PROGRAM 1

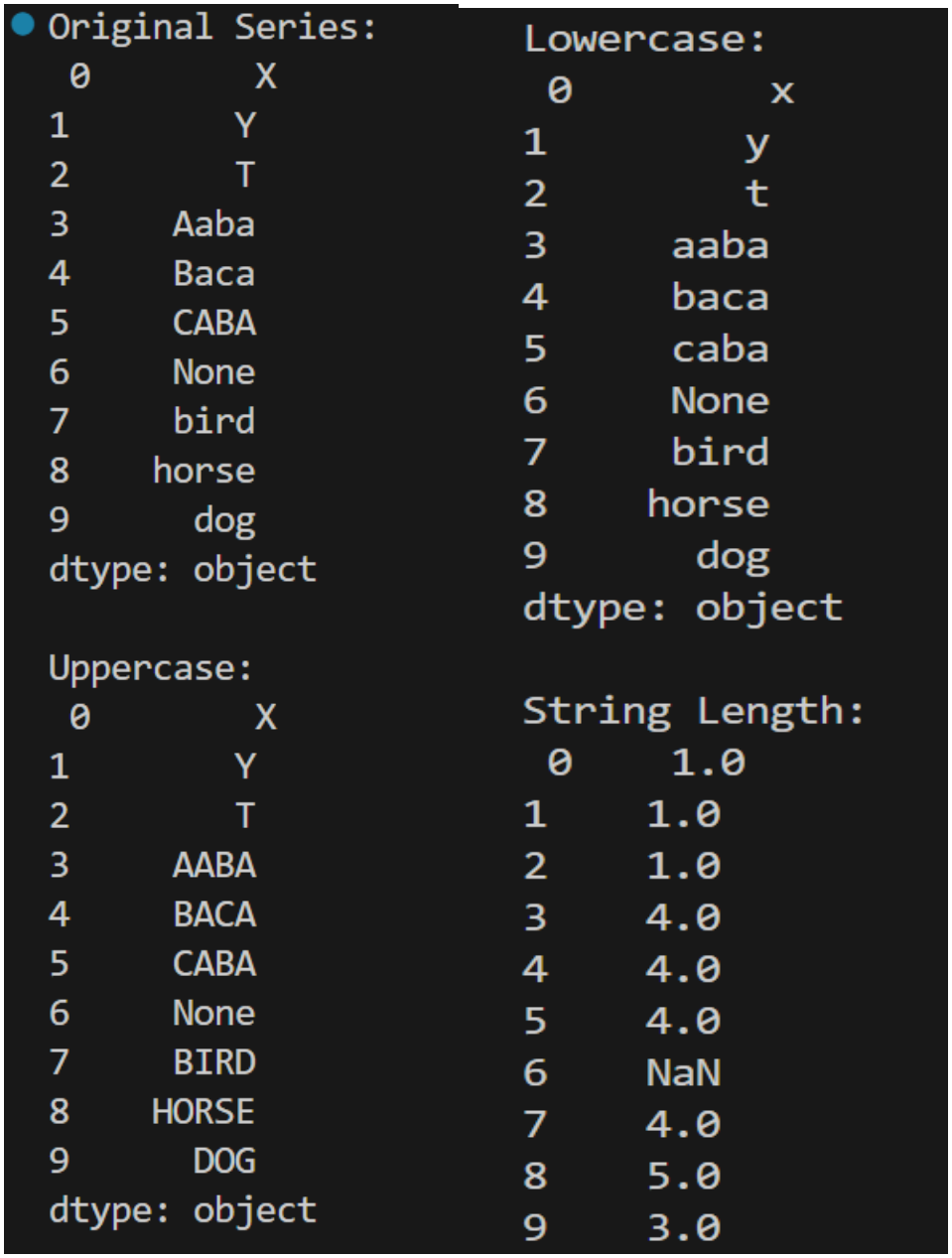
```
import pandas as pd
import datetime

# (a) DateTime object for Jan 15, 2012
date_a = pd.Timestamp("2012-01-15")

# (b) Specific date and time of 9:20 PM
date_b = pd.Timestamp("2012-01-15 21:20")

# (c) Local date and time
date_c = pd.Timestamp.now()

# (d) A date without time
date_d = date_c.date()

# (e) Current date
date_e = pd.Timestamp.today().date()

# (f) Time from a datetime
date_f = date_c.time()

# (g) Current local time
date_g = datetime.datetime.now().time()

# Display results
print("a) DateTime object for Jan 15, 2012:", date_a)
print("b) Specific date and time of 9:20 PM:", date_b)
print("c) Local date and time:", date_c)
print("d) A date without time:", date_d)
print("e) Current date:", date_e)
```

```
print("f) Time from a datetime:", date_f)

print("g) Current local time:", date_g)
```

```
a) DateTime object for Jan 15, 2012: 2012-01-15 00:00:00
b) Specific date and time of 9:20 PM: 2012-01-15 21:20:00
c) Local date and time: 2025-04-10 15:56:59.352864
d) A date without time: 2025-04-10
e) Current date: 2025-04-10
f) Time from a datetime: 15:56:59.352864
g) Current local time: 15:56:59.352889
```

