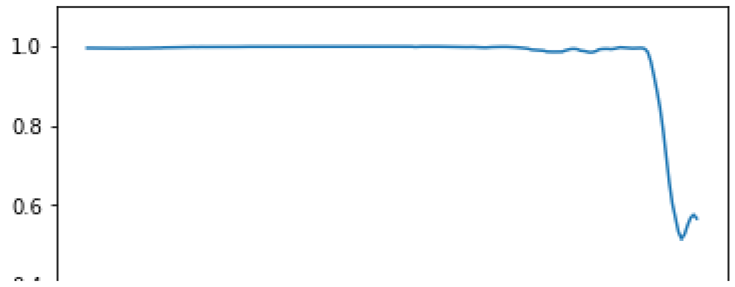
```
Predicted spectrum:
```



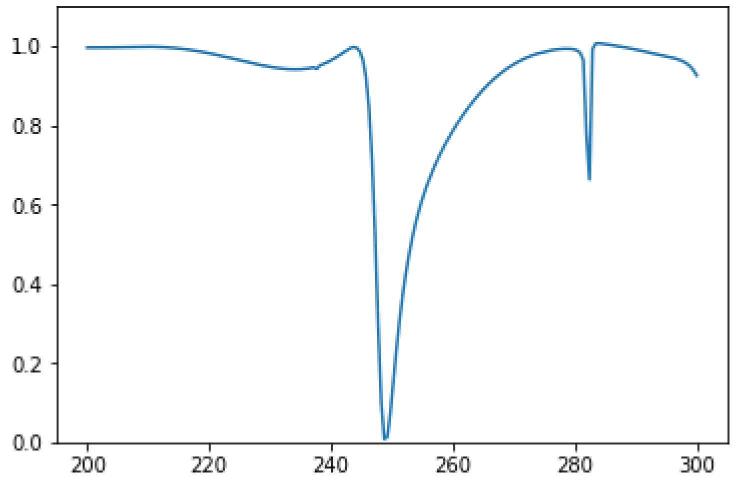
```
Test 22
True spectrum:
```



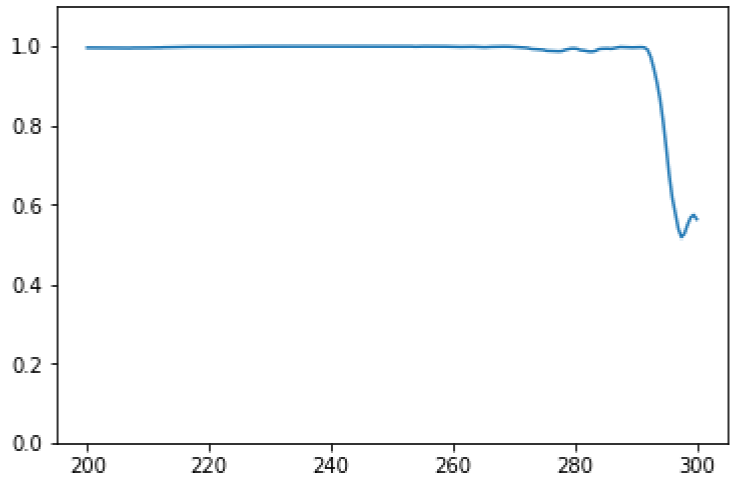
```
Predicted spectrum:
```



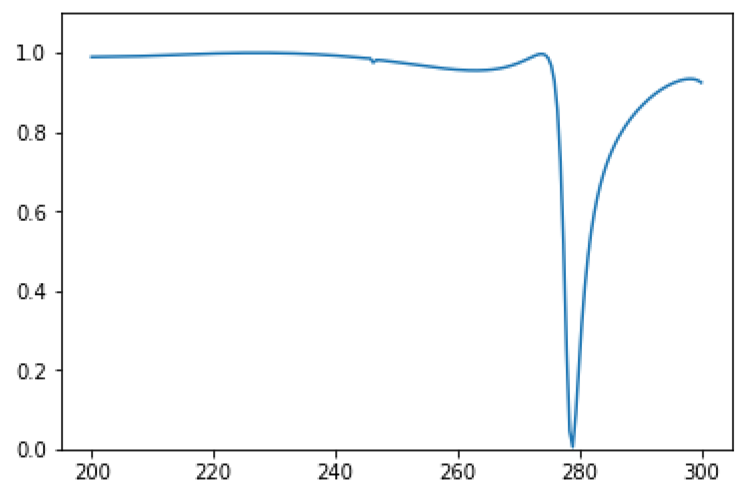
```
Test 23
True spectrum:
```



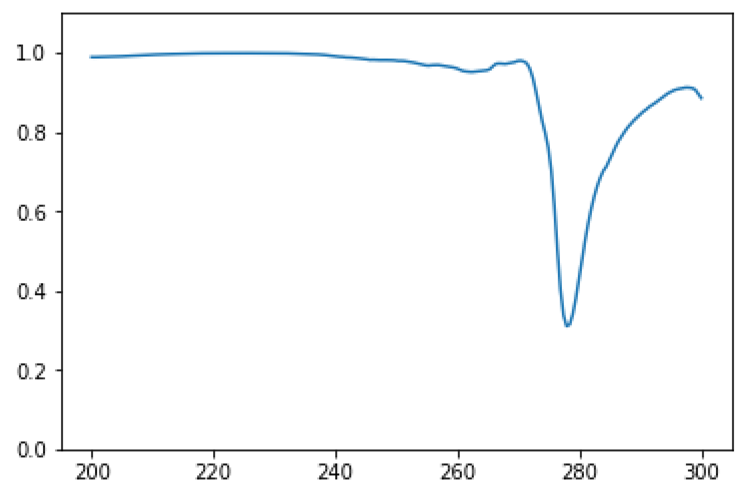
```
Predicted spectrum:
```



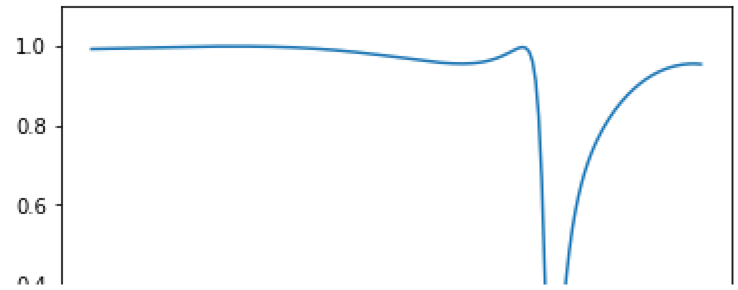
```
Test 24
True spectrum:
```



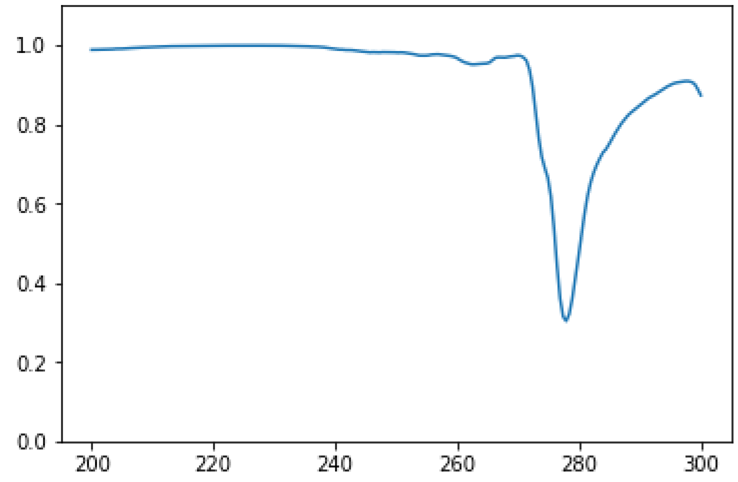
```
Predicted spectrum:
```



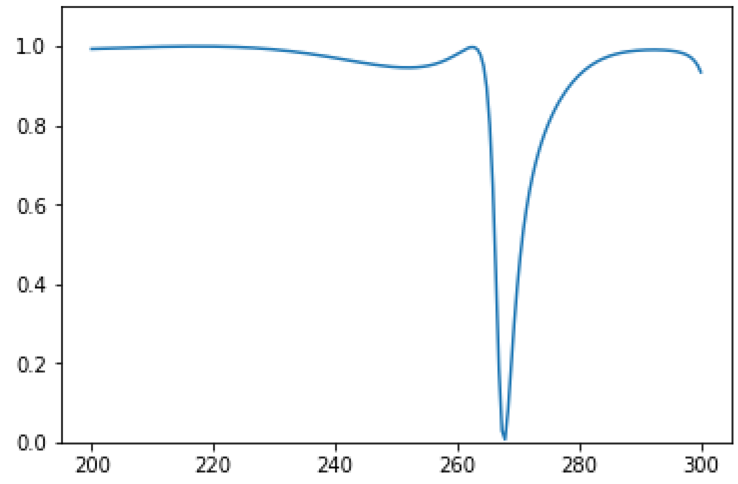
```
Test 25
True spectrum:
```



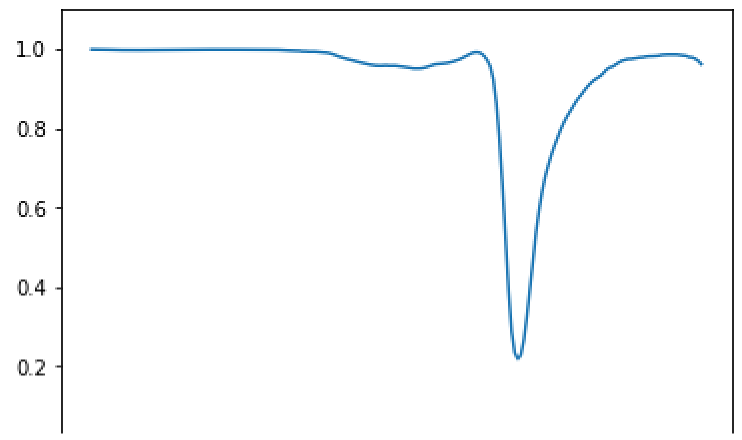
```
Predicted spectrum:
```



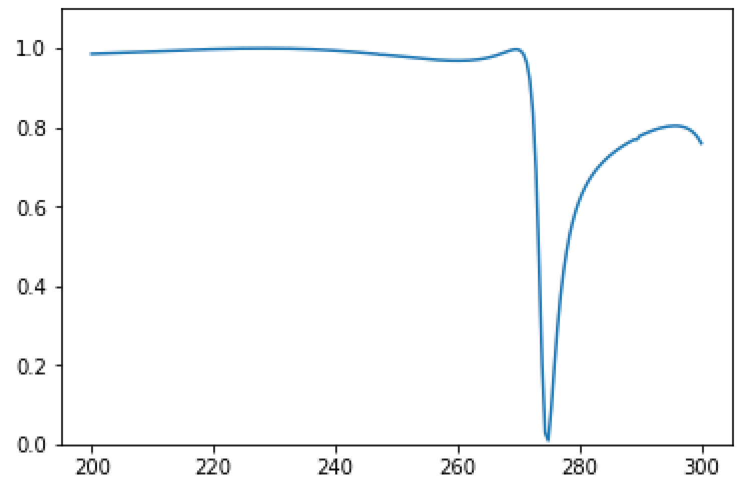
```
Test 26
True spectrum:
```



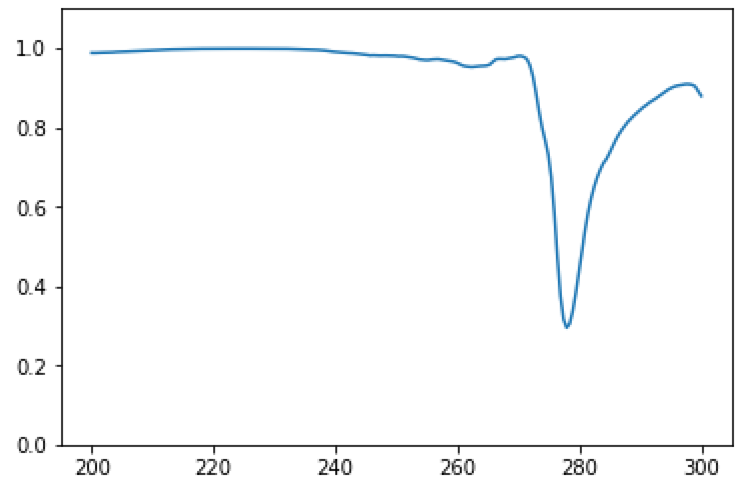
```
Predicted spectrum:
```



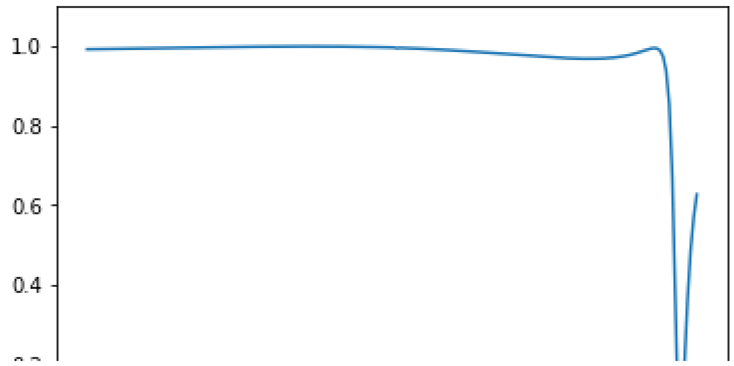
```
Test 27
True spectrum:
```



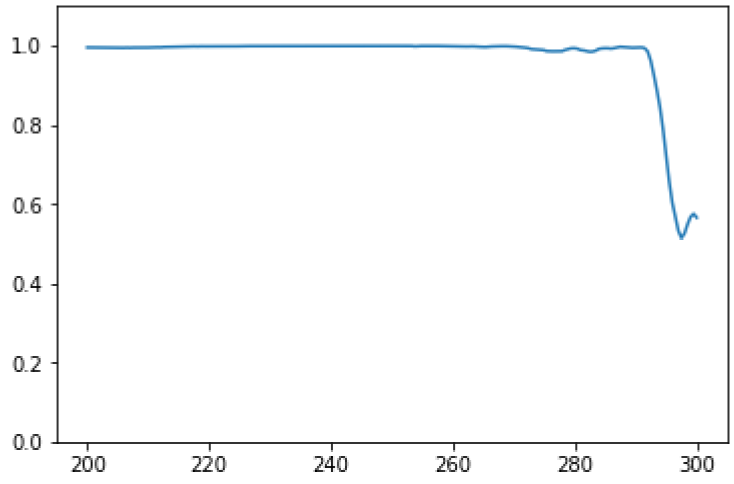
```
Predicted spectrum:
```



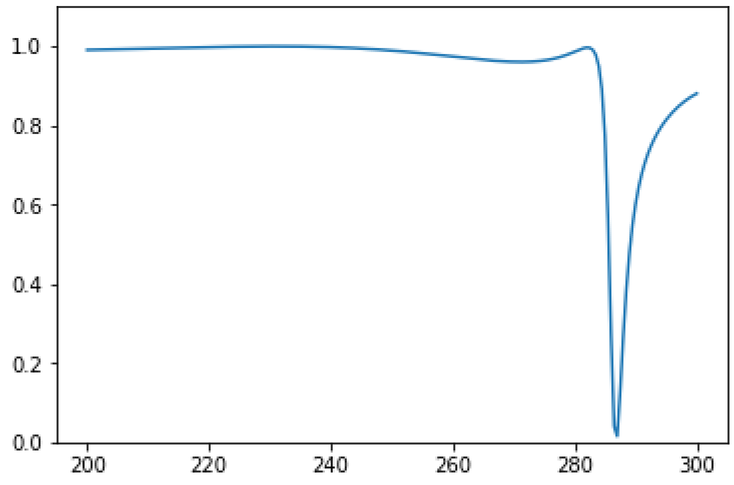
```
Test 28
True spectrum:
```



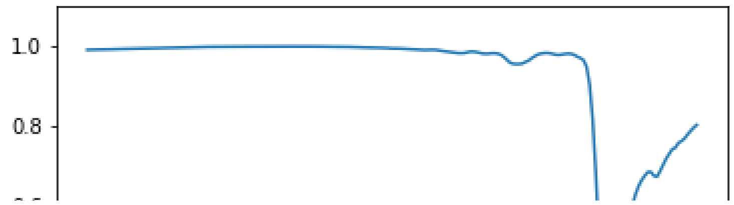
```
Predicted spectrum:
```



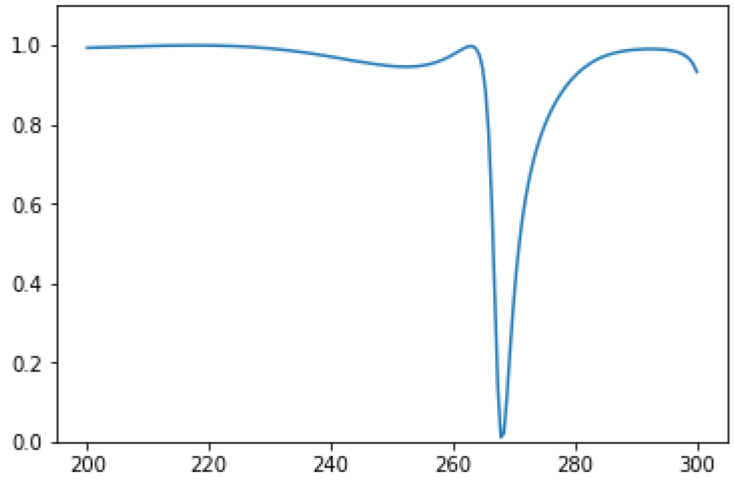
```
Test 29
True spectrum:
```



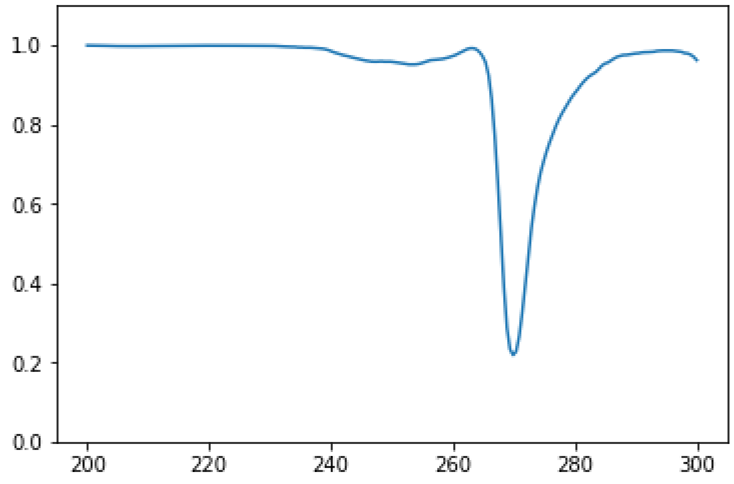
```
Predicted spectrum:
```



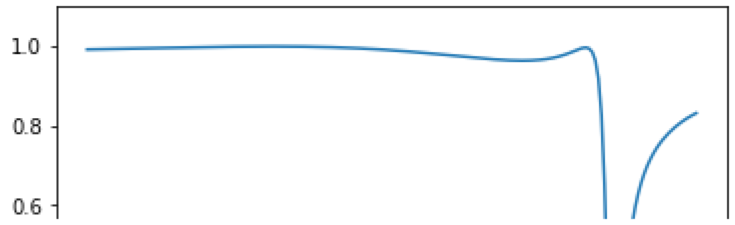
```
Test 30
True spectrum:
```



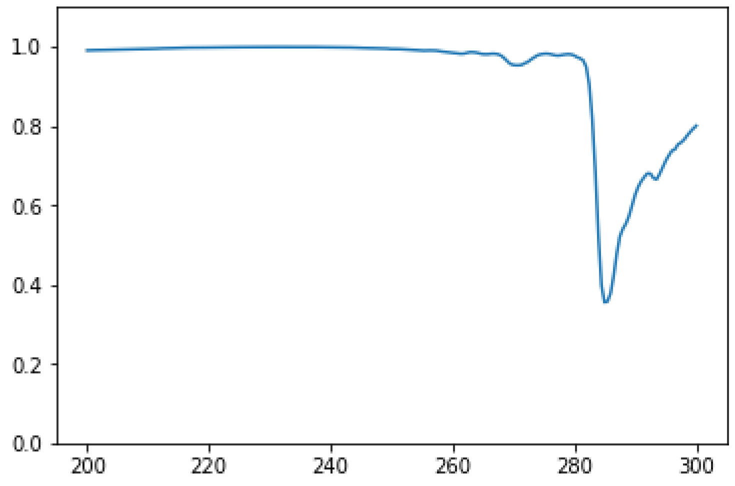
```
Predicted spectrum:
```



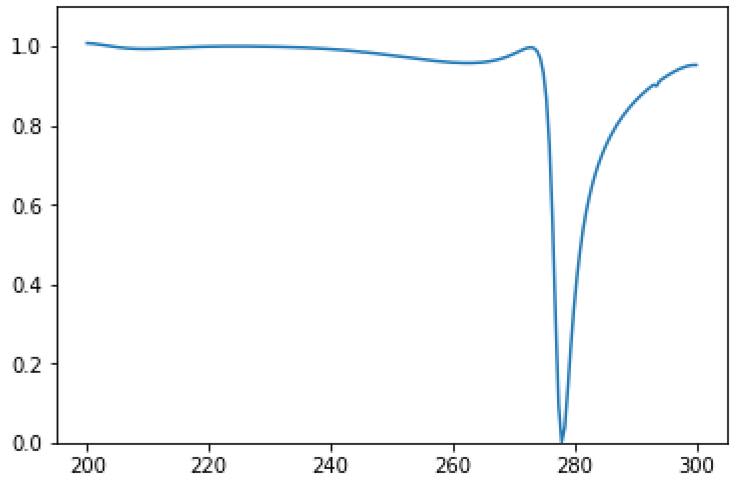
```
Test 31
True spectrum:
