M.HARINI

21ADR015

KONGU ENGINEERING COLLEGE, Perudurai

**Table of content:**

|  |  |
| --- | --- |
| **Content** | **Page no** |
| **Python code** | **1-6** |
| **C Code** | **6-7** |

**Figure Tabel:**

|  |  |
| --- | --- |
| **Figure name** | **Page number** |
| Fig 1: Peaks signal 1 output | 2 |
| Fig 2: Peaks signal 2output | 3 |
| Fig 3: Indices for peak signal 1 | 3 |
| Fig 4: Indices for peak signal 2 | 4 |
| Fig 5 : C programme output | 6 |

**PYTHON CODE :**

import numpy as np

import matplotlib.pyplot as plt

from scipy.signal import find\_peaks

signal\_1 = np.loadtxt('Data\_1.txt')

signal\_2 = np.loadtxt('Data\_2.txt')

peaks\_max\_1, \_ = find\_peaks(signal\_1)

peaks\_max\_2, \_ = find\_peaks(signal\_2)

peaks\_min\_1, \_ = find\_peaks(-signal\_1)

peaks\_min\_2, \_ = find\_peaks(-signal\_2)

plt.figure(figsize=(10, 6))

plt.plot(signal\_1, label='Signal 1', color='black')

plt.plot(peaks\_max\_1, signal\_1[peaks\_max\_1], 'ro', label='Maxima (Red)')

plt.plot(peaks\_min\_1, signal\_1[peaks\_min\_1], 'bo', label='Minima (Blue)')

plt.title('Peaks in Signal 1')

plt.xlabel('Index')

plt.ylabel('Signal Amplitude')

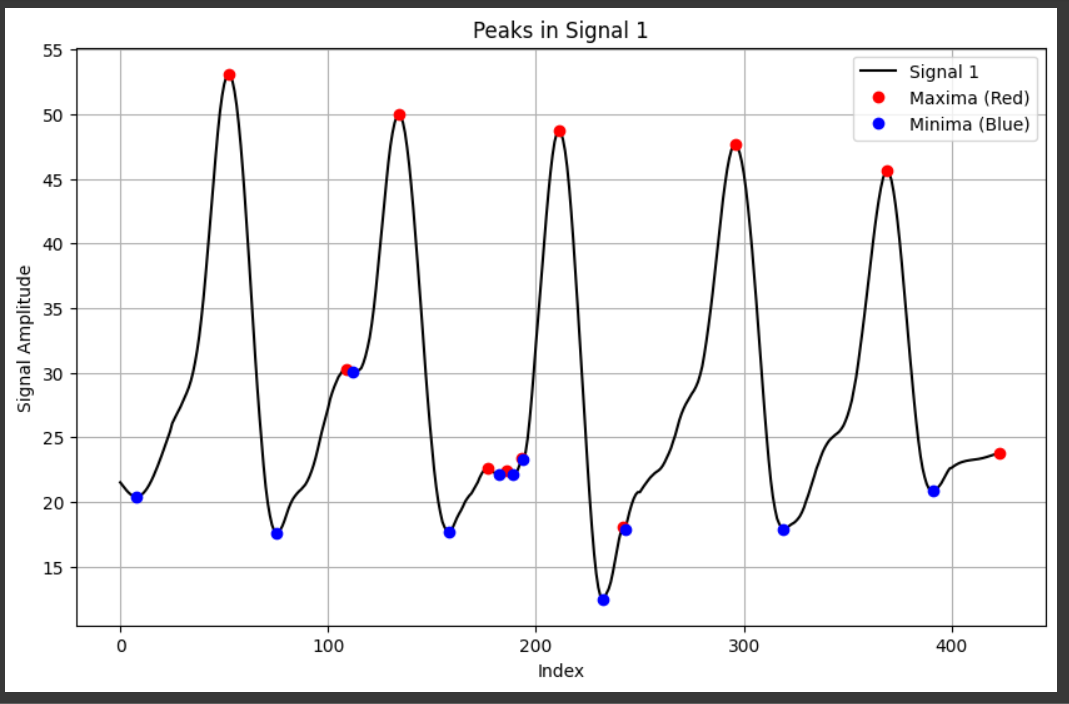
plt.legend()

plt.grid(True)

plt.show()

**1**

**OUTPUT 1:**

****

**Fig 1: Peaks signal 1 output**

plt.figure(figsize=(10, 6))

plt.plot(signal\_2, label='Signal 2', color='black')

plt.plot(peaks\_max\_2, signal\_2[peaks\_max\_2], 'ro', label='Maxima (Red)')

plt.plot(peaks\_min\_2, signal\_2[peaks\_min\_2], 'bo', label='Minima (Blue)')

plt.title('Peaks in Signal 2')

plt.xlabel('Index')

plt.ylabel('Signal Amplitude')