PROGRAM 2

```
import pandas as pd


# Given Series

s = pd.Series(['X', 'Y', 'T', 'Aaba', 'Baca', 'CABA', None, 'bird', 'horse', 'dog'])


# Convert to Upper and Lower case

s_upper = s.str.upper()  # Convert to uppercase

s_lower = s.str.lower()  # Convert to lowercase


# Find Length of Strings (handling None values)

s_length = s.str.len()


# Display results

print("Original Series:\n", s)

print("\nUppercase:\n", s_upper)

print("\nLowercase:\n", s_lower)

print("\nString Length:\n", s_length)

# string serial 0 is one space ahead of all serials to indicate there are mixed datatypes because of None
```



PROGRAM 3

```
import pandas as pd


# Take user input for car asking prices

n = int(input("Enter the number of cars: "))


print("\nEnter the asking prices of the cars:")

asking_prices = pd.Series([int(input(f"Car {i+1} asking price: ")) for i in range(n)])
```

```
print("\nEnter the fair prices of the cars:")
fair_prices = pd.Series([int(input(f"Car {i+1} fair price: ")) for i in range(n)])

# Find indices where asking price is less than fair price
good_deals = asking_prices[asking_prices < fair_prices].index.tolist()

# Display results
print("\nGood deals found at indices:", good_deals)
```

```
Enter the number of cars: 3

Enter the asking prices of the cars:
Car 1 asking price: 25000
Car 2 asking price: 70000
Car 3 asking price: 10000

Enter the fair prices of the cars:
Car 1 fair price: 27000
Car 2 fair price: 68000
Car 3 fair price: 100100

Good deals found at indices: [0, 2]
```

PROGRAM 4

```
import pandas as pd

# Taking user input for John and Judy's schedule
john_schedule = []
judy_schedule = []

print("Enter '1' if John is visiting, '0' otherwise:")
for i in range(1, 11):
    john_schedule.append(int(input(f"Day {i} - John visiting? (1/0): ")))
```

```
print("\nEnter '1' if Judy is visiting, '0' otherwise:")
for i in range(1, 11):
    judy_schedule.append(int(input(f"Day {i} - Judy visiting? (1/0): ")))

# Create DataFrame
df = pd.DataFrame({
    "day": range(1, 11),
    "John": john_schedule,
    "Judy": judy_schedule
})

# Identify party days (when both John and Judy visit)
df["party"] = df["John"] & df["Judy"]

# Compute 'days_til_party'
df["days_til_party"] = (df.index.to_series()
                        .apply(lambda i: (df.loc[i:, "party"].idxmax() - i) if df.loc[i:, "party"].any() else None))

# Display the result
print("\nFinal Schedule with Days Until Next Party:")
print(df[["day", "John", "Judy", "days_til_party"]])
```

```
Final Schedule with Days Until Next Party:
   day  John  Judy  days_til_party
0    1     1     0               9
1    2     0     1               8
2    3     1     0               7
3    4     1     0               6
4    5     0     1               5
5    6     1     0               4
6    7     0     1               3
7    8     0     1               2
8    9     0     1               1
9   10     1     1               0
```

PROGRAM 5

```
import pandas as pd


# Sample dataset
data = {
    "artist": ["A", "B", "A", "C", "B", "A", "C", "A", "B"],
    "venue": ["X", "Y", "X", "Z", "Y", "X", "Z", "Y", "X"],
    "date": ["2024-01-15", "2024-01-22", "2024-02-10", "2024-02-18",
             "2024-03-05", "2024-03-15", "2024-04-10", "2024-04-15", "2024-04-20"]
}


# Convert to DataFrame
df = pd.DataFrame(data)


# Convert 'date' column to datetime and extract 'year-month'
df["date"] = pd.to_datetime(df["date"])
df["year_month"] = df["date"].dt.to_period("M")


# Count concerts per (artist, venue, year-month)
concert_counts = df.groupby(["year_month", "artist", "venue"]).size().reset_index(name="count")


# Pivot table to wide format
wide_table = concert_counts.pivot(index="year_month", columns=["artist", "venue"],
values="count").fillna(0)


# Flatten column names
wide_table.columns = [f"{a}_{v}" for a, v in wide_table.columns]
wide_table.reset_index(inplace=True)


# Print result
print(wide_table)
```

	year_month	A_X	B_Y	C_Z	A_Y	B_X
0	2024-01	1.0	1.0	0.0	0.0	0.0
1	2024-02	1.0	0.0	1.0	0.0	0.0
2	2024-03	1.0	1.0	0.0	0.0	0.0
3	2024-04	0.0	0.0	1.0	1.0	1.0