```



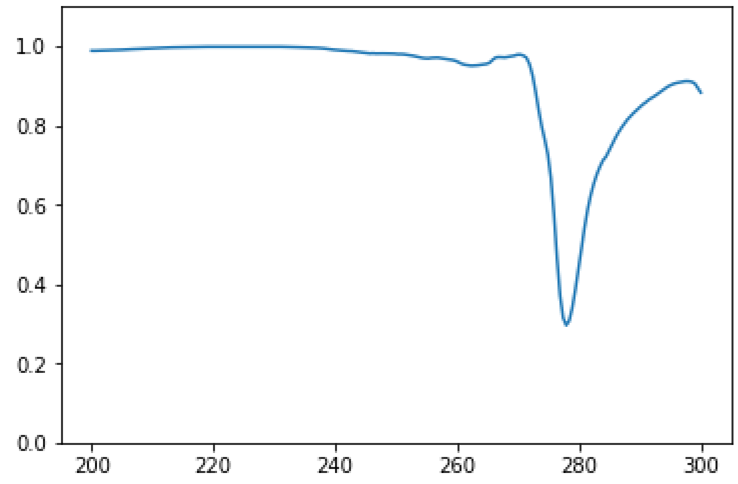
```
Predicted spectrum:
```



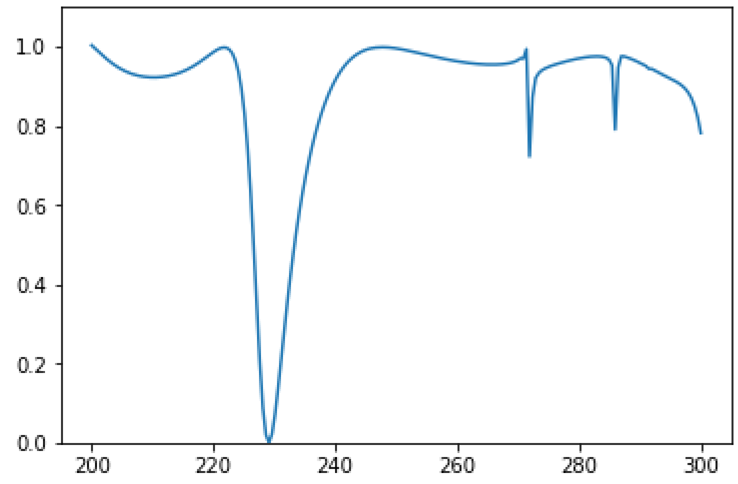
```
Test 32
True spectrum:
```



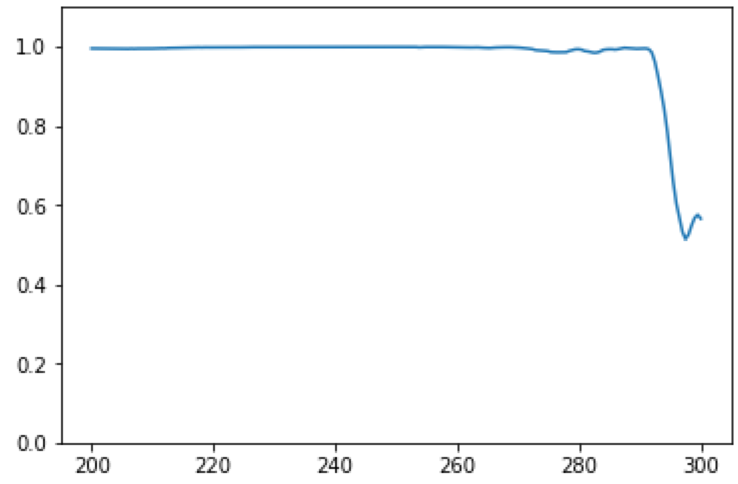
```
Predicted spectrum:
```



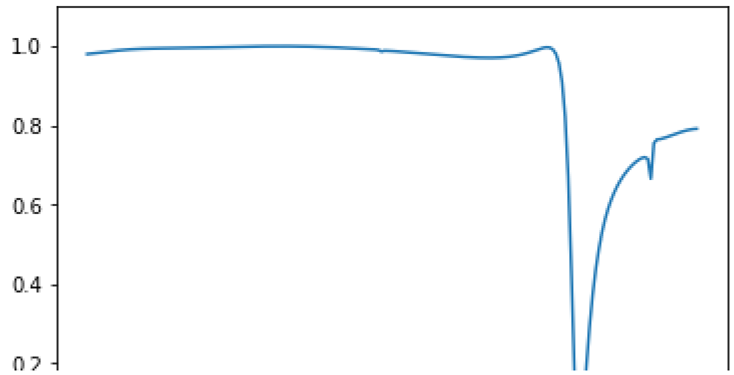
```
Test 33
True spectrum:
```



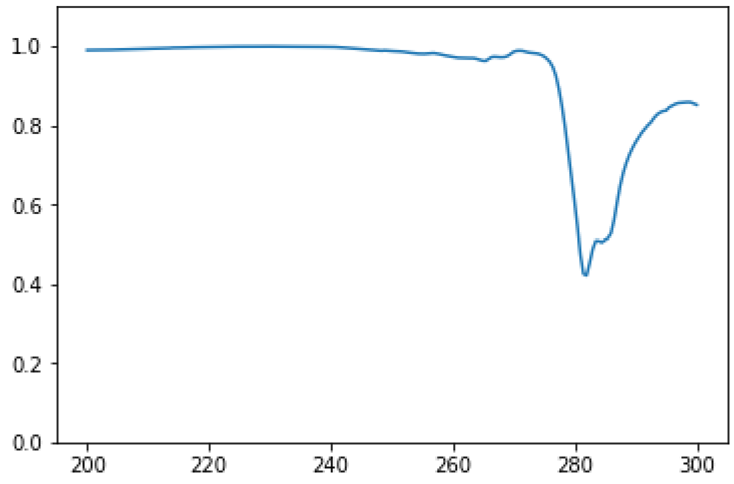
```
Predicted spectrum:
```



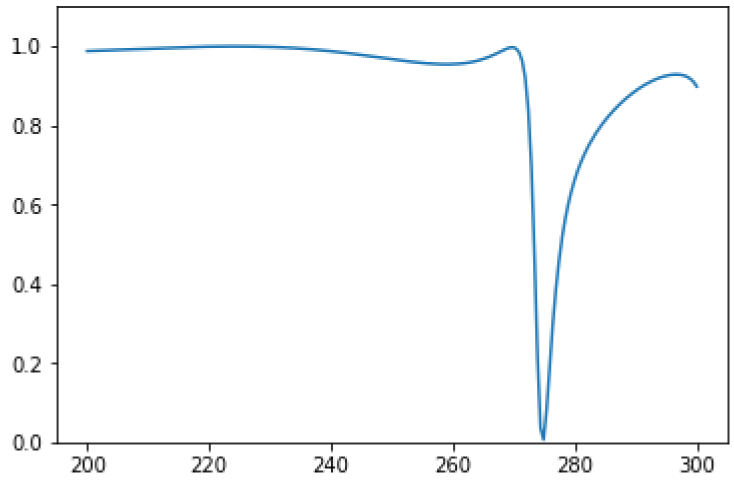
```
Test 34
True spectrum:
```



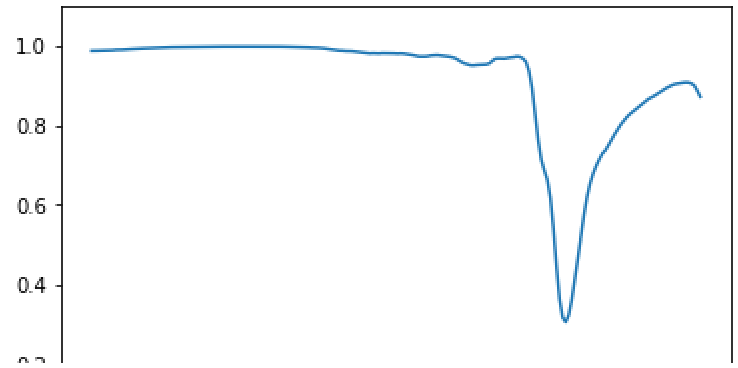
```
Predicted spectrum:
```



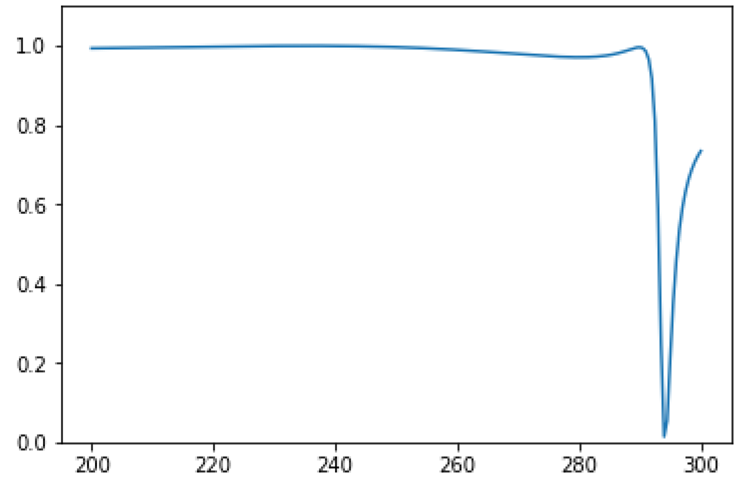
```
Test 35
True spectrum:
```



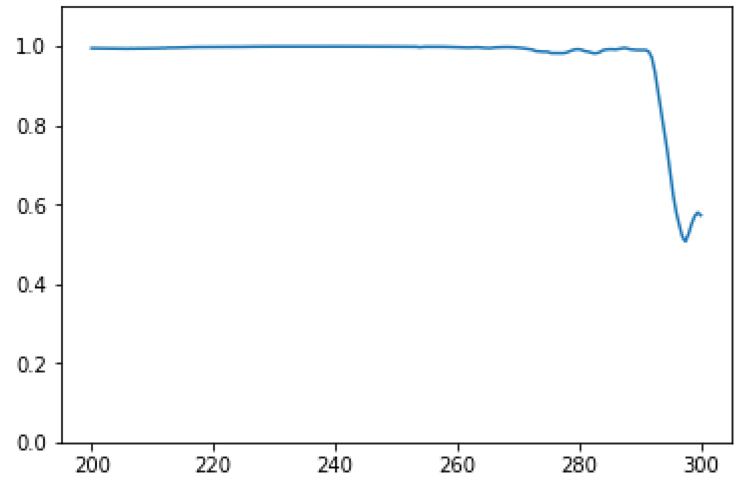
```
Predicted spectrum:
```



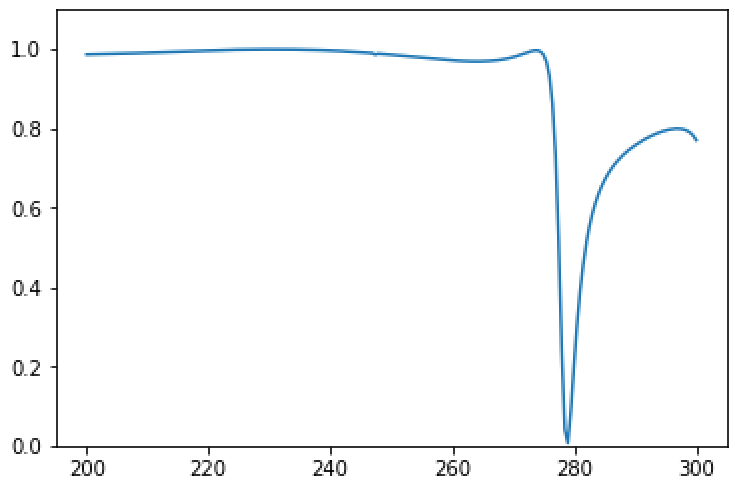
```
Test 36
True spectrum:
```



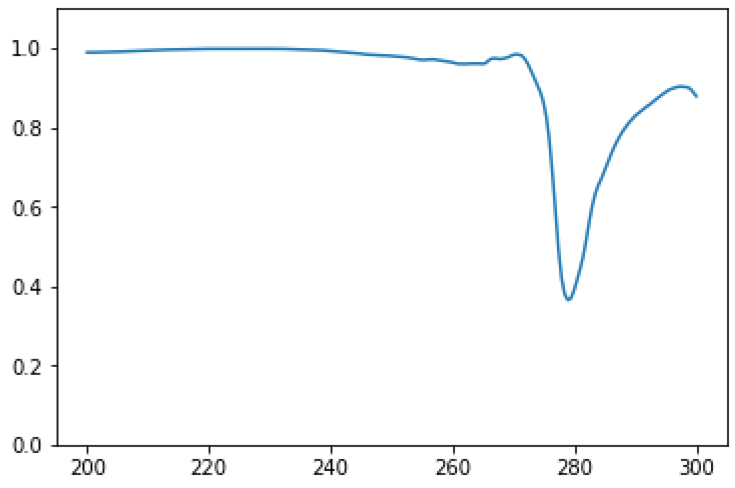
```
Predicted spectrum:
```



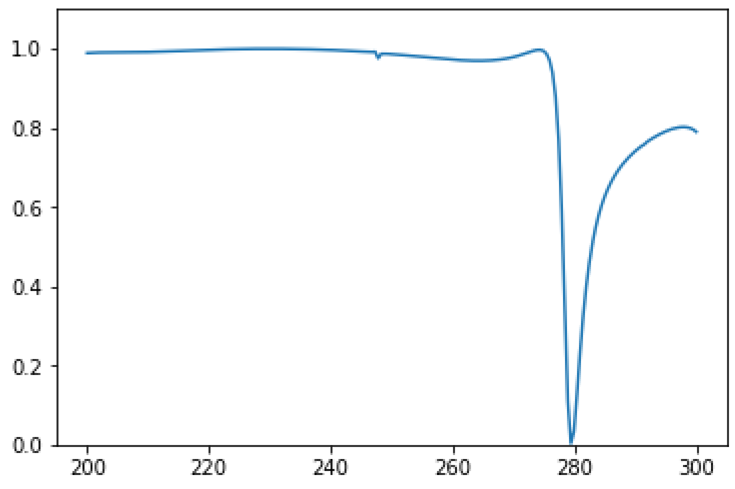
```
Test 37
True spectrum:
```



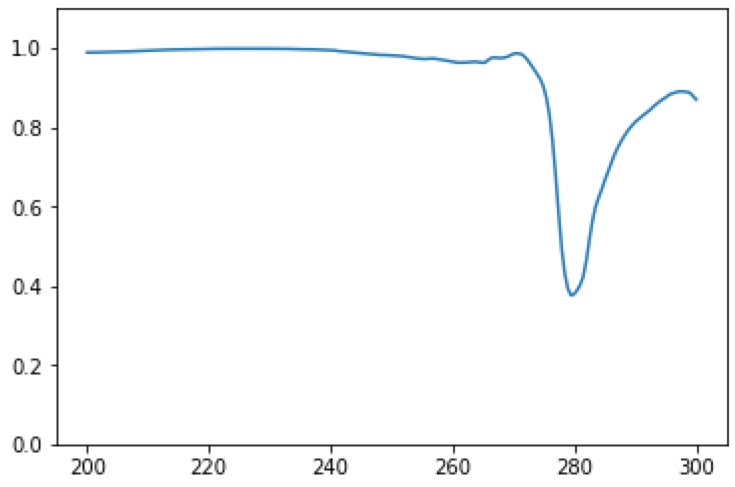
```
Predicted spectrum:
```



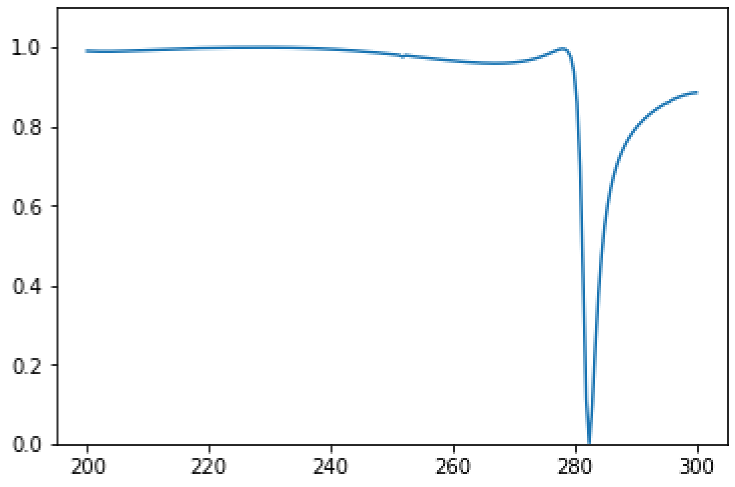
```
Test 38
True spectrum:
```



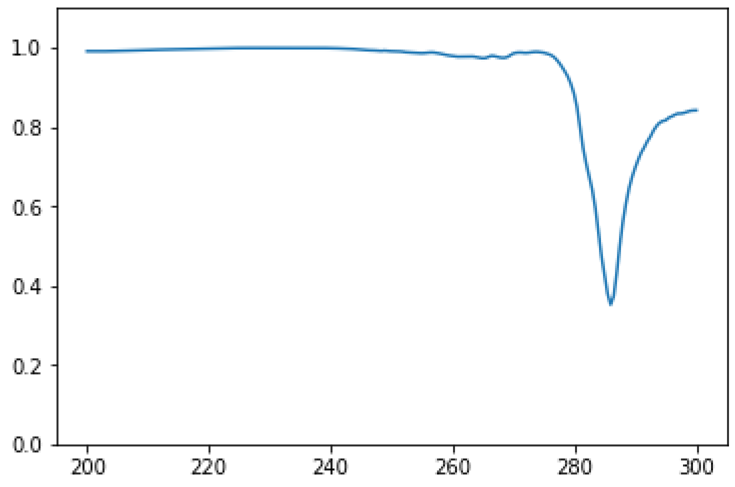
```
Predicted spectrum:
```



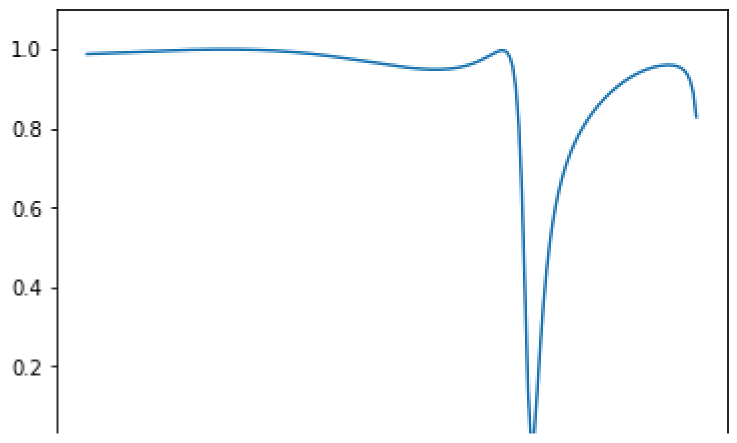
```
Test 39
True spectrum:
```



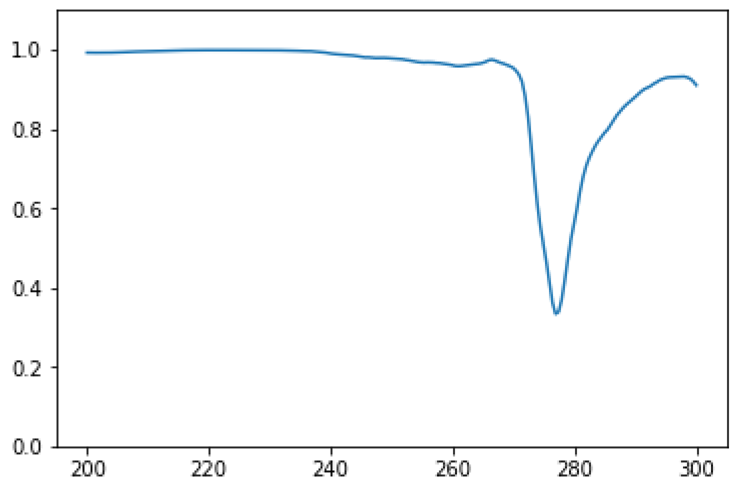
```
Predicted spectrum:
```



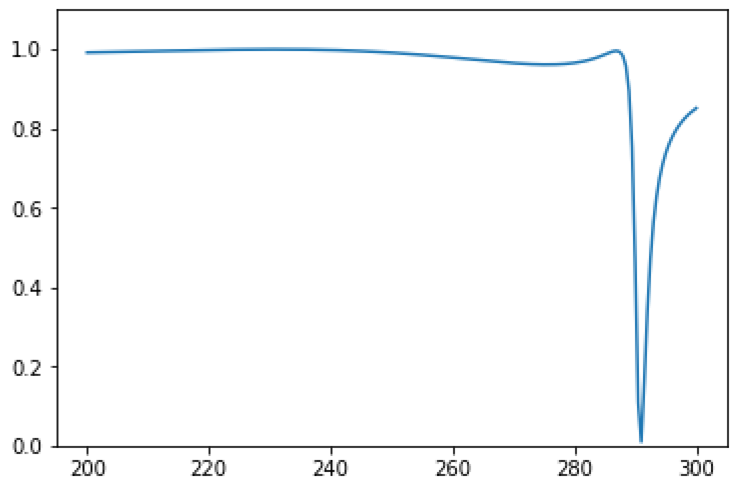
```
Test 40
True spectrum:
```



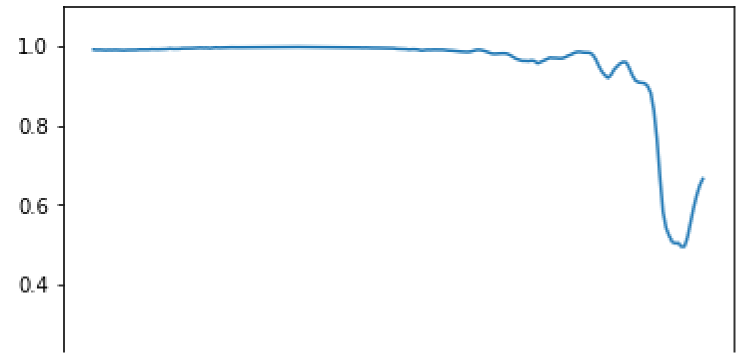
```
Predicted spectrum:
```



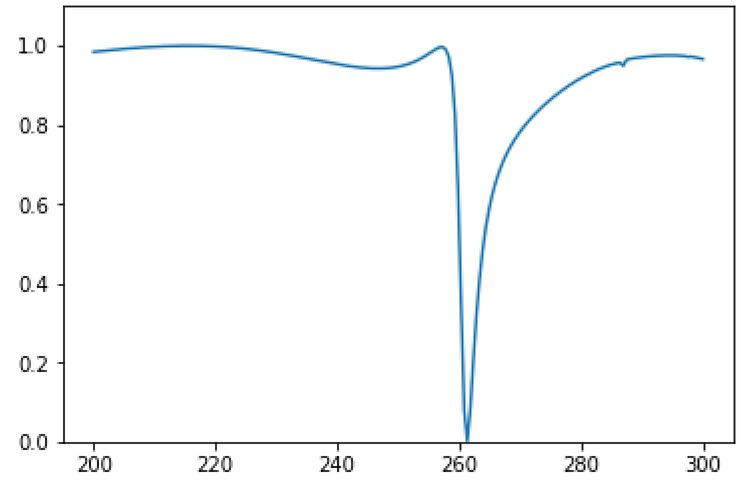
```
