TASK 1

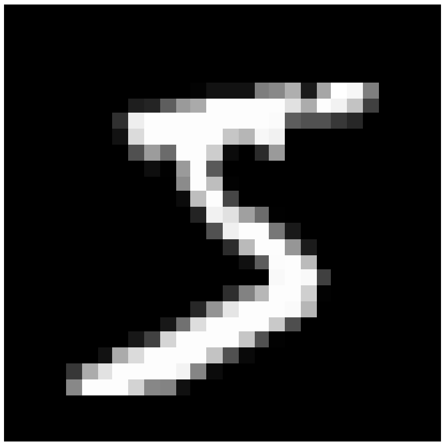
NAME: YEO ZHENG XU ISAAC

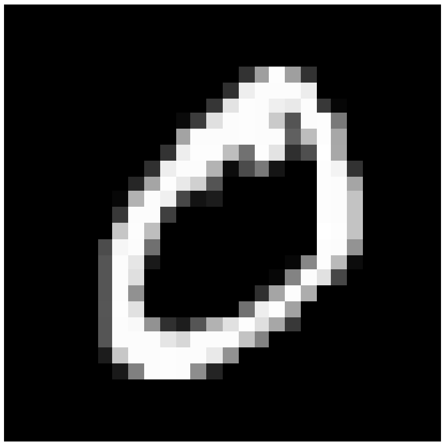
In [1]:
```
import pandas as pd
import numpy as np
```

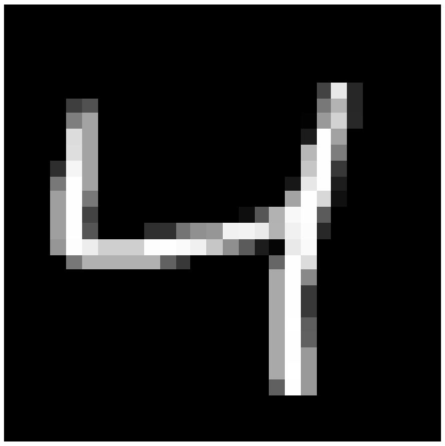
In [3]:
```
from sklearn.datasets import fetch_openml
your_data_home = "C:/Users/Isaac Yeo/Desktop"
mnist = fetch_openml('mnist_784', version=1, data_home=your_data_home)
X = mnist["data"]
```

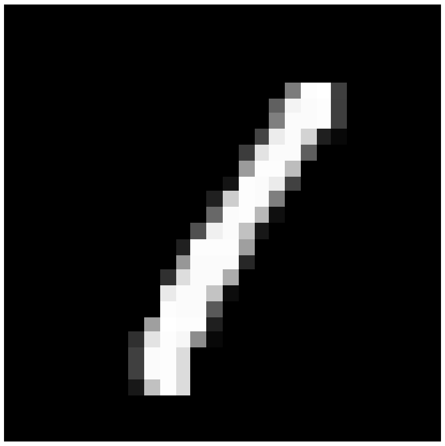
In [4]:
```
import matplotlib as mpl
import matplotlib.pyplot as plt
```

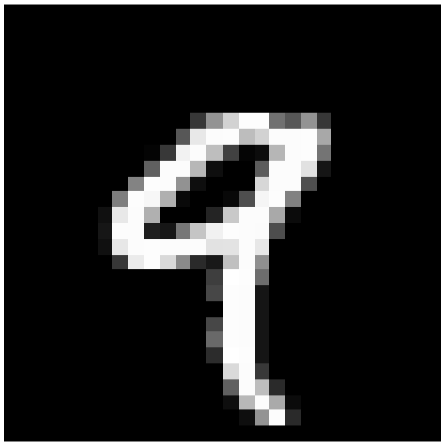
In [5]:
```
for i in range(0,5):
    some_digit = X[i]
    pixels = some_digit.reshape((28,28))
    plt.imshow(pixels, cmap = "gray" , interpolation = "nearest")
    plt.axis("off")
    plt.show()
```











