**Puzzle 1**

**Explanation:**

Importing pandas so that we can import data and use Excel file format.

Create a variable to store the read\_csv() function which gives the data from csv-sample.csv and lets us read the data and label.

Print the message and use the built-in function len() which returns the length of an object.

**Code:**

**import pandas as pd**

**rows = pd.read\_csv('csv-sample.csv')**

**print("Number of rows :",len(rows))**

**Output:**

****

**Puzzle 2**

Sorry, I wasn’t able to complete this puzzle.

**Dockerfile:**

**FROM mcr.microsoft.com/dotnet/sdk:6.0 AS build**

**WORKDIR /src**

**COPY myCoreAPI.csproj .**

**RUN dotnet restore "myCoreAPI.csproj"**

**COPY . .**

**RUN dotnet publish "myCoreAPI.csproj" -c Release -o /publish**

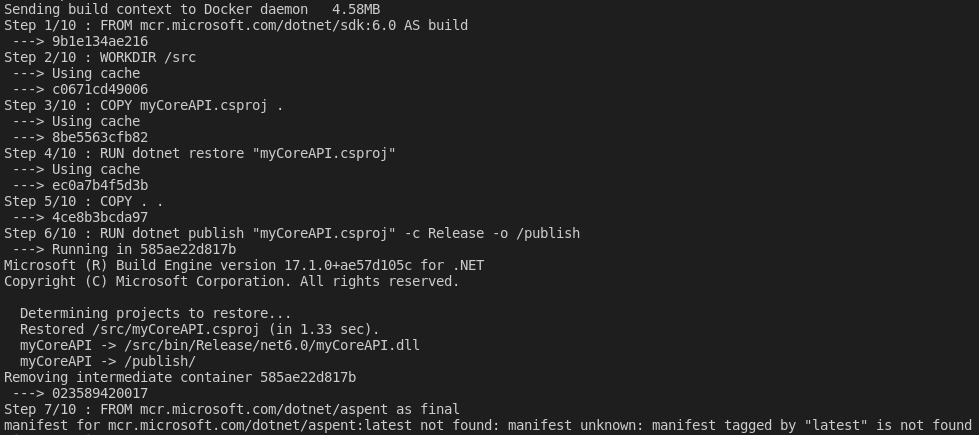
**FROM mcr.microsoft.com/dotnet/aspent as final**

**WORKDIR /app**

**COPY --from=build /publish .**

**ENTRYPOINT ["dotnet","myCoreAPI.dll"]**

**Output:**

****

**Puzzle 3**

**Puzzle Sample 1**

**Explanation:**

From the 1st clue, we know that among 1, 4, 7 there is a possibility of them being in the code.

From the 1st, 2nd & 3rd clues we know that 9, 6, 7, 8 are the two numbers in the code with 9 being in the last position of the code.

In the 2nd, 3rd, 4th, and 5th clues, we can eliminate 1, 8, 5, 2, 3, 4 so, 6 has the possibility to be in the 1st position.

So, in the 1st clue, 1 & 4 are already eliminated and 6 & 9 already took the 1st and 3rd position of the code so, 7 lies in the 2nd position of the code.

Now, we know that the final combination of the padlock is **679.**

**Code:**

**def checknr(number, checknumber, nr\_correct,nr\_correct\_position):**

**count\_nr\_correct = 0**

**count\_nr\_correct\_position = 0**

**lnumber = str(number).zfill(len(checknumber))**

**for i in range(len(checknumber)):**

**if lnumber[i] in checknumber:**

**count\_nr\_correct += 1**

**if lnumber[i] == checknumber[i]:**

**count\_nr\_correct\_position += 1**

**return nr\_correct == count\_nr\_correct and \**

**nr\_correct\_position == count\_nr\_correct\_position**

**for cnumber in range (1000):**

**if checknr(cnumber, '147', 1,0) and \**

**checknr(cnumber, '189', 1,1) and \**

**checknr(cnumber, '964', 2,0) and \**

**checknr(cnumber, '532', 0,0) and \**

**checknr(cnumber, '286', 1,0) :**

**print("Code found: " + str(cnumber).zfill(3))**

**Output:**

****

**Puzzle Sample 2**

**Explanation:**

From the 1st clue, 6, 8 & 2 have the possibility of beginning in the code among them one of their places is correct.

In the 1st, 2nd, 3rd, and 4th clues, we can eliminate 6, 8 now, we know that 2, 0, 1, 4 with 2 taking the last position of the code.

From the 3rd, and 5th clues, 2 has already taken its place in the code and 0 doesn’t lie in the 2nd position so it takes the 1st position in the code.

From the 2nd clue, we know that 6, 1 is already eliminated and 0 and 2 have already taken their position so, 4 lie in the 2nd position in the code.

Now, we know that the final combination of the padlock is **042.**

**Code:**

**def checknr(number, checknumber, nr\_correct,nr\_correct\_position):**

**count\_nr\_correct = 0**

**count\_nr\_correct\_position = 0**

**lnumber = str(number).zfill(len(checknumber))**

**for i in range(len(checknumber)):**

**if lnumber[i] in checknumber:**

**count\_nr\_correct += 1**

**if lnumber[i] == checknumber[i]:**

**count\_nr\_correct\_position += 1**

**return nr\_correct == count\_nr\_correct and \**

**nr\_correct\_position == count\_nr\_correct\_position**

**for cnumber in range (1000):**

**if checknr(cnumber, '682', 1,1) and \**

**checknr(cnumber, '614', 1,0) and \**

**checknr(cnumber, '206', 2,0) and \**

**checknr(cnumber, '738', 0,0) and \**

**checknr(cnumber, '380', 1,0) :**

**print("Code found: " + str(cnumber).zfill(3))**

**Output:**

****