Test 41
True spectrum:
```



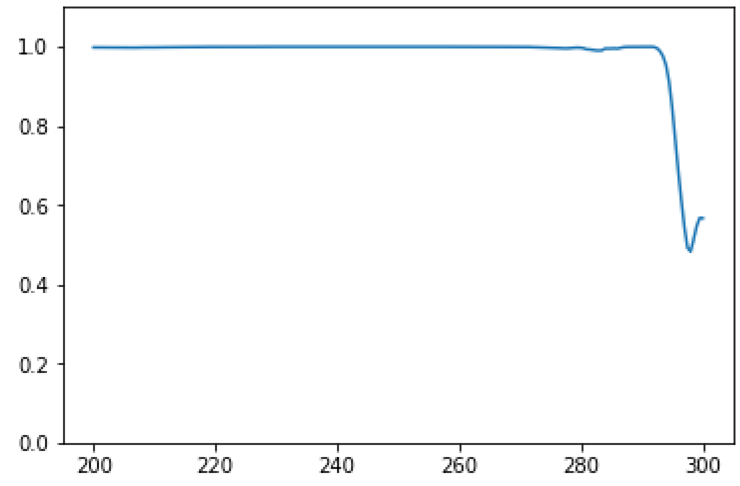
```
Predicted spectrum:
```



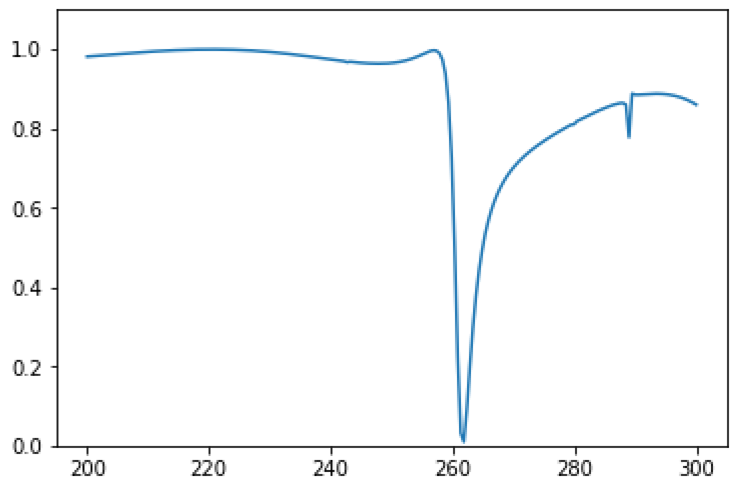
```
Test 42
True spectrum:
```



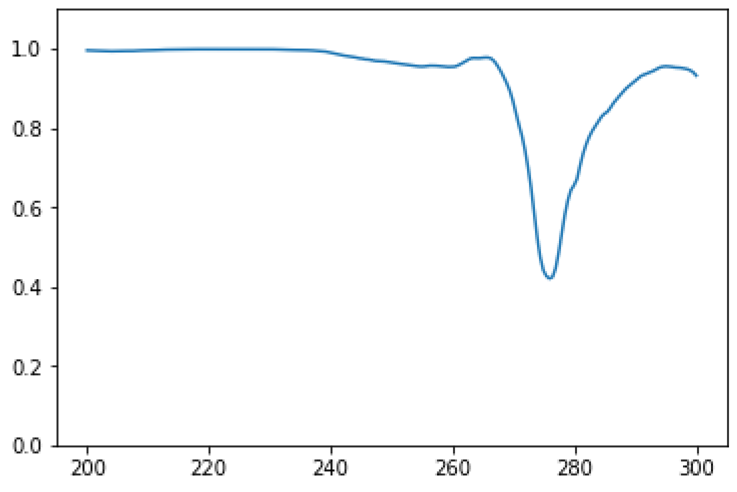
```
Predicted spectrum:
```



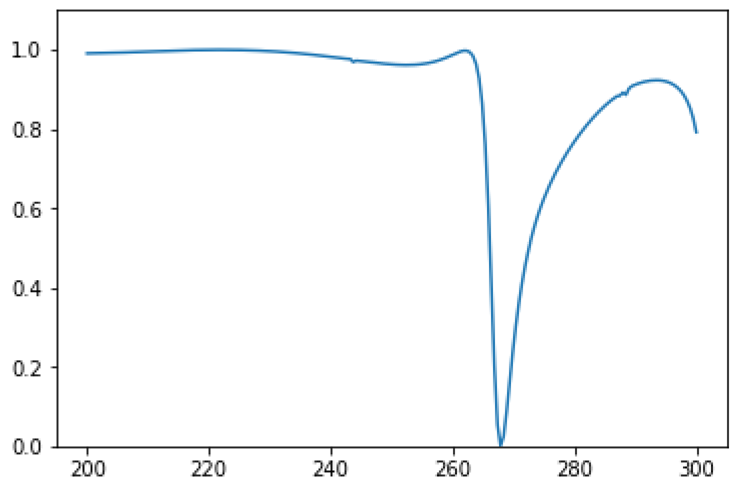
```
Test 43
True spectrum:
```



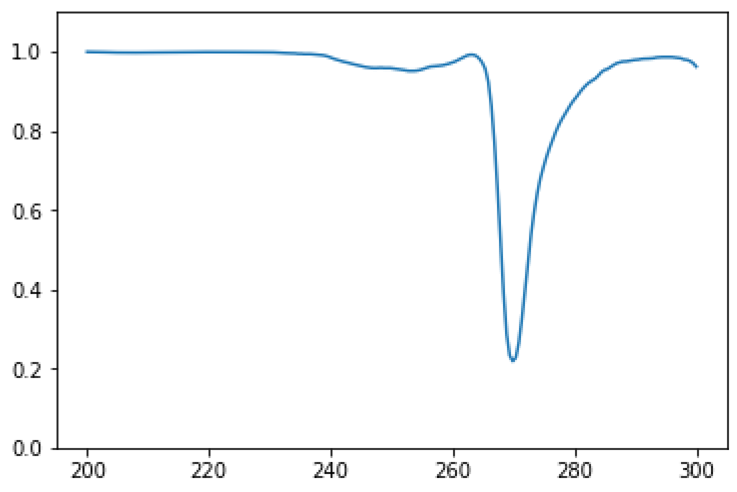
```
Predicted spectrum:
```



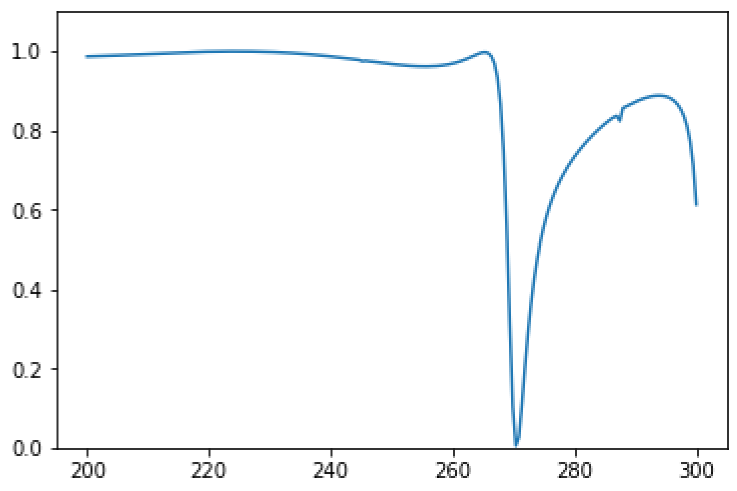
```
Test 44
True spectrum:
```



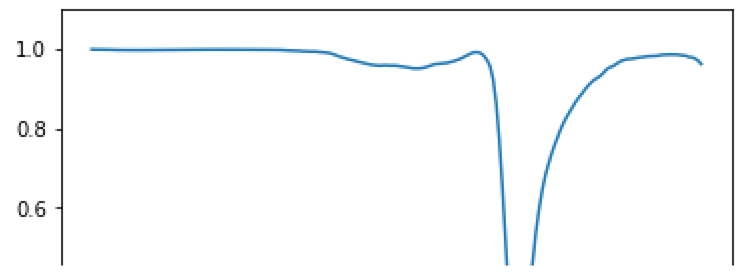
```
Predicted spectrum:
```



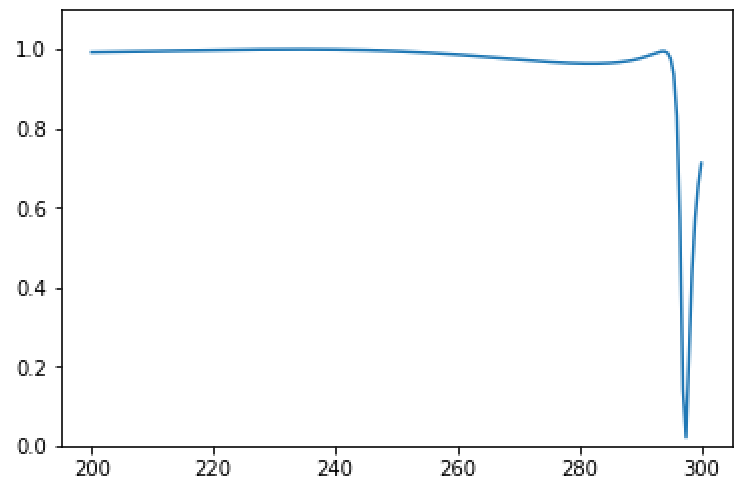
```
Test 45
True spectrum:
```



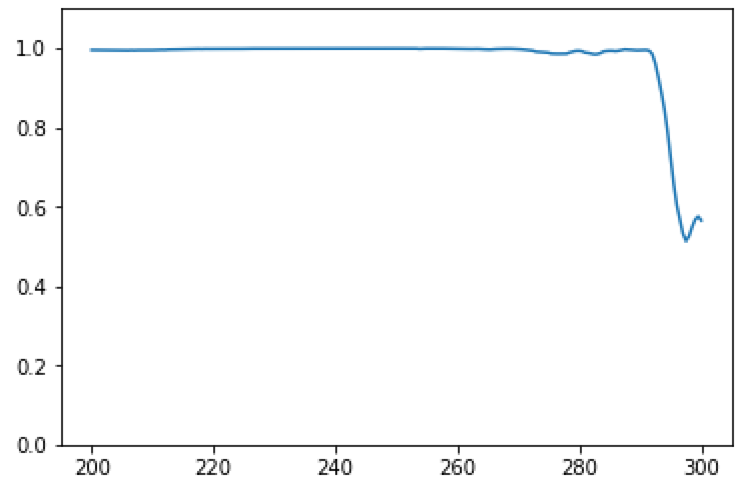
```
Predicted spectrum:
```



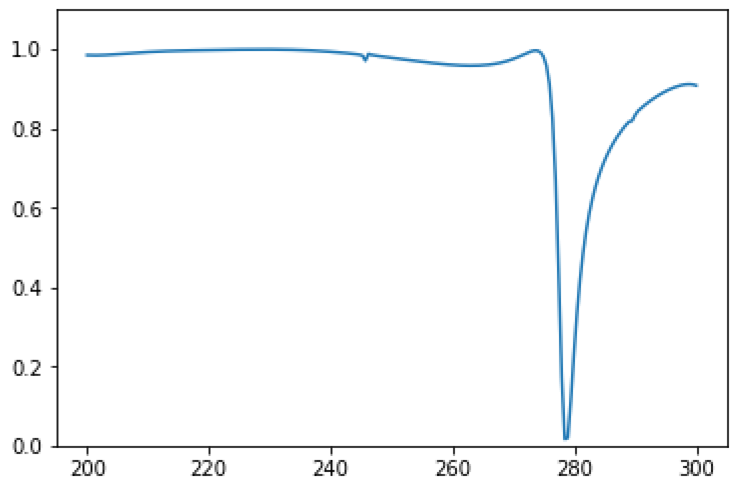
```
Test 46
True spectrum:
```



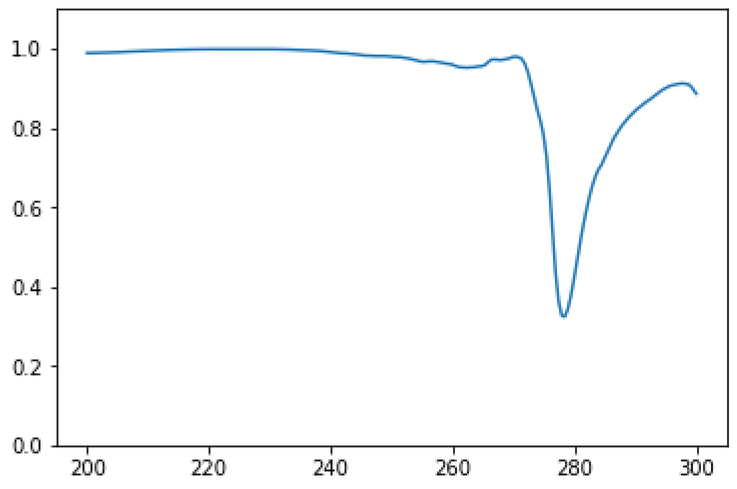
```
Predicted spectrum:
```



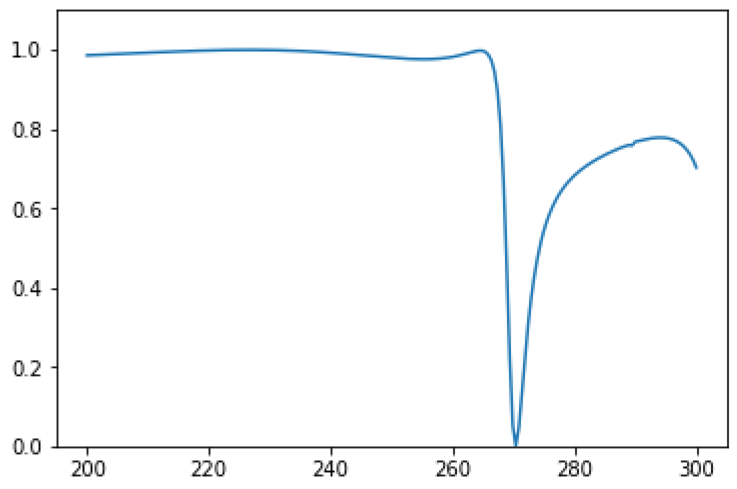
```
Test 47
True spectrum:
```



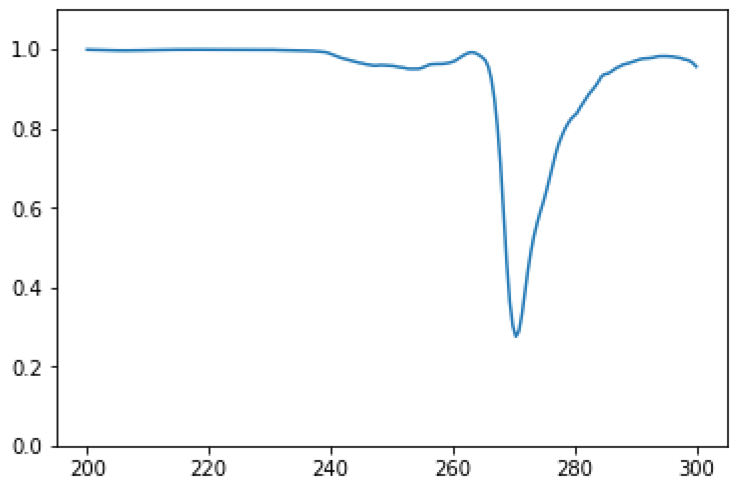
```
Predicted spectrum:
```



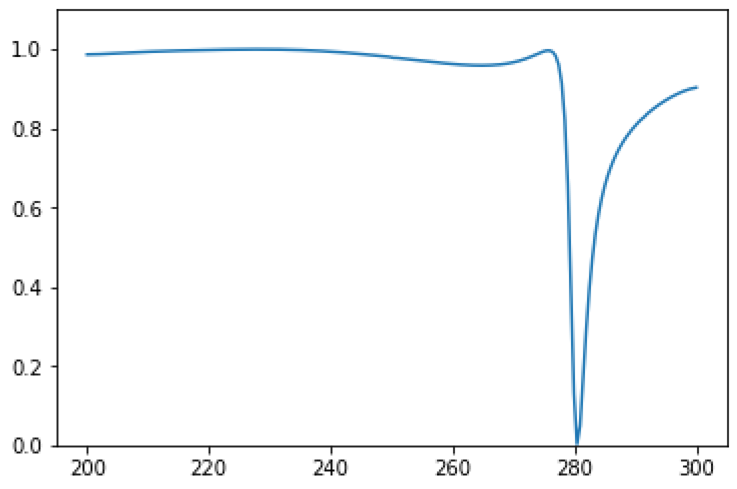
```
Test 48
True spectrum:
```



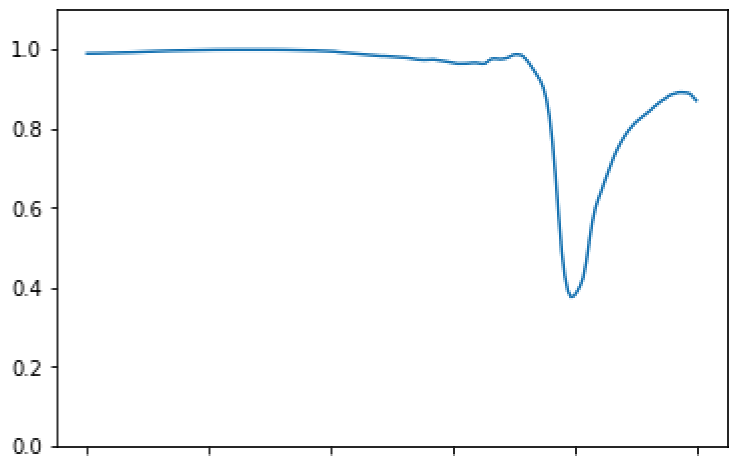
```
Predicted spectrum:
```



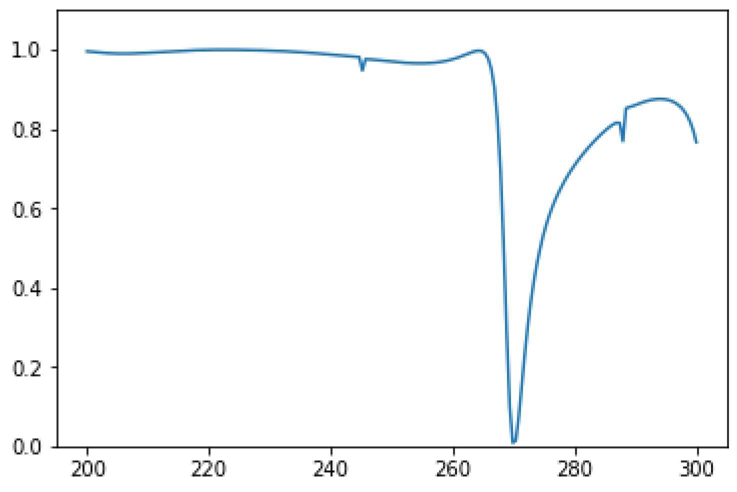
```
Test 49
True spectrum:
```



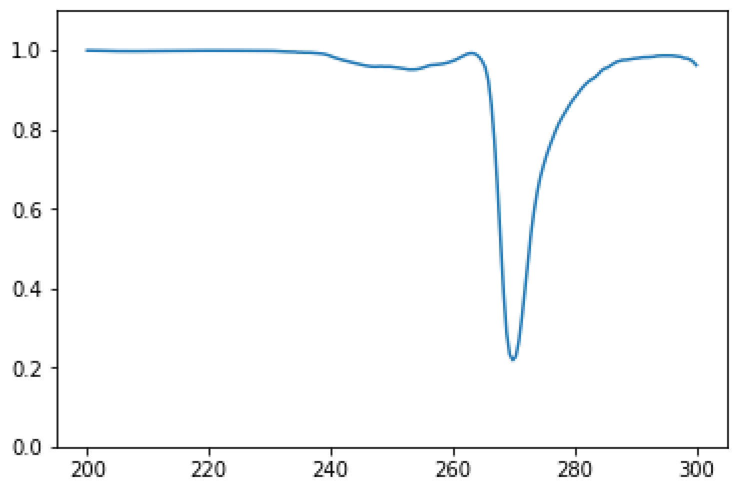
```
Predicted spectrum:
```



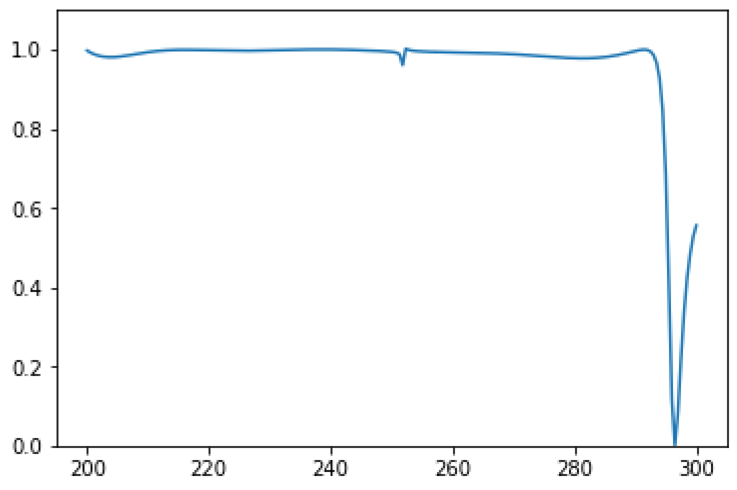
```
Test 50
True spectrum:
```



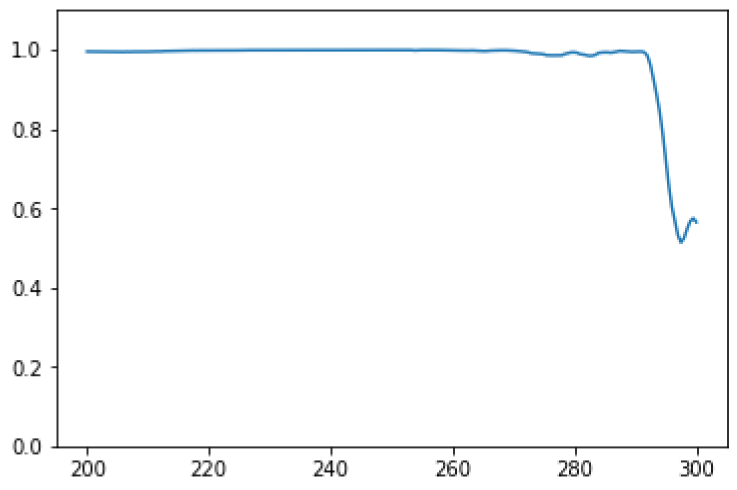
```
Predicted spectrum:
```



```
Test 51
True spectrum:
```