In [6]:
```
x , y = mnist["data"], mnist["target"]
print(" Shape of x:",x.shape,"\n","Shape of y:",y.shape)
```

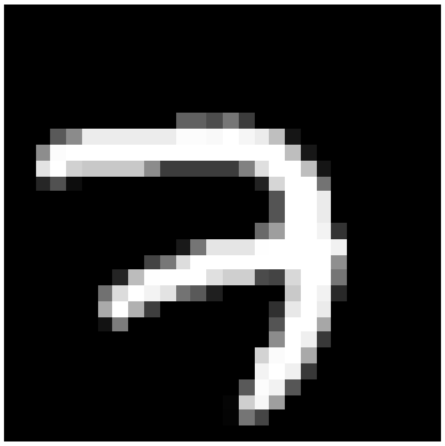
```
 Shape of x: (70000, 784)
 Shape of y: (70000,)
```

In [7]:
```
y[5000]
```

Out[7]:
```
'7'
```

In [8]:
```
 image_5000 = X[5000].reshape((28,28))
plt.axis("off")
plt.imshow(image_5000 , cmap = "gray" ,interpolation = "nearest")
```

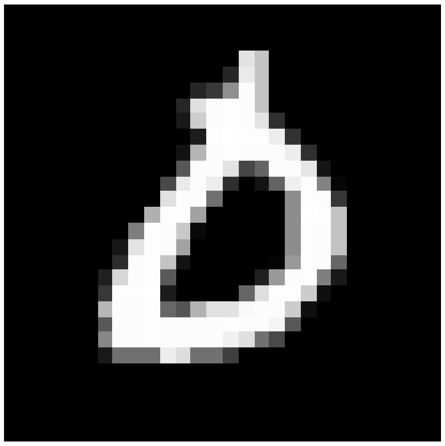
Out[8]:
```
<matplotlib.image.AxesImage at 0x13280809b08>
```



In [9]:
```
X_train, X_test , y_train, y_test = X[:60000] , X[60000:], y[:60000],y[60000:]
```

In [10]:
```
shuffle_index = np.random.permutation(60000)
X_train,y_train = X_train[shuffle_index], y_train[shuffle_index]
```

In [11]:
```
X_1 = X_train[0].reshape((28,28))
plt.imshow(X_1, cmap = "gray" , interpolation = "nearest")
plt.axis("off")
plt.show()
```



In [12]:
```
y_train[0]
```

Out[12]:
```
'0'
```

In [15]:
```
from sklearn.linear_model import SGDClassifier
SGD_classifier = SGDClassifier(random_state = 42)
SGD_classifier.fit(X_train, y_train)
prediction = SGD_classifier.predict([some_digit])
```

In [16]:
```
print(prediction)
```

```
['9']
```

In [17]:
```
from sklearn import tree
from sklearn import metrics
model = tree.DecisionTreeClassifier()
model.fit(X_train,y_train)
predict = model.predict(X_test)
```

In [18]:
```
accuracy = metrics.accuracy_score(y_test, predict)
accuracy
```

Out[18]:
```
0.877
```

In [19]:
```
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics
```

In [20]:
```
KNN_model = KNeighborsClassifier()
KNN_model.fit(X_train,y_train)
KNN_predict = KNN_model.predict(X_test)
KNN_accuracy = metrics.accuracy_score(y_test,KNN_predict)
KNN_accuracy
```

Out[20]:
```
0.9688
```

In [21]:
```
from sklearn.ensemble import RandomForestClassifier
import time
```

In [28]:
```
start_time = time.time()
RF_model = RandomForestClassifier(n_estimators=8)
RF_model.fit(X_train,y_train)
RF_predict = RF_model.predict(X_test)
RF_accuracy = metrics.accuracy_score(y_test,RF_predict)
print("RandomForestClassifier Accuracy :" ,RF_accuracy,
     "training time :",(time.time()-start_time))
```

```
RandomForestClassifier Accuracy : 0.9442 training time : 4.336768388748169
```

In [ ]: