

Question 1:

Python Code : For 1st dataset

```
import numpy as np
import matplotlib.pyplot as plt

def find_peaks_and_valleys(data):
    """Find the indices of peaks and valleys in the dataset."""
    data = np.array(data)
    peaks = []
    valleys = []

    # Find peaks and valleys
    for i in range(1, len(data) - 1):
        if data[i] > data[i - 1] and data[i] > data[i + 1]:
            peaks.append(i)
        elif data[i] < data[i - 1] and data[i] < data[i + 1]:
            valleys.append(i)

    return peaks, valleys

def plot_data(data, peaks, valleys, filename):
    """Plot the data and highlight peaks and valleys."""
    plt.figure(figsize=(10, 6))
    plt.plot(data, label='Signal', color='black')
    plt.plot(peaks, np.array(data)[peaks], 'ro', label='Peaks')
    plt.plot(valleys, np.array(data)[valleys], 'bo', label='Valleys')
    plt.title('Signal with Peaks and Valleys')
    plt.xlabel('Index')
    plt.ylabel('Value')
    plt.legend()
    plt.grid(True)
    plt.savefig(filename)
    plt.show()

# Load data from file
def load_data(filename):
    """Load data from a text file."""
    with open(filename, 'r') as file:
        data = [float(line.strip()) for line in file]
    return data

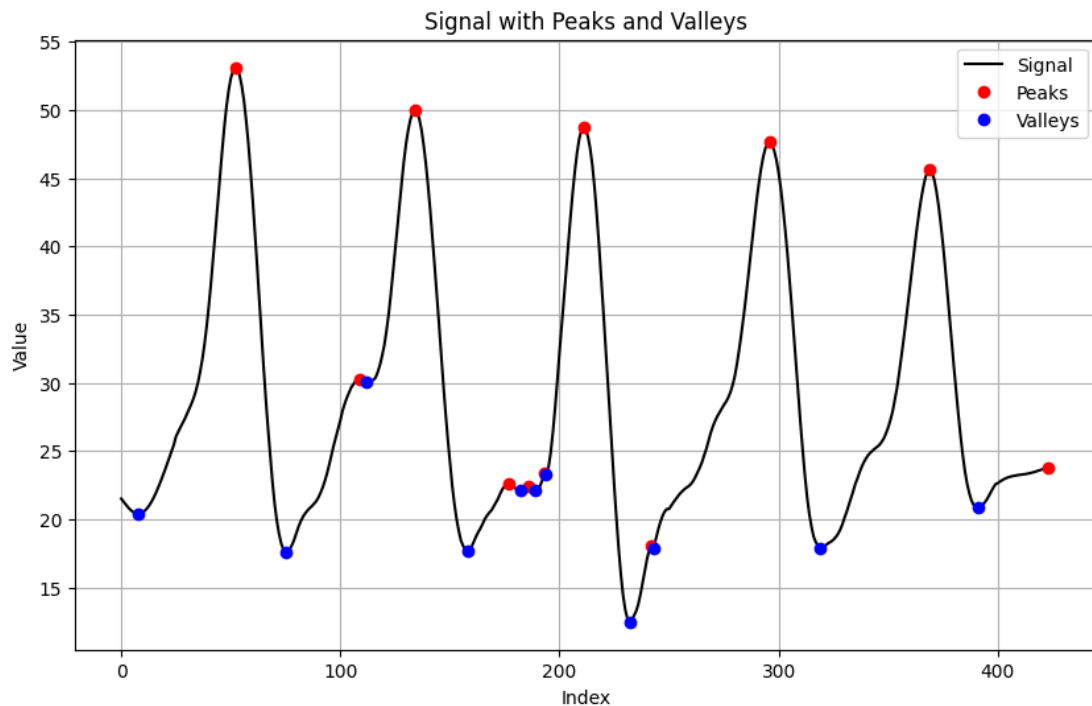
# Example usage
data_file = 'Data_1.txt' # Replace with your data file name
data = load_data(data_file)
peaks, valleys = find_peaks_and_valleys(data)

# Print indices of peaks and valleys
```

```
print("Indices of peaks:", peaks)
print("Indices of valleys:", valleys)
```

```
# Plot data with peaks and valleys
plot_data(data, peaks, valleys, 'plot.png')
```

```
Indices of peaks: [52, 109, 134, 177, 186, 193, 211, 242, 296, 369, 423]
Indices of valleys: [8, 75, 112, 158, 182, 189, 194, 232, 243, 319, 391]
```