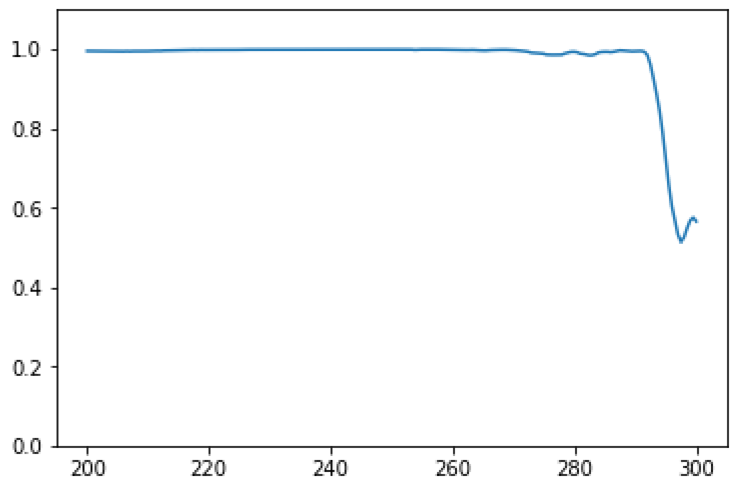


```
Predicted spectrum:
```

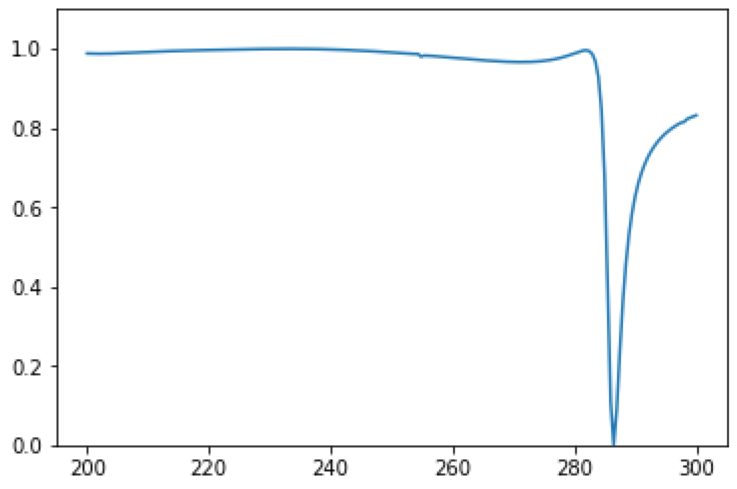


```
Test 52
True spectrum:
```

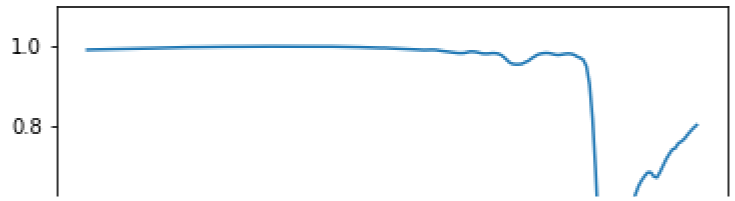
```
Predicted spectrum:
```



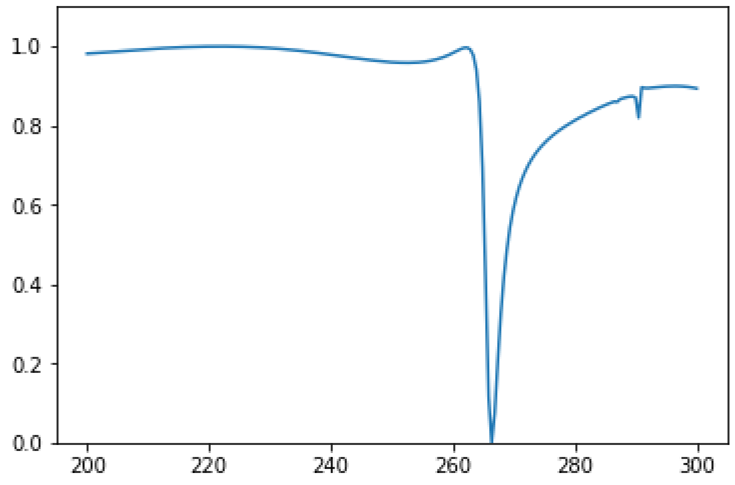
```
Test 53
True spectrum:
```



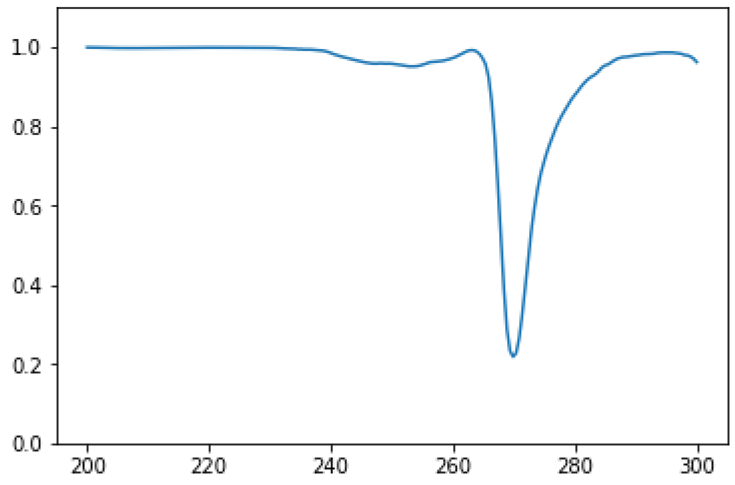
```
Predicted spectrum:
```



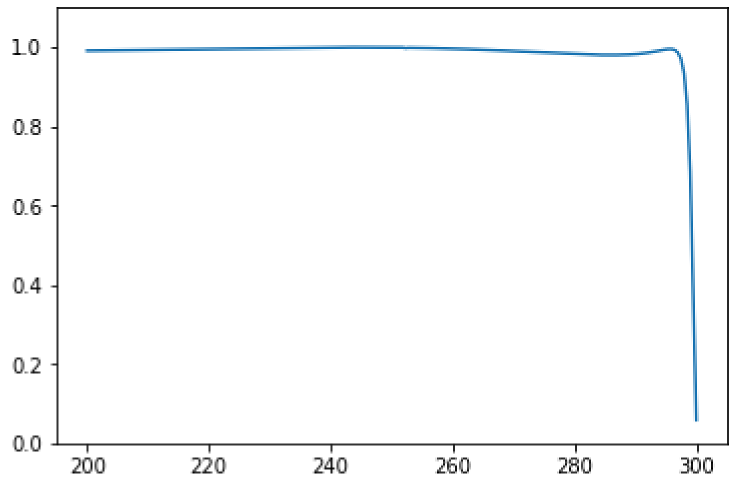
```
Test 54
True spectrum:
```



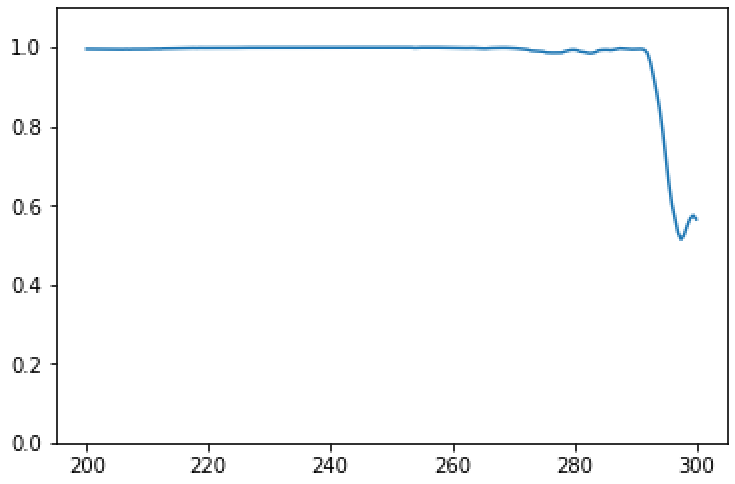
```
Predicted spectrum:
```



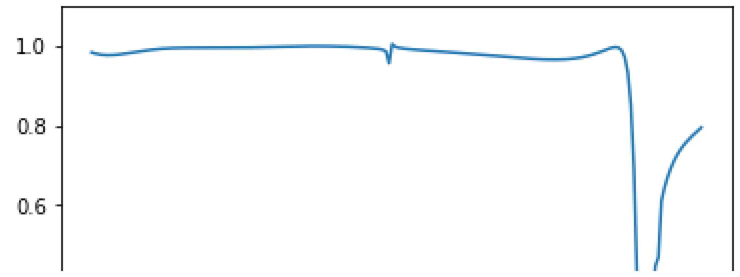
```
Test 55
True spectrum:
```



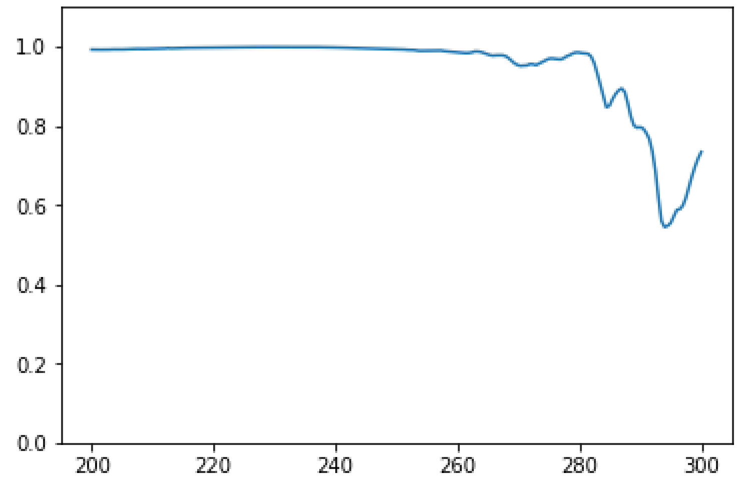
```
Predicted spectrum:
```



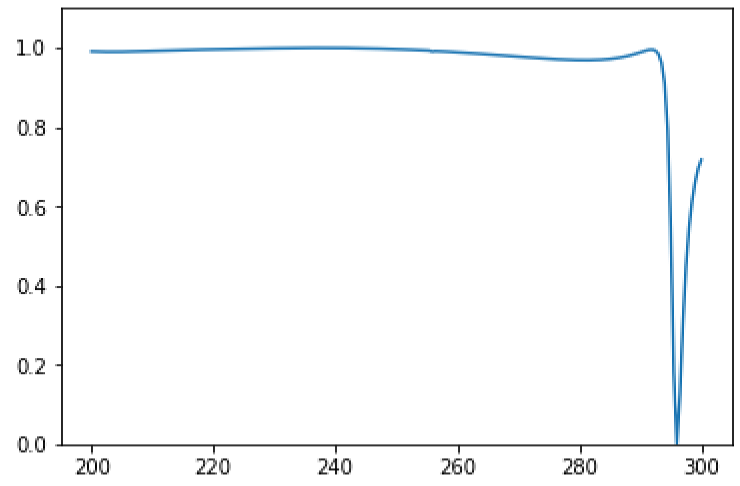
```
Test 56
True spectrum:
```



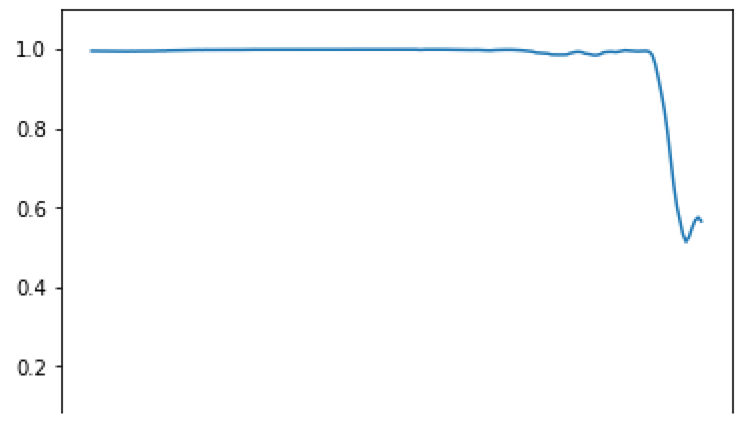
```
Predicted spectrum:
```



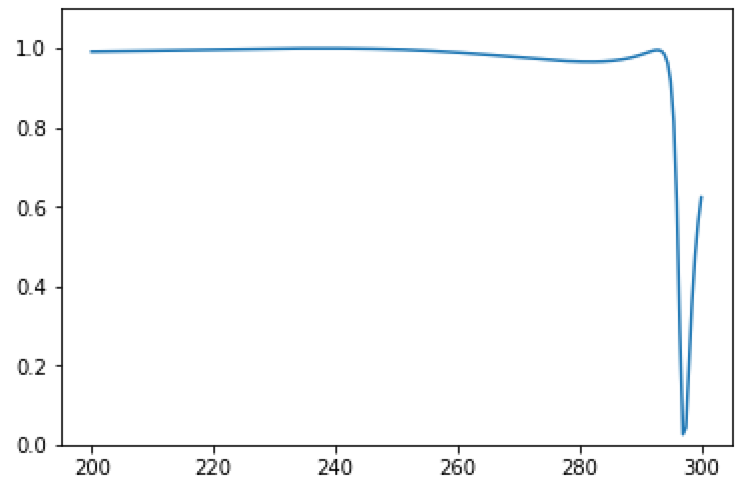
```
Test 57
True spectrum:
```



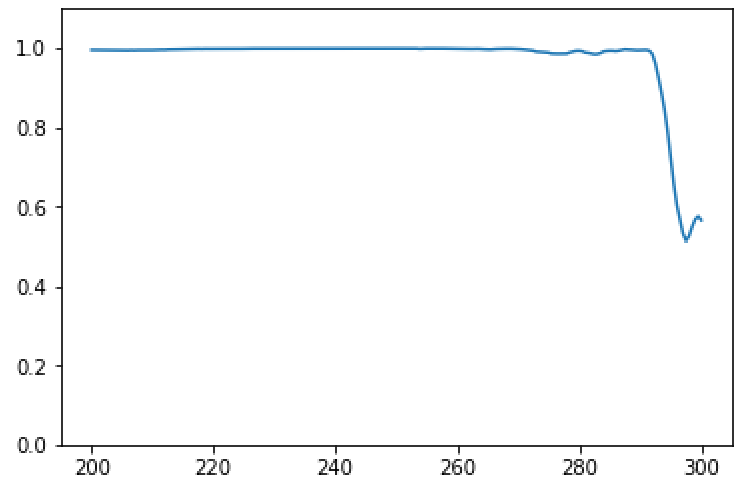
```
Predicted spectrum:
```



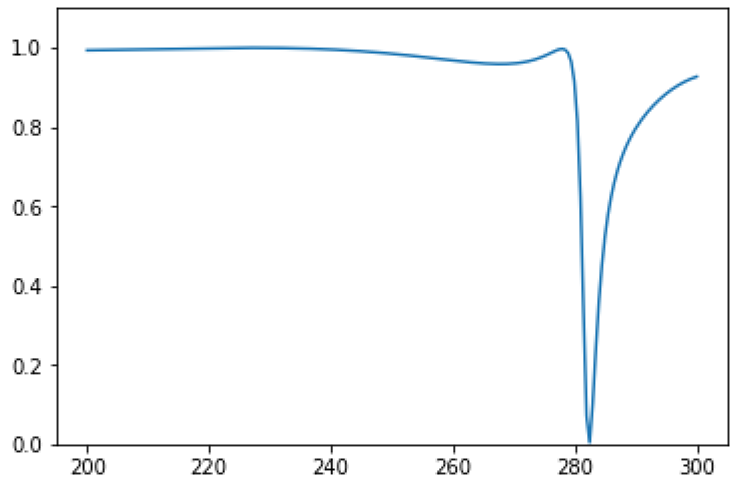
```
Test 58
True spectrum:
```



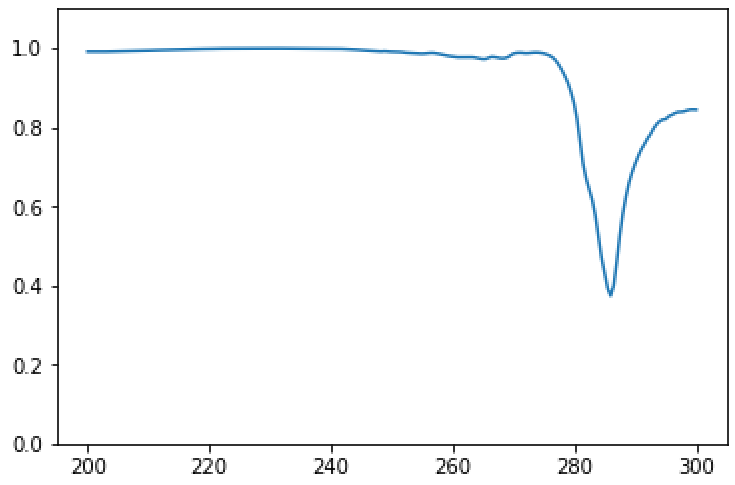
```
Predicted spectrum:
```



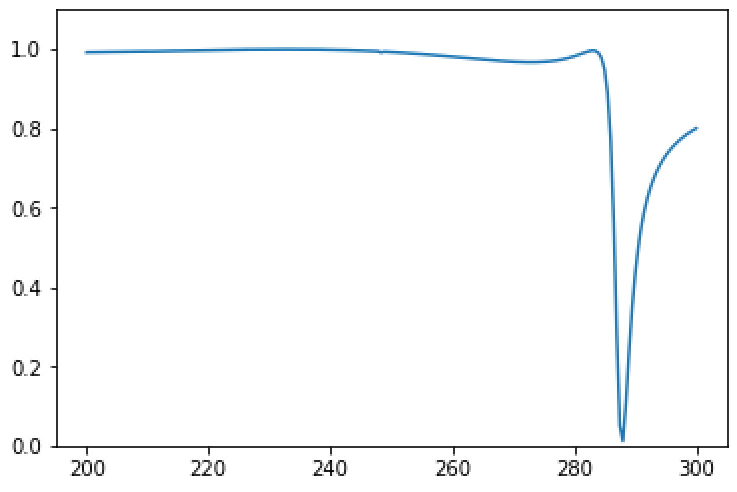
```
Test 59
True spectrum:
```



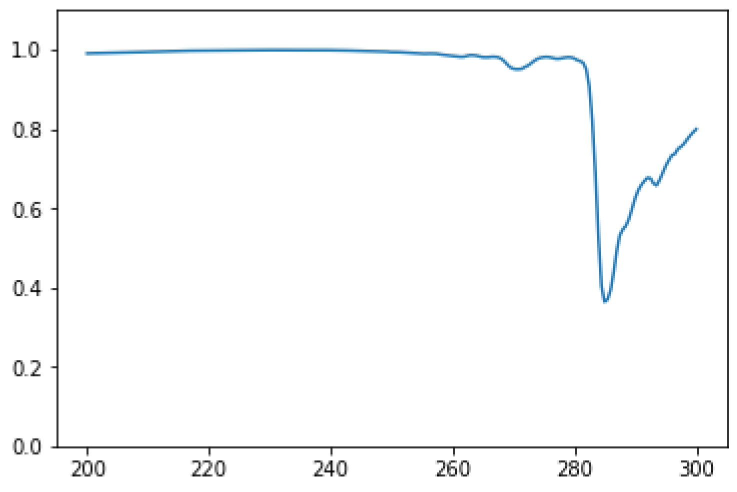
```
Predicted spectrum:
```



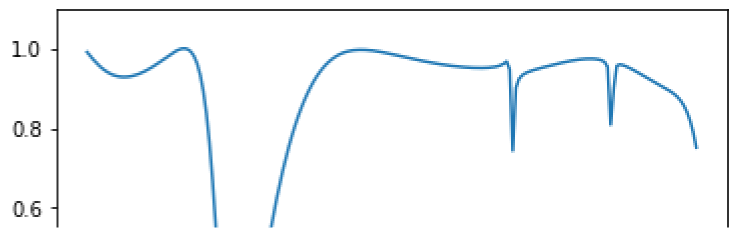
```
Test 60
True spectrum:
```



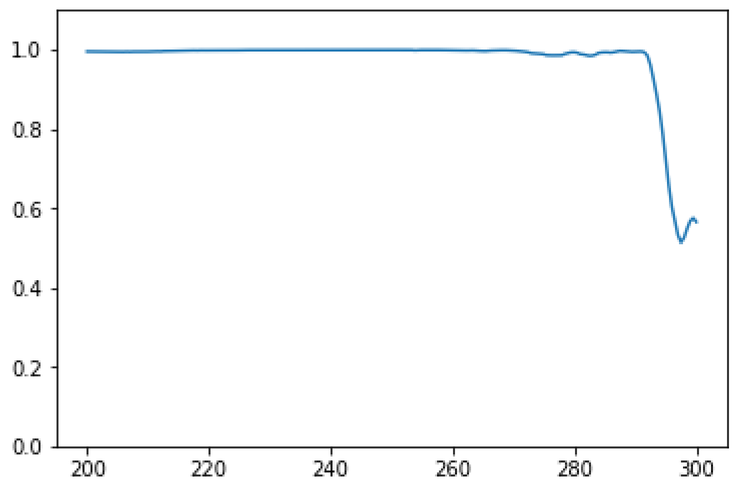
```
Predicted spectrum:
```



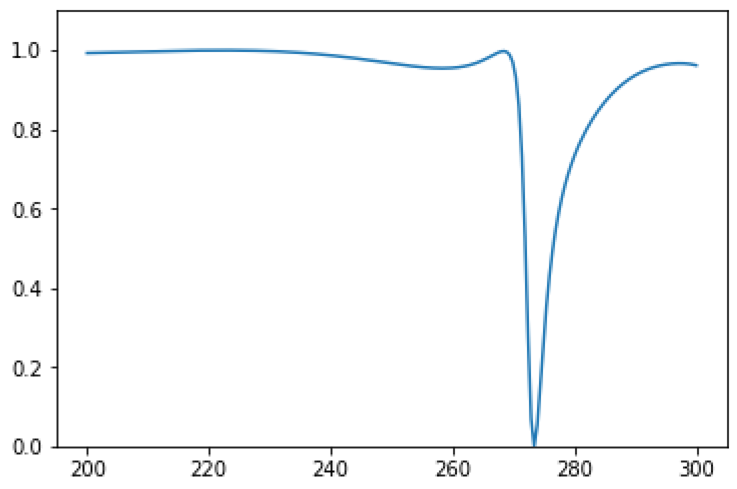
```
Test 61
True spectrum:
```



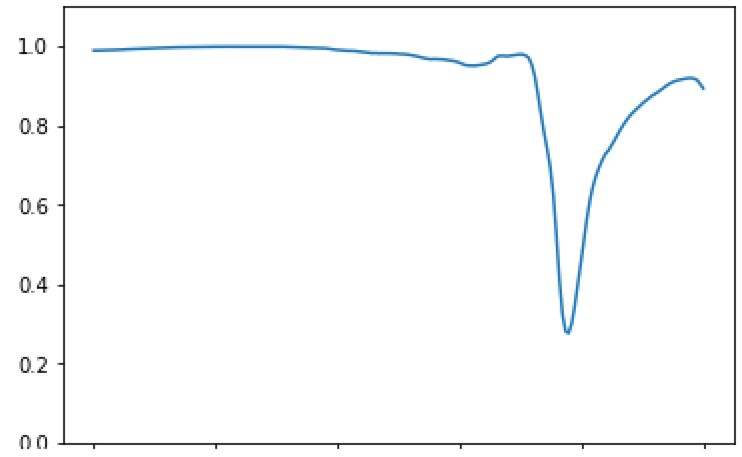
```
Predicted spectrum:
```



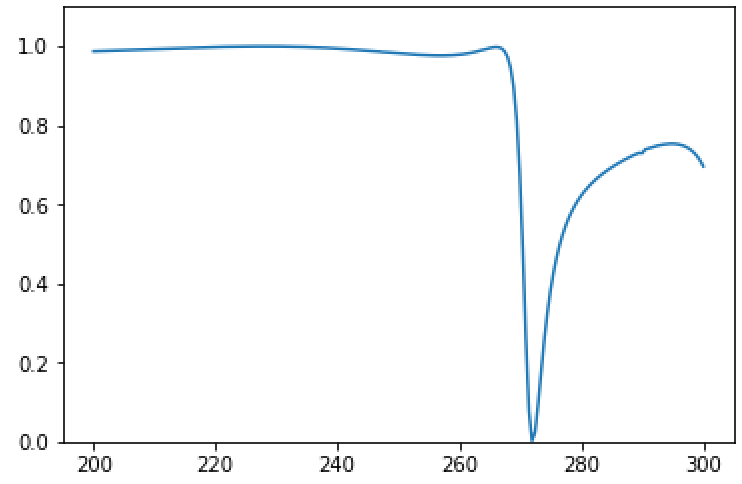
```
Test 62
True spectrum:
```



```
Predicted spectrum:
```

