```python
import pandas as pd

# Load the CSV file
file_path = '/content/HeSfFhYssJvTCRgk.csv'
data = pd.read_csv(file_path)

# Display the first few rows
print(data.head())
```

```
     student_id  score
0             1   1018
1             2   1218
2             3    611
3             4    723
4             5    541
```

```python
from scipy.stats import binom

# Given parameters
n = 10  # number of trials (visitors)
p = 0.8  # probability of success (buying souvenirs)

# 1. Probability that every visitor will buy souvenirs (P(X = 10))
prob_all_buy = binom.pmf(10, n, p)
print(f"Probability that all visitors buy souvenirs: {prob_all_buy}")

# 2. Probability that a maximum of 7 visitors will buy souvenirs (P(X <= 7))
prob_max_7_buy = binom.cdf(7, n, p)
print(f"Probability that a maximum of 7 visitors buy souvenirs: {prob_max_7_buy}")
```

```
    Probability that all visitors buy souvenirs: 0.10737418240000006
    Probability that a maximum of 7 visitors buy souvenirs: 0.32220047359999987
```

part 2

```python
import pandas as pd

# Load the dataset (replace 'your_file_path.csv' with the actual file path)
data = pd.read_csv('/content/20sMGjG0yMJtVQGi (1).csv')

# Rename columns for easier access
data.columns = ['Bug ID', 'Time Taken']

# 1. Probability that debugging takes less than 3 hours
prob_less_than_3 = (data['Time Taken'] < 3).mean()
print(f"Probability that debugging takes less than 3 hours: {prob_less_than_3}")

# 2. Probability that debugging takes more than 2 hours
prob_more_than_2 = (data['Time Taken'] > 2).mean()
print(f"Probability that debugging takes more than 2 hours: {prob_more_than_2}")

# 3. 50th percentile (median) of debugging time
percentile_50 = data['Time Taken'].median()
print(f"50th percentile (median) of debugging time: {percentile_50}")
```

```
    Probability that debugging takes less than 3 hours: 0.49761677788369874
    Probability that debugging takes more than 2 hours: 0.7521448999046711
    50th percentile (median) of debugging time: 3.005
```

```python
import pandas as pd
import numpy as np

# Load the SAT score data
data = pd.read_csv('/content/HeSfFhYssJvTCRgk.csv')

# 1. Calculate the probability that a student scores less than 800
prob_less_than_800 = (data['score'] < 800).mean()

# 2. Calculate the probability that a student scores more than 1300
```

```
prob_more_than_1300 = (data['score'] > 1300).mean()


# 3. Calculate the minimum score a student must achieve to be in the 90th percentile
score_90th_percentile = np.percentile(data['score'], 90)


# 4. Calculate the minimum score a student must achieve to be in the top 5%
score_top_5_percentile = np.percentile(data['score'], 95)


# Print the results
print(f"Probability that a student scores less than 800: {prob_less_than_800}")
print(f"Probability that a student scores more than 1300: {prob_more_than_1300}")
print(f"Score required to be in the 90th percentile: {score_90th_percentile}")
print(f"Score required to be in the top 5%: {score_top_5_percentile}")
```

```
Probability that a student scores less than 800: 0.157
Probability that a student scores more than 1300: 0.068
Score required to be in the 90th percentile: 1269.0
Score required to be in the top 5%: 1338.1
```