For 2nd Dataset:

```
import numpy as np
import matplotlib.pyplot as plt

def find_peaks_and_valleys(data):
    """Find the indices of peaks and valleys in the dataset."""
    data = np.array(data)
    peaks = []
    valleys = []

    # Find peaks and valleys
    for i in range(1, len(data) - 1):
        if data[i] > data[i - 1] and data[i] > data[i + 1]:
            peaks.append(i)
        elif data[i] < data[i - 1] and data[i] < data[i + 1]:
            valleys.append(i)

    return peaks, valleys

def plot_data(data, peaks, valleys, filename):
```

```

    """Plot the data and highlight peaks and valleys."""
    plt.figure(figsize=(10, 6))
    plt.plot(data, label='Signal', color='black')
    plt.plot(peaks, np.array(data)[peaks], 'ro', label='Peaks')
    plt.plot(valleys, np.array(data)[valleys], 'bo', label='Valleys')
    plt.title('Signal with Peaks and Valleys')
    plt.xlabel('Index')
    plt.ylabel('Value')
    plt.legend()
    plt.grid(True)
    plt.savefig(filename)
    plt.show()

# Load data from file
def load_data(filename):
    """Load data from a text file."""
    with open(filename, 'r') as file:
        data = [float(line.strip()) for line in file]
    return data

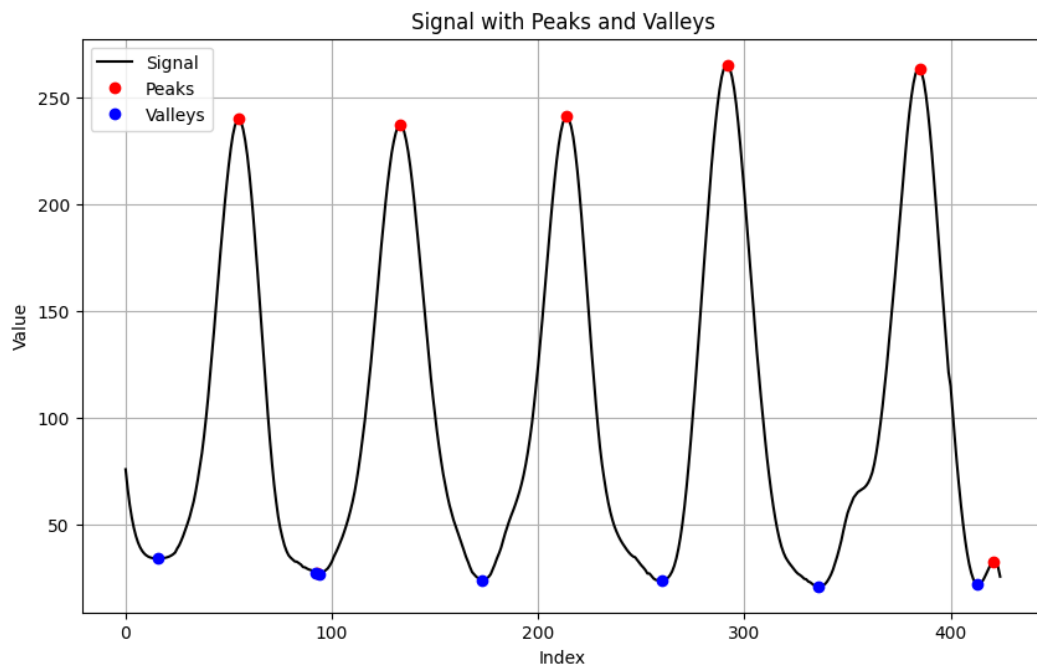
# Example usage
data_file = 'Data_2.txt' # Replace with your data file name
data = load_data(data_file)
peaks, valleys = find_peaks_and_valleys(data)

# Print indices of peaks and valleys
print("Indices of peaks:", peaks)
print("Indices of valleys:", valleys)

# Plot data with peaks and valleys
plot_data(data, peaks, valleys, 'plot.png')