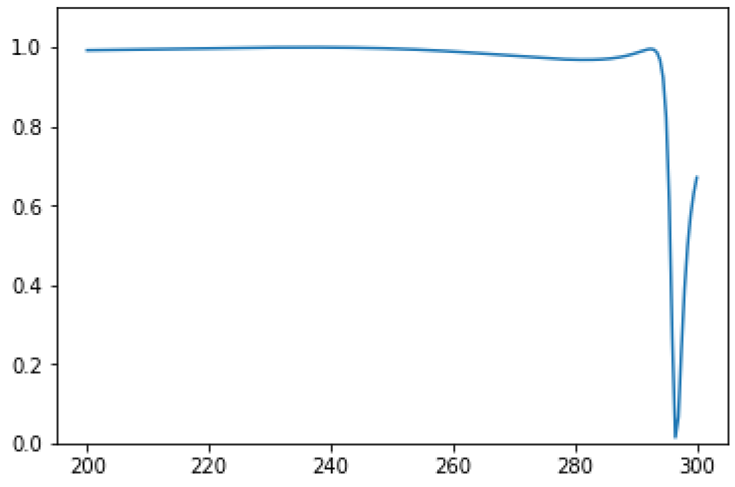


```
Test 63
True spectrum:
```

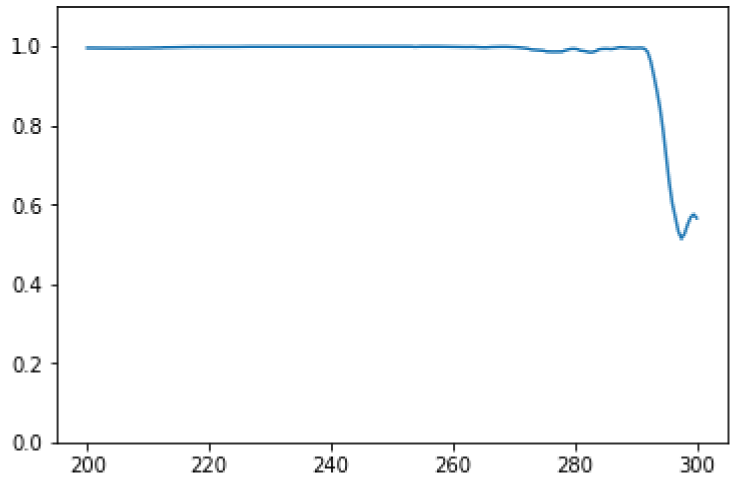


```
Predicted spectrum:
```

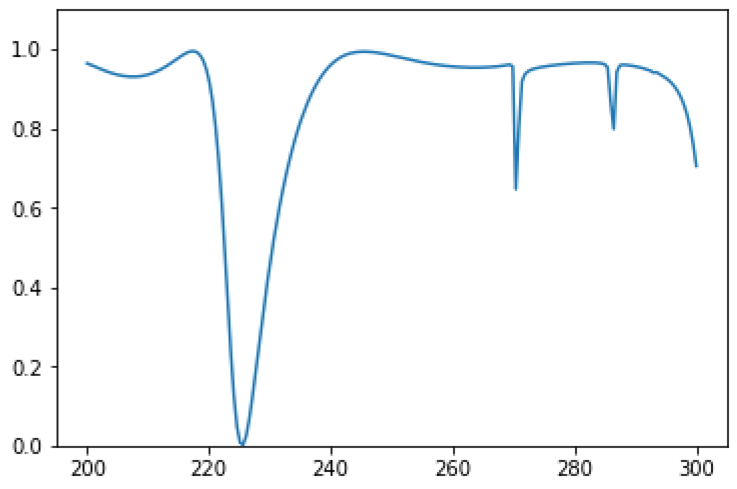
```
Test 64
True spectrum:
```



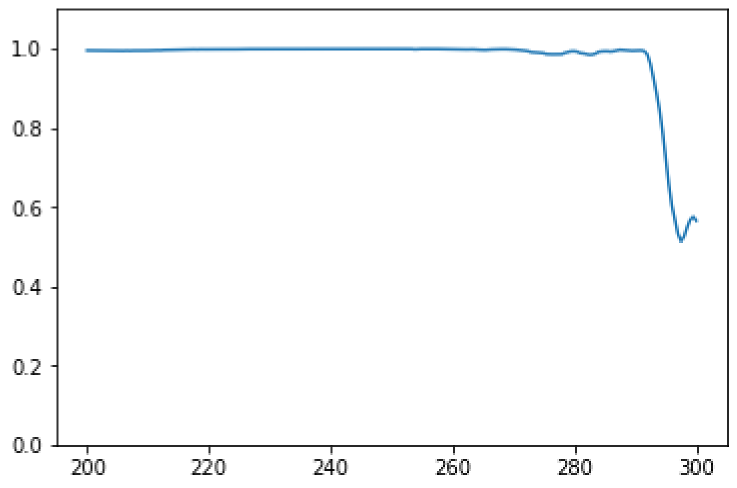
```
Predicted spectrum:
```



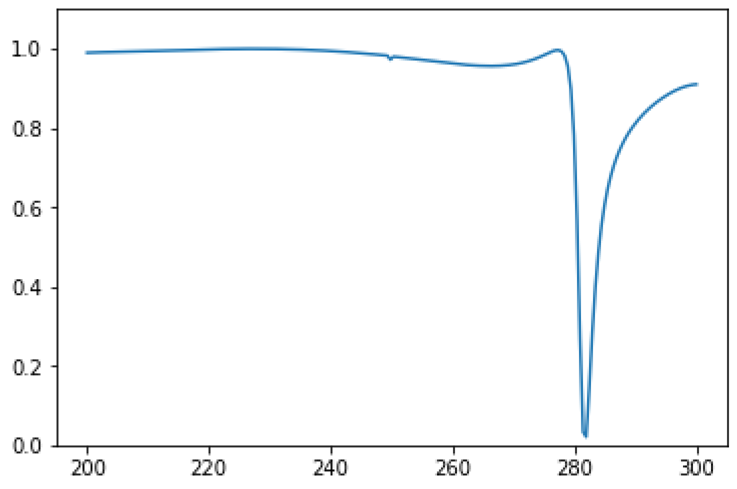
```
Test 65
True spectrum:
```



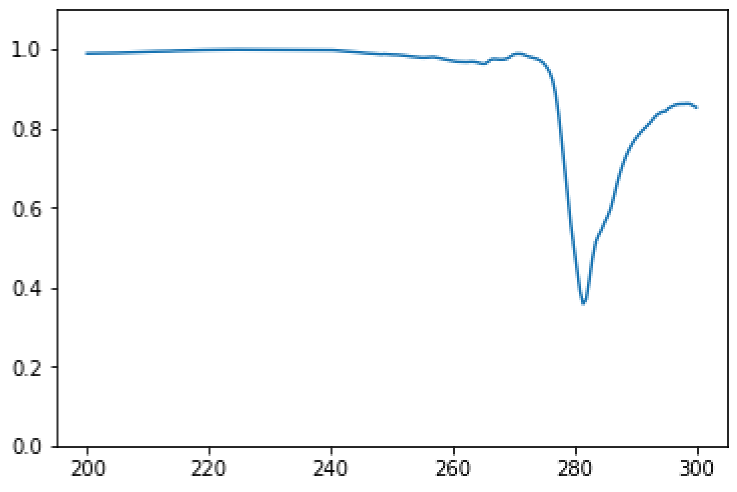
```
Predicted spectrum:
```



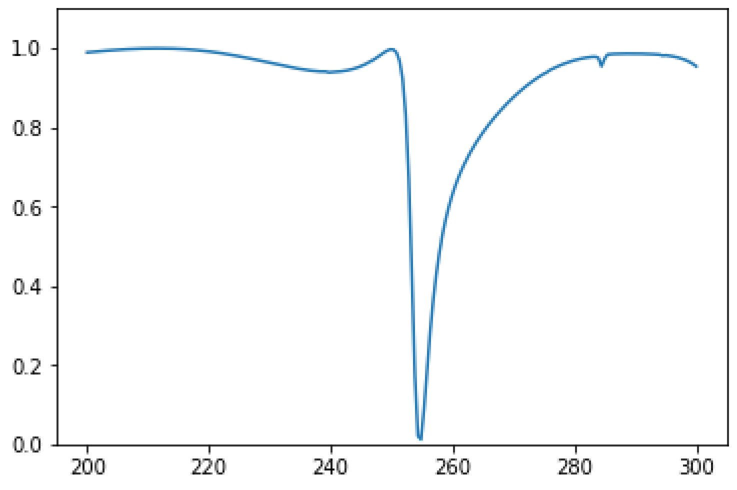
```
Test 66
True spectrum:
```



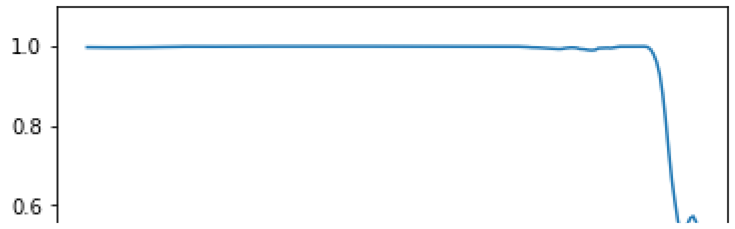
```
Predicted spectrum:
```



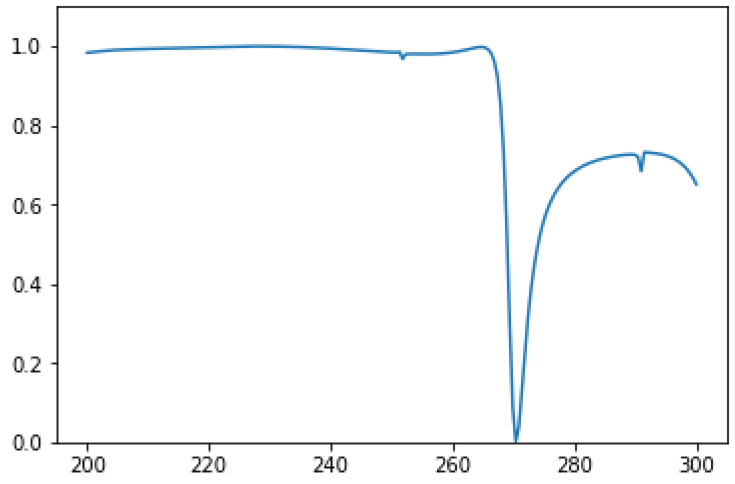
```
Test 67
True spectrum:
```



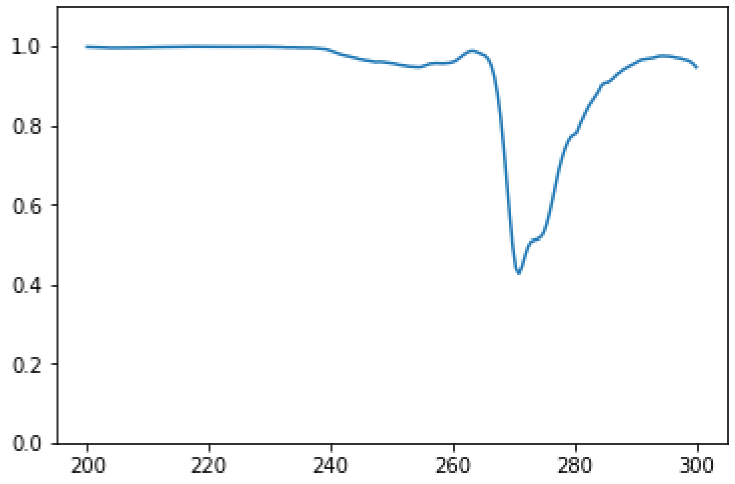
```
Predicted spectrum:
```



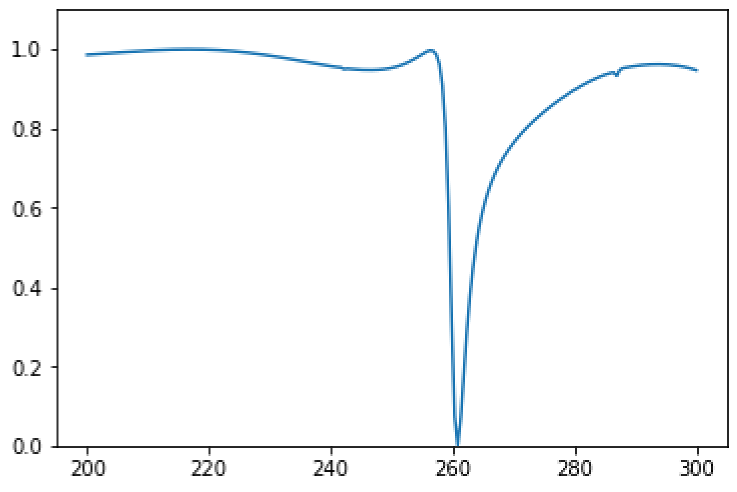
```
Test 68
True spectrum:
```



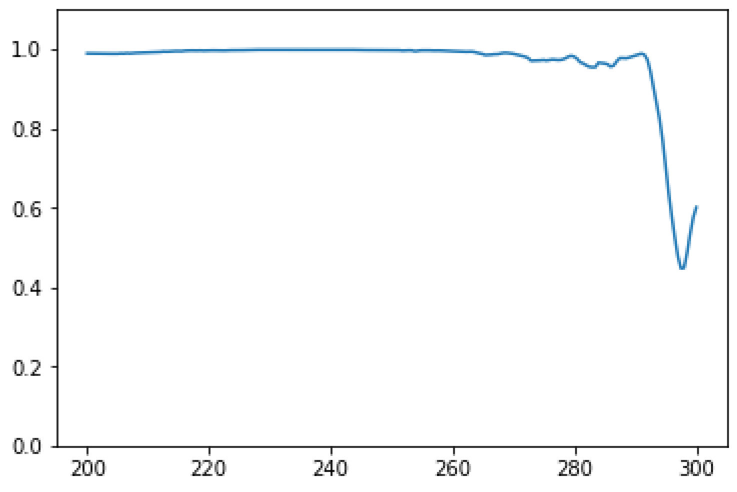
```
Predicted spectrum:
```



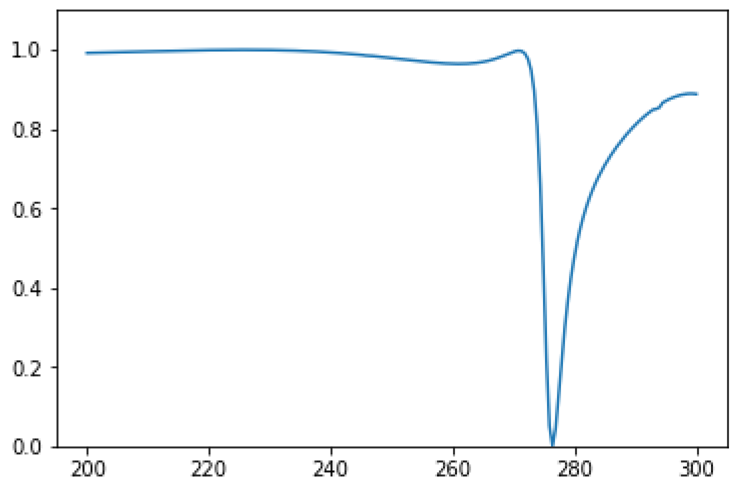
```
Test 69
True spectrum:
```



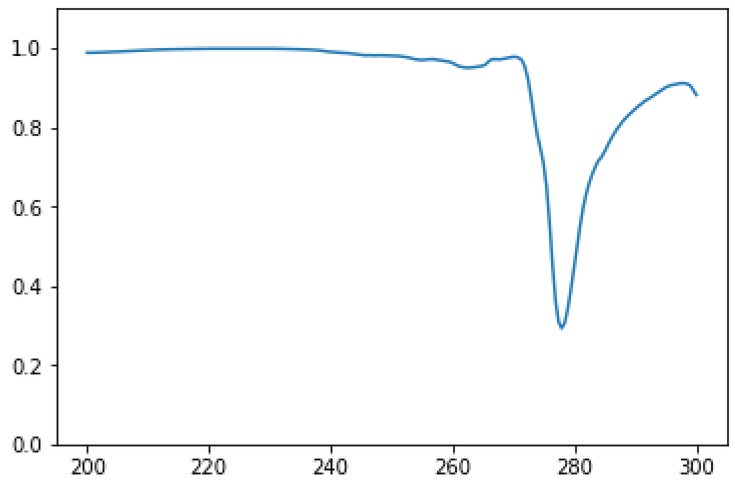
```
Predicted spectrum:
```



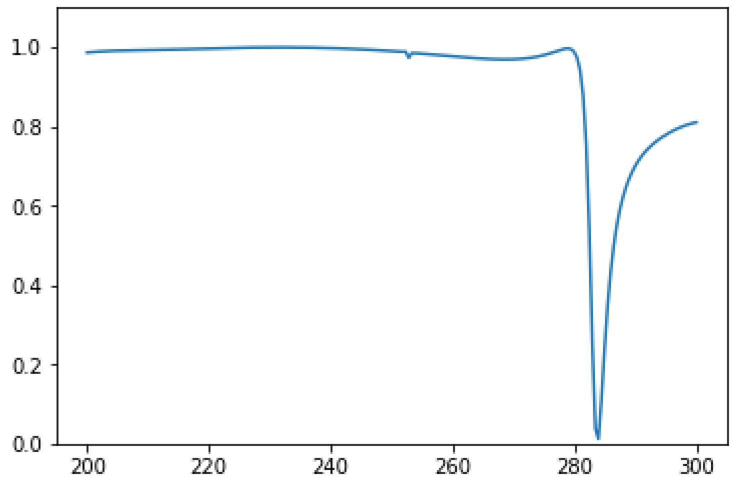
```
Test 70
True spectrum:
```



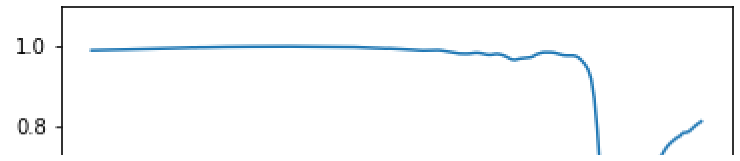
```
Predicted spectrum:
```



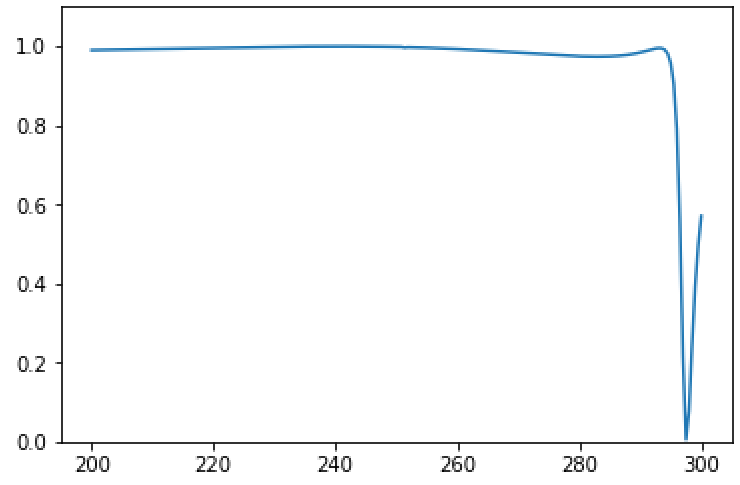
```
Test 71
True spectrum:
```



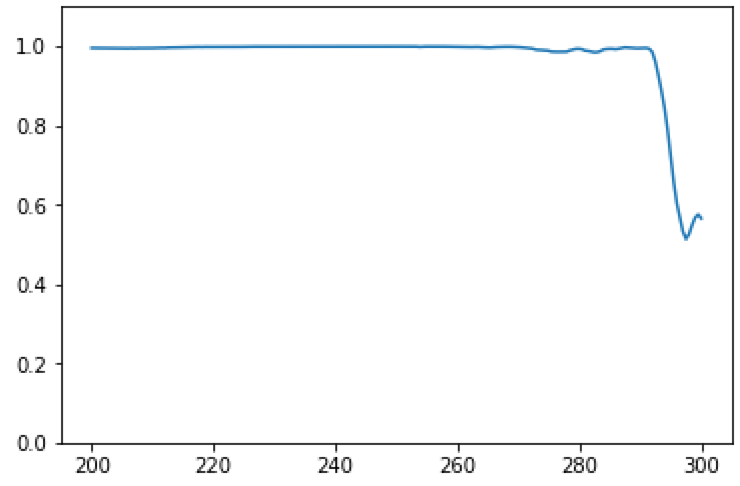
```
Predicted spectrum:
```



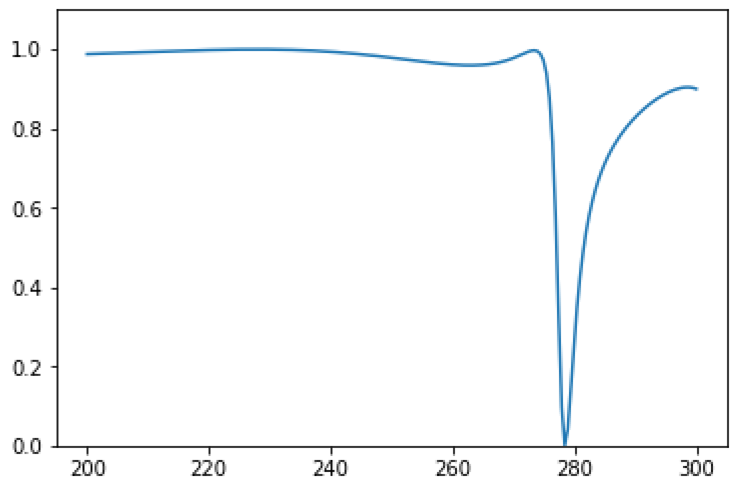
```
Test 72
True spectrum:
```



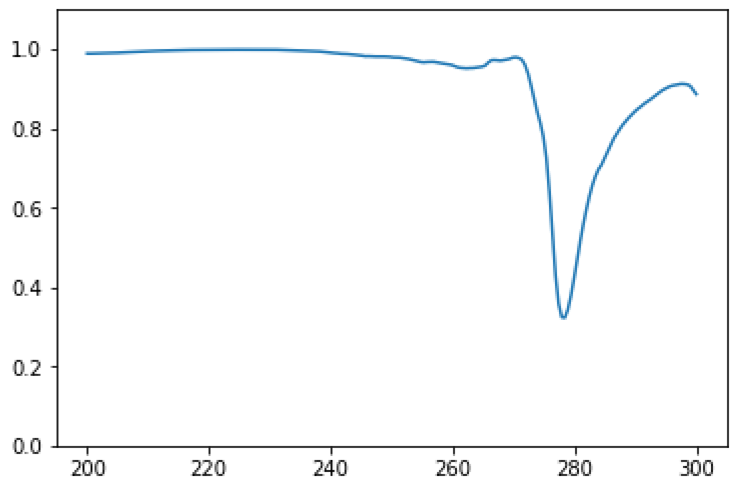
```
Predicted spectrum:
```



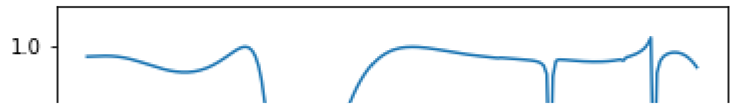
```
Test 73
True spectrum:
```



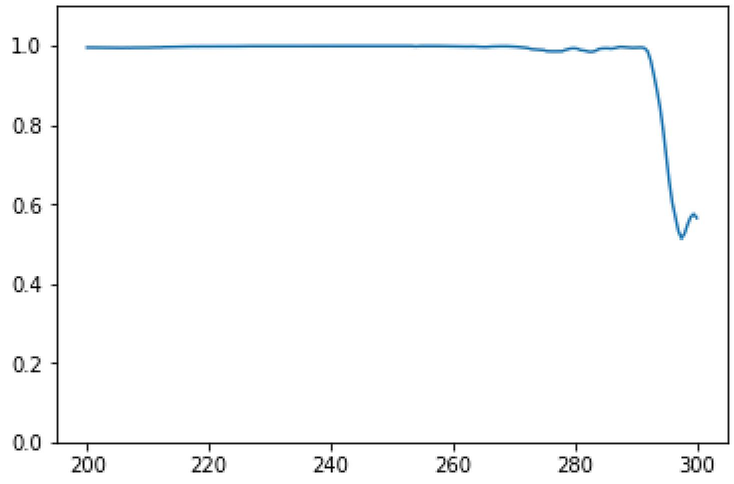
```
Predicted spectrum:
```



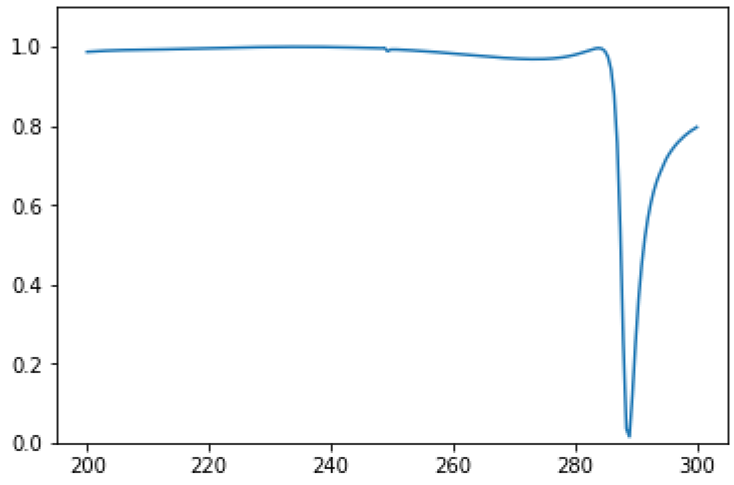
```
Test 74
True spectrum:
```



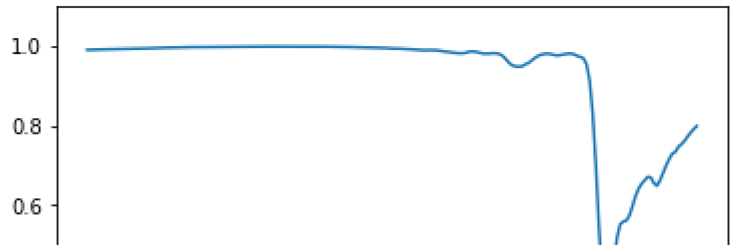
```
Predicted spectrum:
```



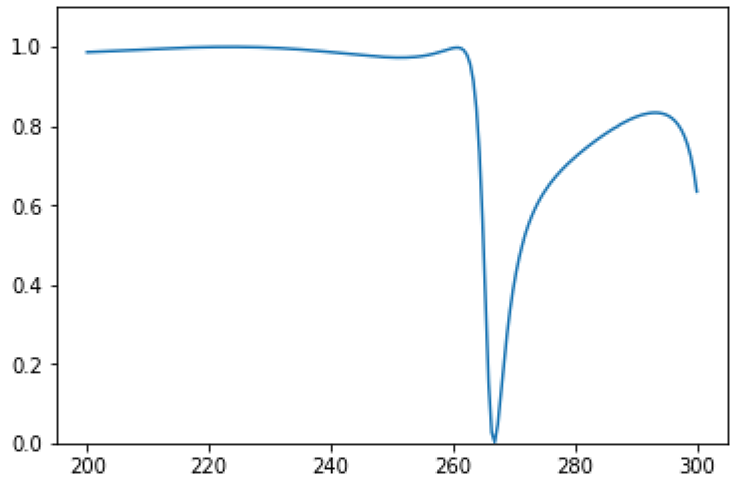
```
Test 75
True spectrum:
```



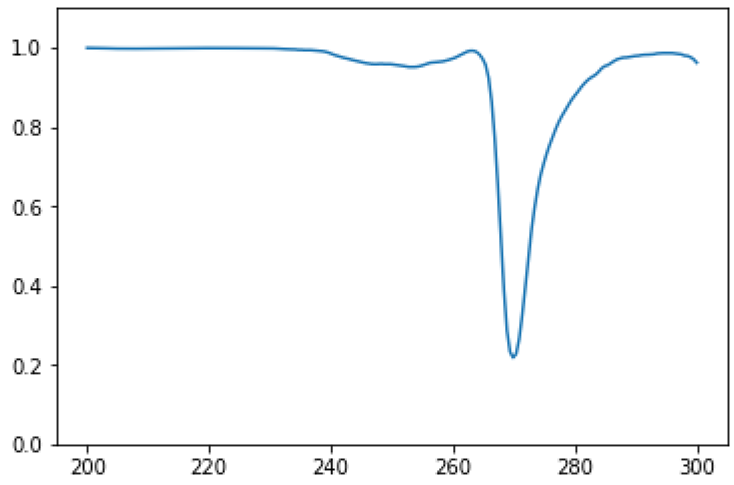
```
Predicted spectrum:
```



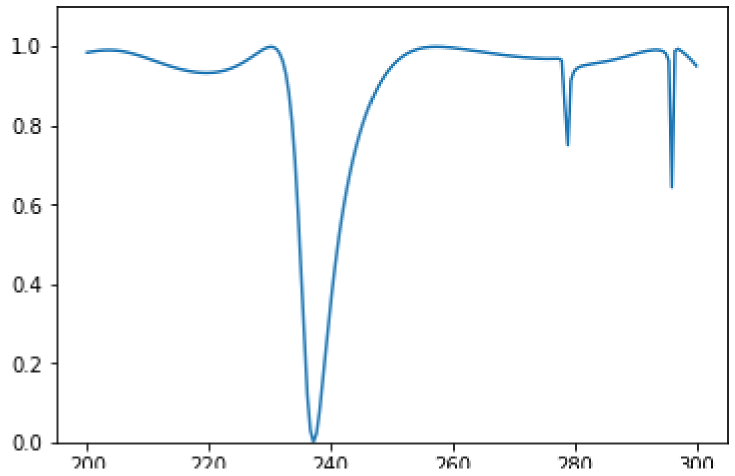
```
Test 76
True spectrum:
```



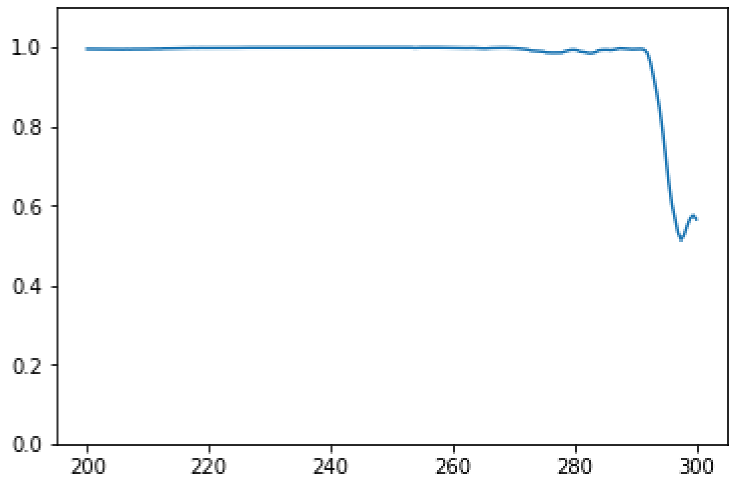
```
Predicted spectrum:
```



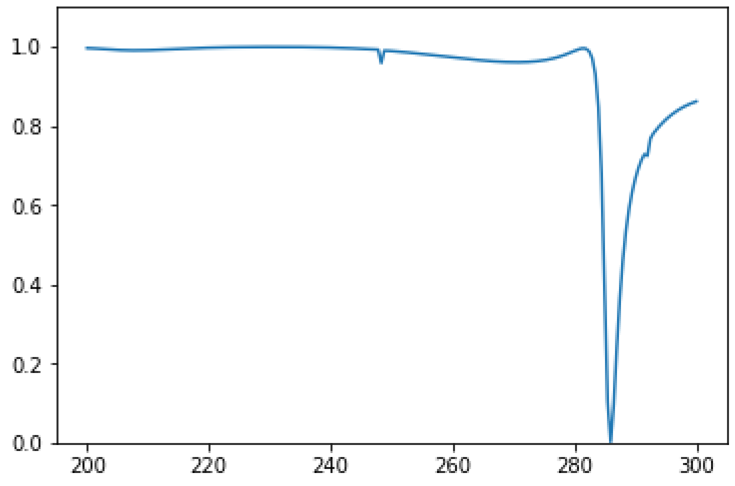
```
Test 77
True spectrum:
```



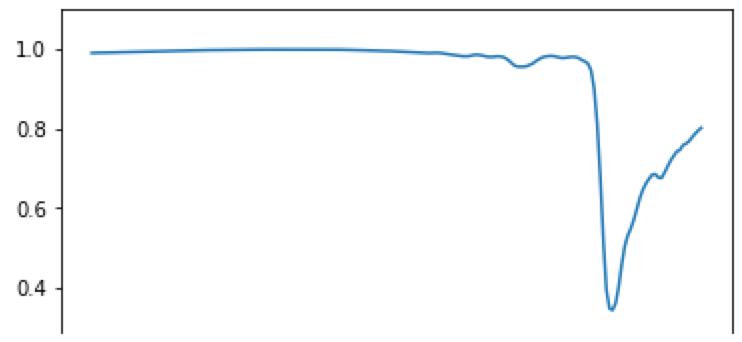
```
Predicted spectrum:
```



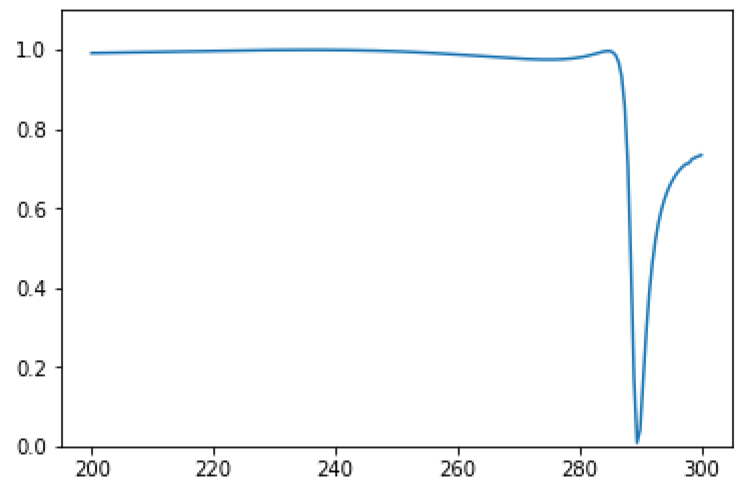
```
Test 78
True spectrum:
```



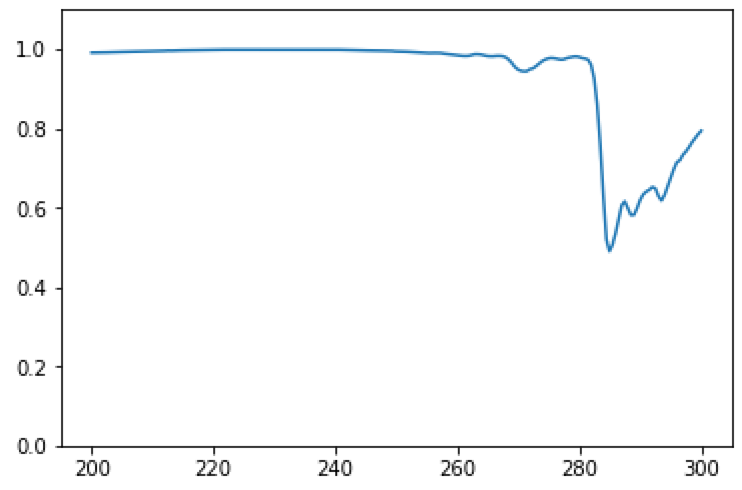
```
Predicted spectrum:
```



```
Test 79
True spectrum:
```