```

Indices of peaks: [55, 93, 133, 214, 292, 385, 421]
Indices of valleys: [16, 92, 94, 173, 260, 336, 413]



C CODE :

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
// Function to find peaks and valleys
```

```
void find_peaks_and_valleys(const double *data, int length, int *peaks, int *num_peaks, int *valleys, int *num_valleys) {
```

```
    *num_peaks = 0;
```

```
    *num_valleys = 0;
```

```
    for (int i = 1; i < length - 1; i++) {
```

```
        if (data[i] > data[i - 1] && data[i] > data[i + 1]) {
```

```
            peaks[(*num_peaks)++] = i;
```

```
        } else if (data[i] < data[i - 1] && data[i] < data[i + 1]) {
```

```
            valleys[(*num_valleys)++] = i;
```

```
        }
```

```
    }
```

```
}
```

```
int main() {
```

```
    // Open data files
```

```
    FILE *file1 = fopen("Data_1.txt", "r");
```

```
    FILE *file2 = fopen("Data_2.txt", "r");
```

```
    if (file1 == NULL || file2 == NULL) {
```

```
        perror("Error opening file");
```

```
        return 1;
```

```
    }
```

```
    // Read data into arrays
```

```
    double data1[1000], data2[1000]; // Adjust size as needed
```

```
    int length1 = 0, length2 = 0;
```

```
    while (fscanf(file1, "%lf", &data1[length1++]) != EOF);
```

```
    while (fscanf(file2, "%lf", &data2[length2++]) != EOF);
```

```
    fclose(file1);
```

```
    fclose(file2);
```

```
    // Find peaks and valleys
```

```
    int peaks1[1000], valleys1[1000], num_peaks1, num_valleys1;
```

```
    int peaks2[1000], valleys2[1000], num_peaks2, num_valleys2;
```

```
    find_peaks_and_valleys(data1, length1, peaks1, &num_peaks1, valleys1, &num_valleys1);
```

```
    find_peaks_and_valleys(data2, length2, peaks2, &num_peaks2, valleys2, &num_valleys2);
```

```
    // Output results
```

```
    printf("Data Set 1:\nMaxima: ");
```

```

for (int i = 0; i < num_peaks1; i++) printf("%d ", peaks1[i]);

printf("\nMinima: ");

for (int i = 0; i < num_valleys1; i++) printf("%d ", valleys1[i]);

printf("\n");

printf("Data Set 2:\nMaxima: ");

for (int i = 0; i < num_peaks2; i++) printf("%d ", peaks2[i]);

printf("\nMinima: ");

for (int i = 0; i < num_valleys2; i++) printf("%d ", valleys2[i]);

printf("\n");

return 0;
}

```

The screenshot shows a Replit IDE with a C program. The program reads two data files, "Data 1.txt" and "Data 2.txt", and processes them to find peaks and valleys. The code is as follows:

```

8
9
10 for (int i = 1; i < length - 1; i++) {
11     if (data[i] > data[i - 1] && data[i] > data[i + 1]) {
12         peaks[(*num_peaks)++] = i;
13     } else if (data[i] < data[i - 1] && data[i] < data[i + 1]) {
14         valleys[(*num_valleys)++] = i;
15     }
16 }
17
18 int main() {
19     // Open data files
20     FILE *file1 = fopen("Data 1.txt", "r");
21     FILE *file2 = fopen("Data 2.txt", "r");
22
23     if (file1 == NULL || file2 == NULL) {
24         perror("Error opening file");
25         return 1;
26     }
27
28     // Read data into arrays
29     double data1[1000], data2[1000]; // Adjust size as needed
30     int length1 = 0, length2 = 0;
31
32     while (fscanf(file1, "%lf", &data1[length1++]) != EOF);
33     while (fscanf(file2, "%lf", &data2[length2++]) != EOF);

```

The output of the program is shown in the console:

```

Data Set 1:
Maxima: 52 189 134 177 186 193 211 242 296 369 423
Minima: 8 75 112 158 182 189 194 232 243 319 391
Data Set 2:
Maxima: 55 93 133 214 292 385 421
Minima: 16 92 94 173 268 336 413

```