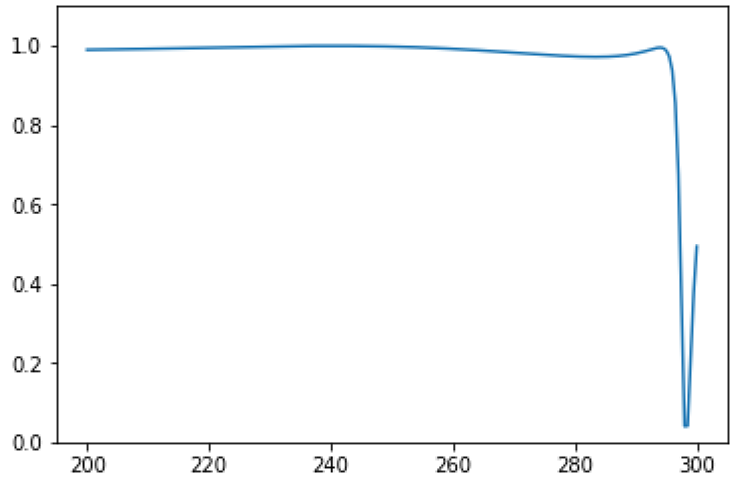


```
Predicted spectrum:
```

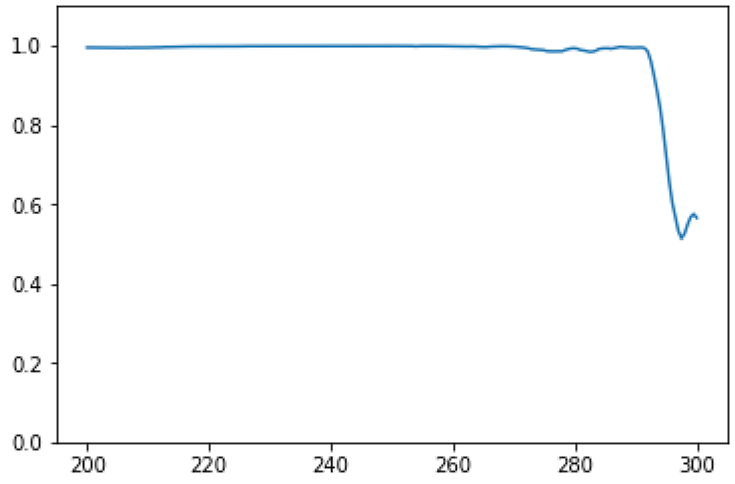


```
Test 80
```

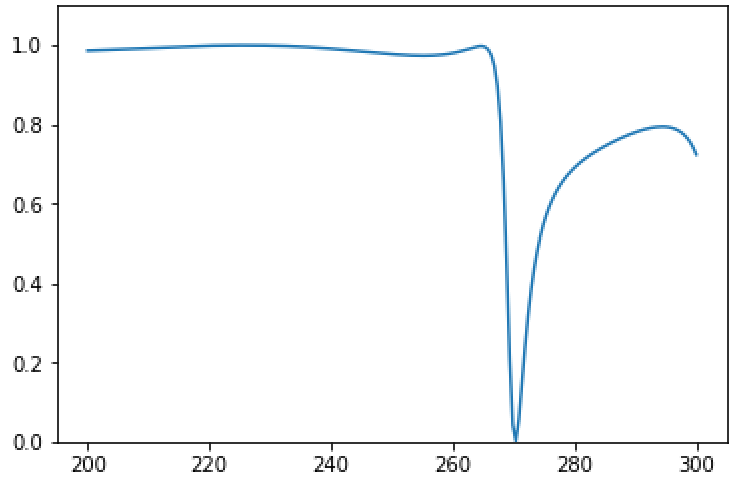
```
True spectrum:
```



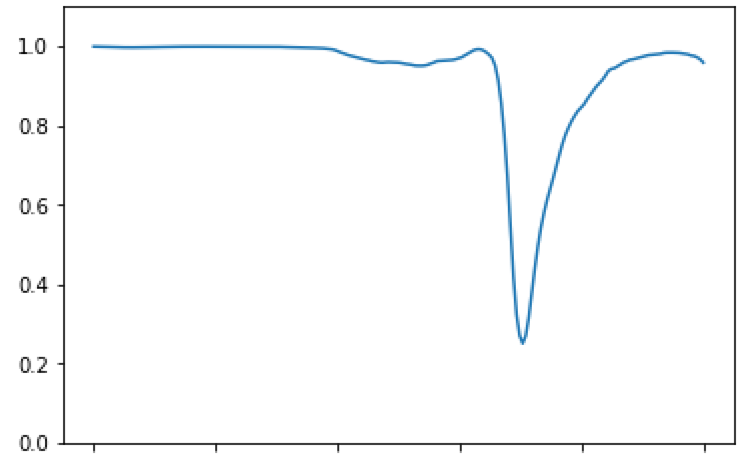
```
Predicted spectrum:
```



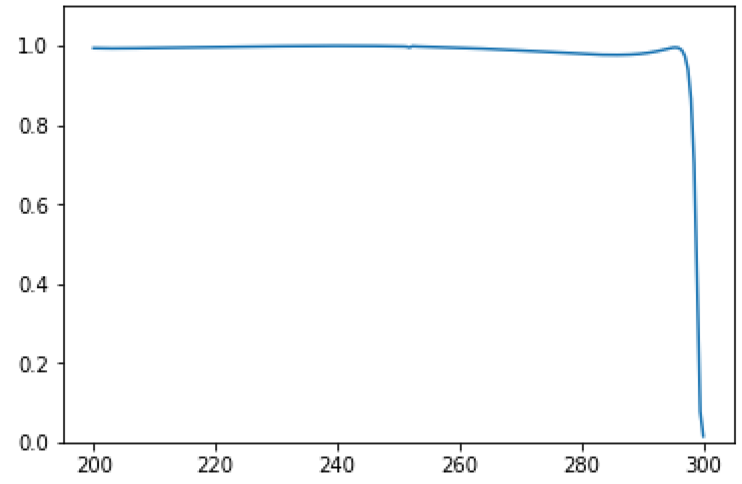
```
Test 81
True spectrum:
```



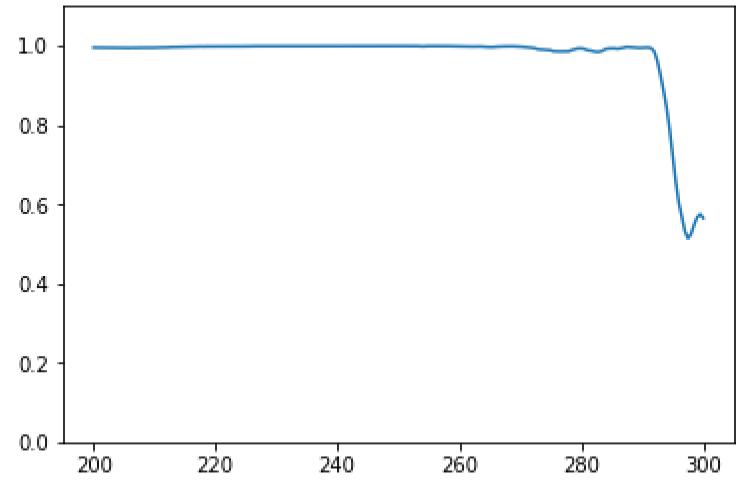
```
Predicted spectrum:
```



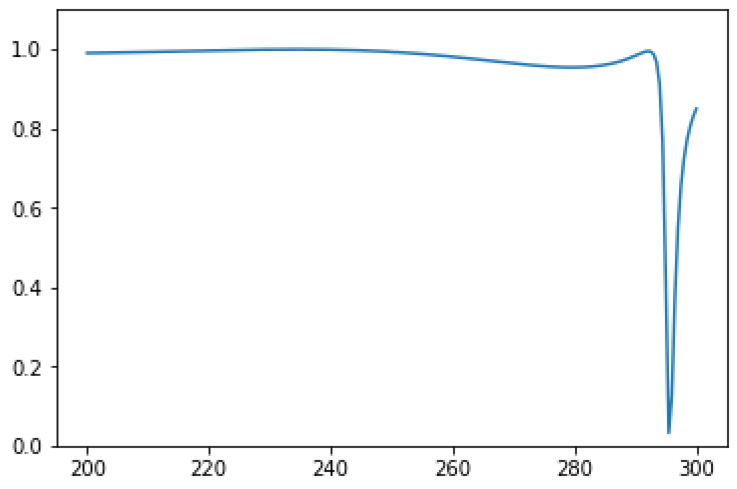
```
Test 82
True spectrum:
```



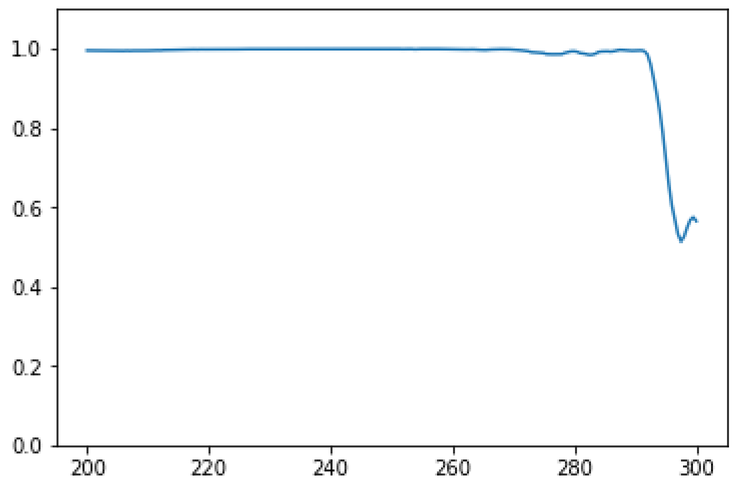
```
Predicted spectrum:
```



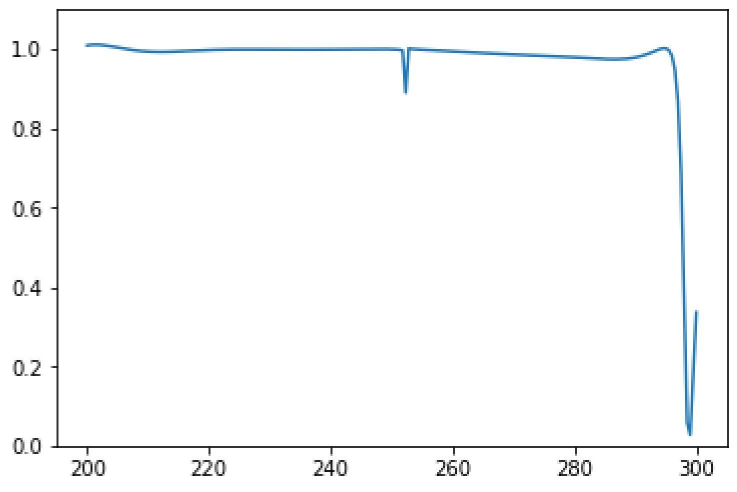
```
Test 83
True spectrum:
```



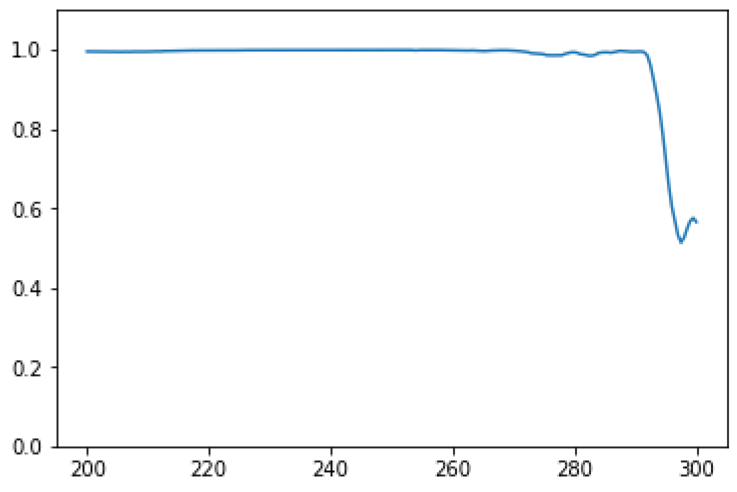
```
Predicted spectrum:
```



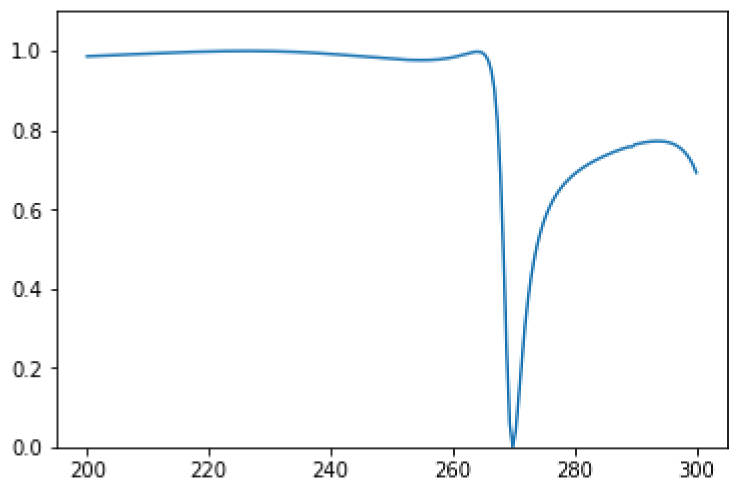
```
Test 84
True spectrum:
```



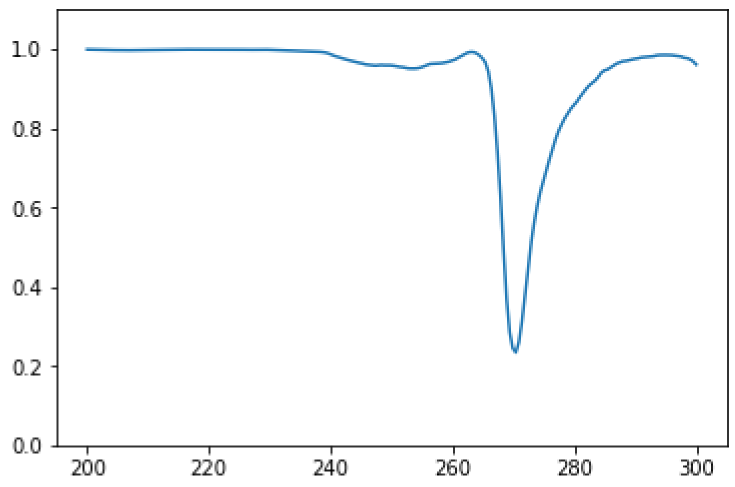
```
Predicted spectrum:
```



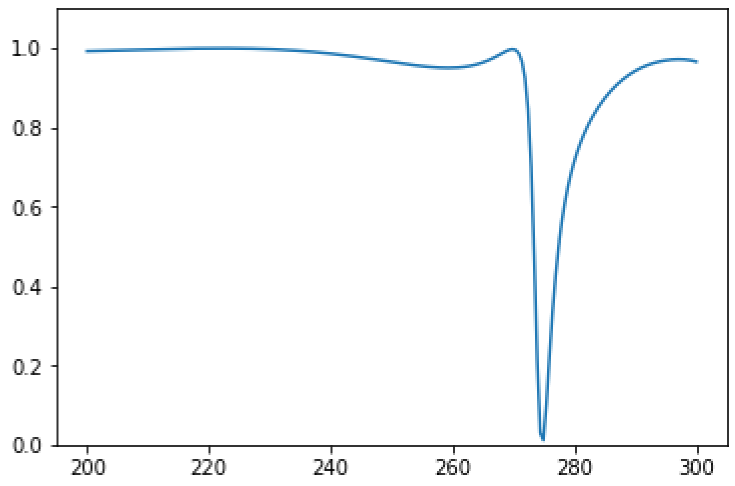
```
Test 85
True spectrum:
```



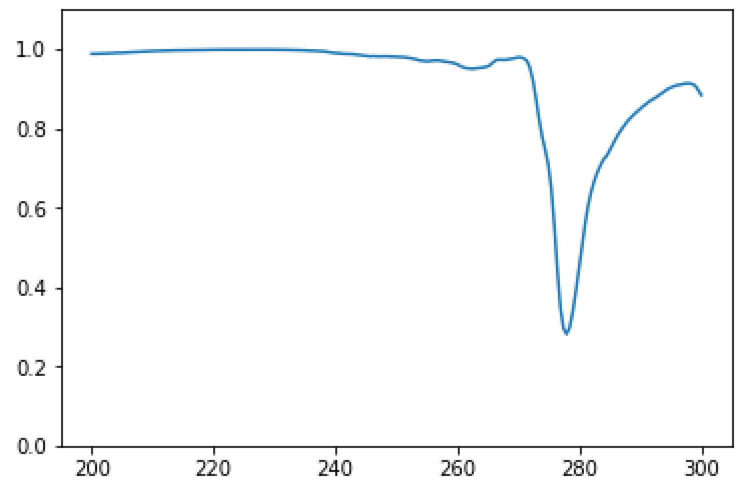
```
Predicted spectrum:
```



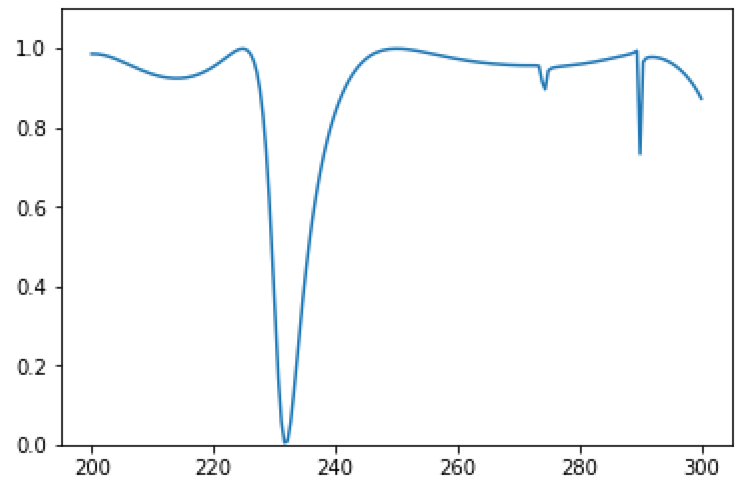
```
Test 86
True spectrum:
```



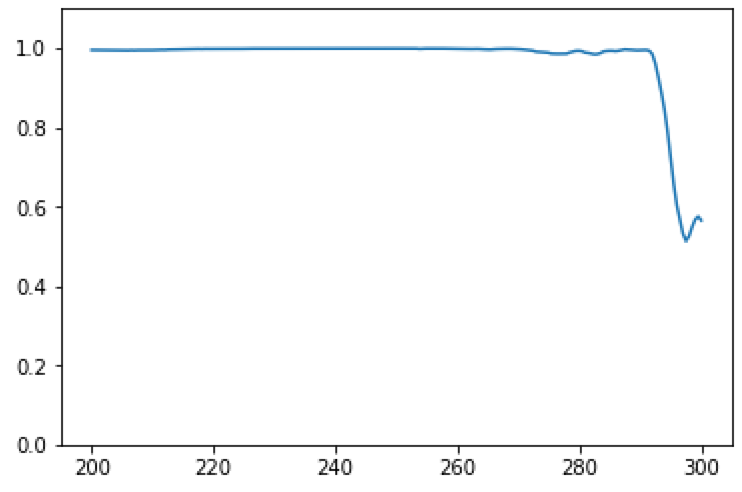
```
Predicted spectrum:
```



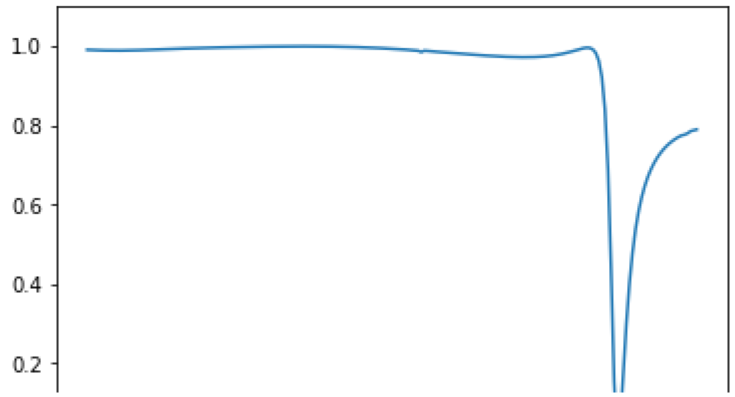
```
Test 87
True spectrum:
```



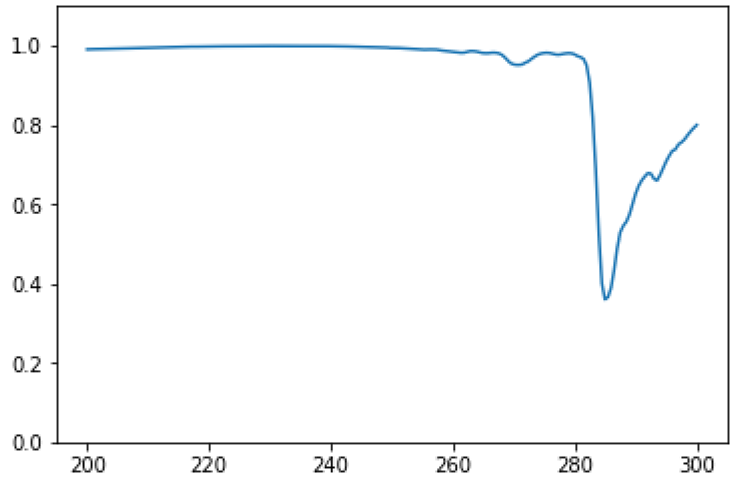
```
Predicted spectrum:
```



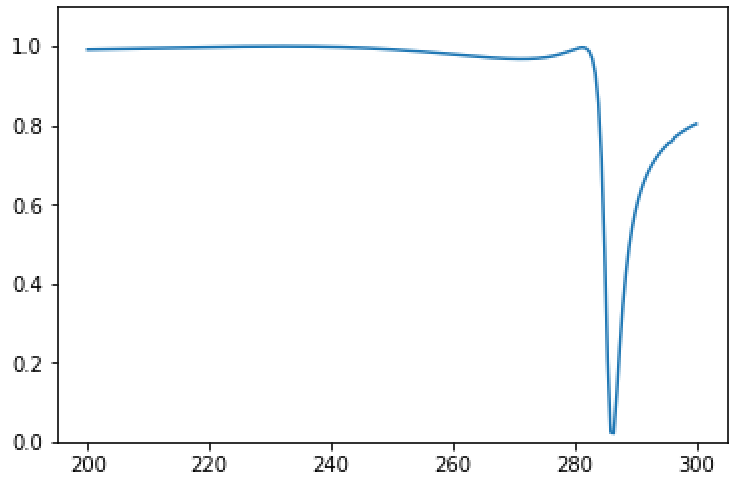
```
Test 88
True spectrum:
```



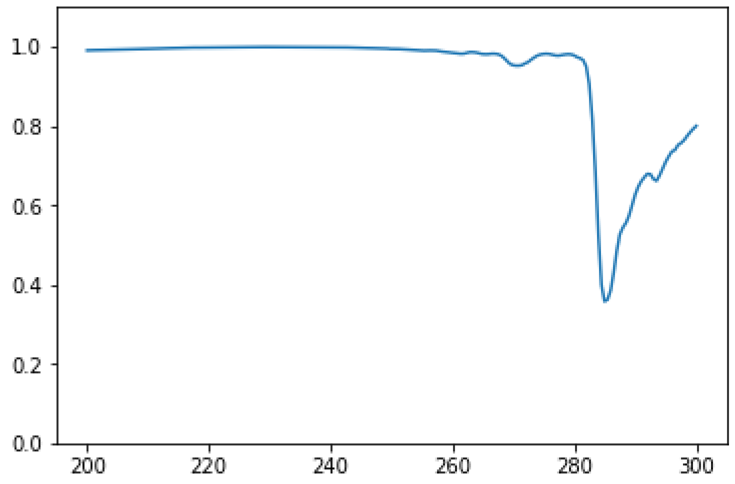
```
Predicted spectrum:
```



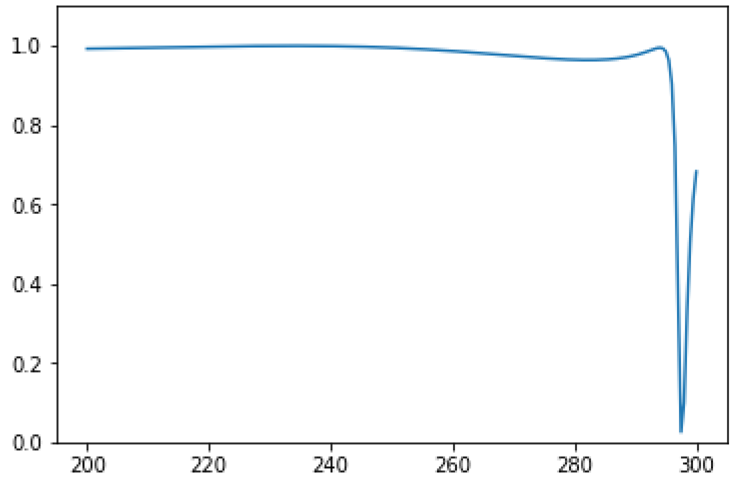
```
Test 89
True spectrum:
```



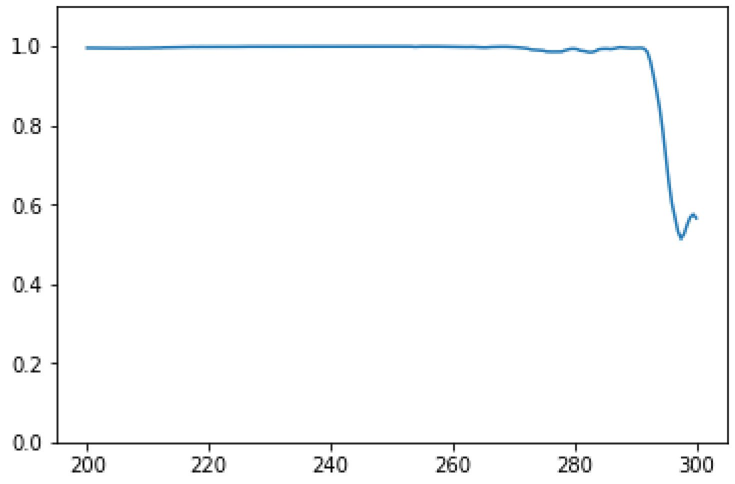
```
Predicted spectrum:
```



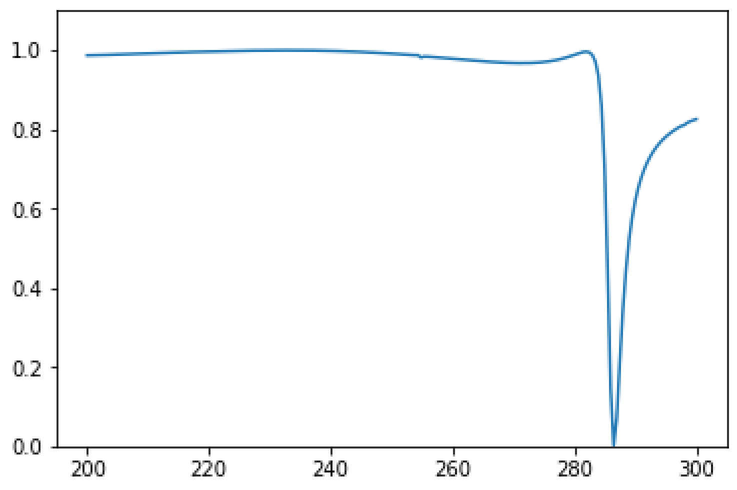
```
Test 90
True spectrum:
```



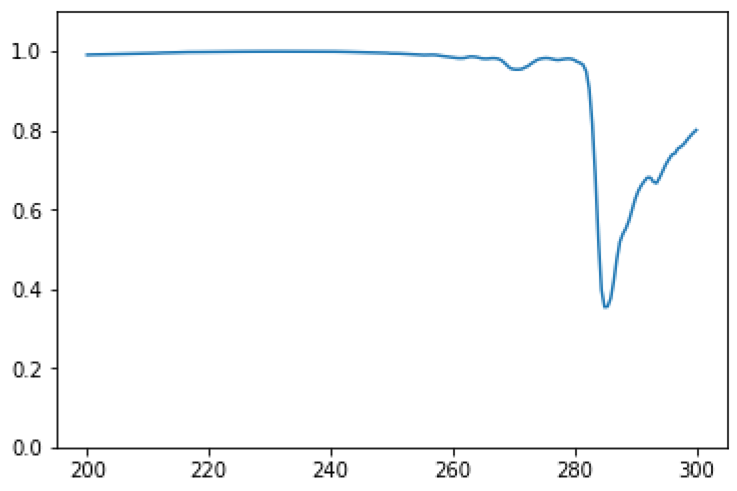
```
Predicted spectrum:
```



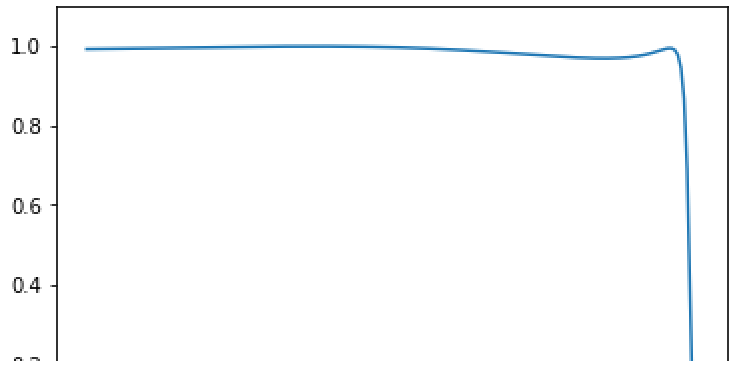
```
Test 91
True spectrum:
```



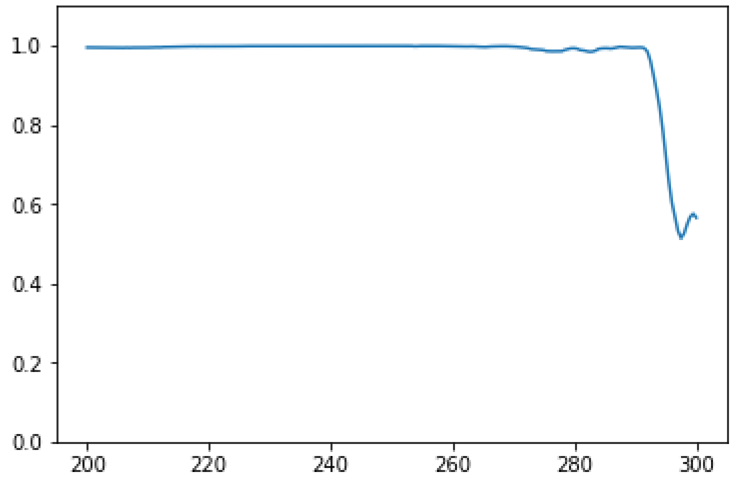
```
Predicted spectrum:
```



```
Test 92
True spectrum:
```



```
Predicted spectrum:
```

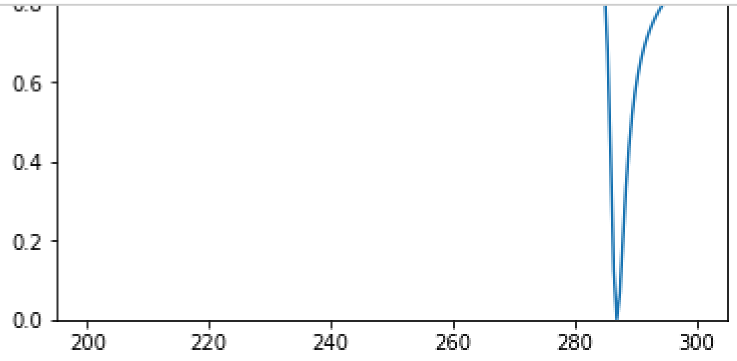


In [42]:
```
inverse = Sequential()
for layer in tandem.layers[:5]:
    layer.trainable=False
    inverse.add(layer)
```

In [43]:
```
inverse.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_36 (Dense)             (None, 512)               102912
_________________________________________________________________
dropout_29 (Dropout)         (None, 512)               0
_________________________________________________________________
dense_37 (Dense)             (None, 256)               131328
_________________________________________________________________
dropout_30 (Dropout)         (None, 256)               0
_________________________________________________________________
dense_38 (Dense)             (None, 6)                 1542
=================================================================
Total params: 235,782
Trainable params: 0
Non-trainable params: 235,782
_________________________________________________________________
```

In [46]:
```
predicted = []
dfnn_o = load_model('DFNN_O')
x = np.genfromtxt('meep_code/data/SP_xaxis.txt')
for i in range(len(test_Y)):
    print('Test '+str(i))
    print('True spectrum: ')
    plt.ylim(0, 1.1)
    plt.plot(x, test_Y[i])
    plt.show()
    print('Predicted spectrum: ')
    print(inverse.predict(np.reshape(test_Y[i], (1, 200))))
    predicted.append(*inverse.predict(np.reshape(test_Y[i], (1, 200))))
    print(dfnn_o.predict(np.reshape(test_Y[i], (1, 200))))
    predicted.append(*dfnn_o.predict(np.reshape(test_Y[i], (1, 200))))
```



```
Predicted spectrum:
[[4.2351186e-03 0.0000000e+00 0.0000000e+00 9.9975860e-01 1.6891352e-06
  5.0451337e-07]]
[[0.09126508 0.04418209 0.08624175 0.07967344 0.11641663 0.07916234]]
Test 4
True spectrum:
```

In [45]:
```
simulator.save('T_S')
tandem.save('T_T')
inverse.save('T_I')
```

In [ ]:
```
predicted
```

In [ ]:
```
np.savetxt('prediction.txt', predicted)
```