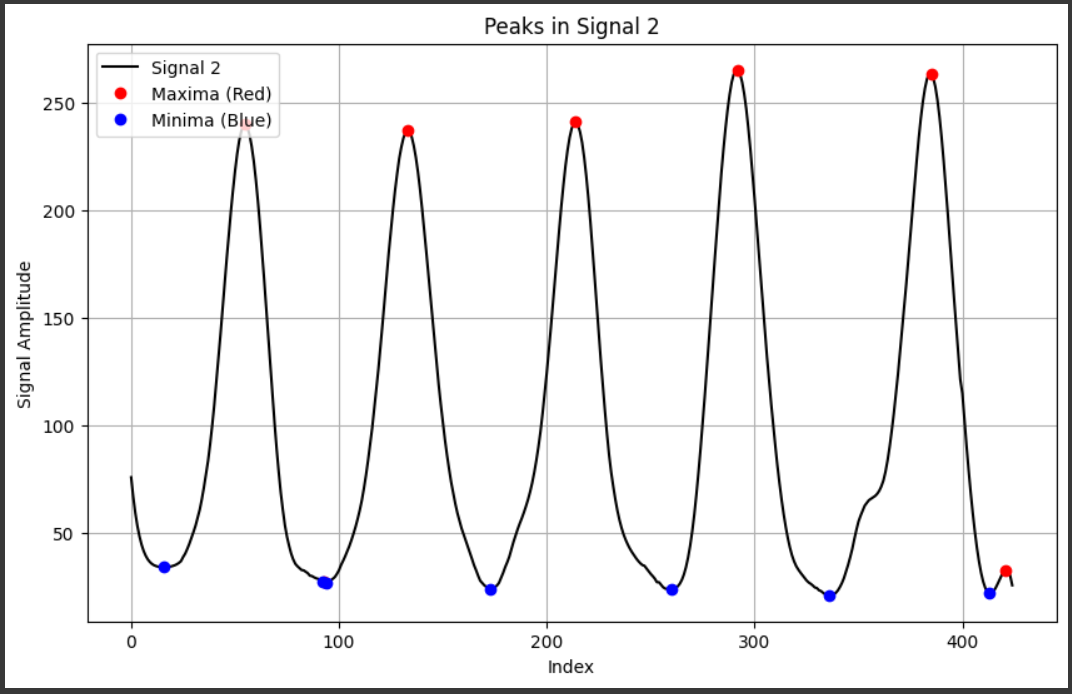
plt.legend()

plt.grid(True)

plt.show()

**2**

**OUTPUT 2:**

****

**Fig 2: Peaks signal 2output**

print("Maxima indices for Signal 1:", peaks\_max\_1)

print("Minima indices for Signal 1:", peaks\_min\_1)

**OUTPUT 3:**



**Fig 3: Indices for peak signal 1**

print("Maxima indices for Signal 2:", peaks\_max\_2)

print("Minima indices for Signal 2:", peaks\_min\_2)

**3**

**OUTPUT 4:**

****

**Fig 4: Indices for peak signal 2**

**C code :**

#include <stdio.h>

#include <stdlib.h>

#define MAX\_LENGTH 10000

int load\_data(const char \*filename, double data[], int max\_length) {

FILE \*file = fopen(filename, "r");

if (file == NULL) {

perror("Error opening file");

return -1;

}

int count = 0;

while (count < max\_length && fscanf(file, "%lf", &data[count]) == 1) {

count++;

}

fclose(file);

return count;

}

void print\_maxima(double data[], int n, const char \*dataset\_name) {

printf("Maxima in %s:\n", dataset\_name);

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

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

printf("Maxima at index %d, value: %f\n", i, data[i]);

}

**4**

}

void print\_minima(double data[], int n, const char \*dataset\_name) {

printf("Minima in %s:\n", dataset\_name);

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

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

printf("Minima at index %d, value: %f\n", i, data[i]);

}

}

}

void find\_peaks(double data[], int n, const char \*dataset\_name) {

print\_maxima(data, n, dataset\_name);

print\_minima(data, n, dataset\_name);

}

int main() {

double data1[MAX\_LENGTH], data2[MAX\_LENGTH];

int count1 = load\_data("Data\_1.txt", data1, MAX\_LENGTH);

if (count1 == -1) return 1;

int count2 = load\_data("Data\_2.txt", data2, MAX\_LENGTH);

if (count2 == -1) return 1;

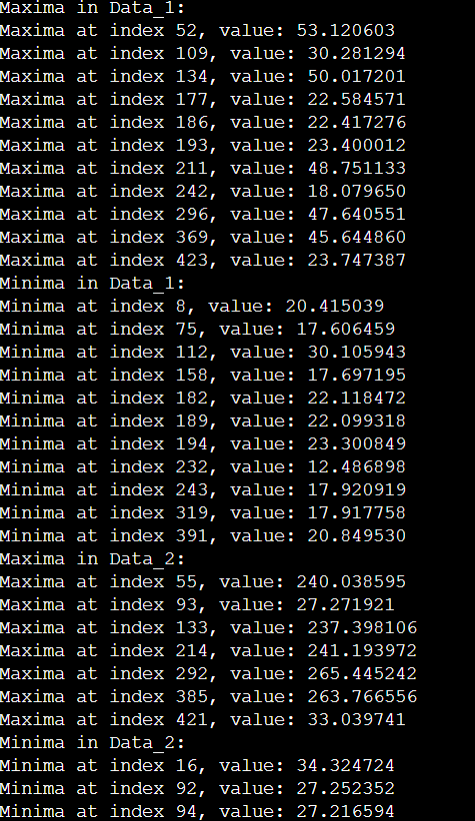
find\_peaks(data1, count1, "Data\_1");

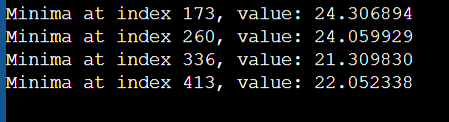
find\_peaks(data2, count2, "Data\_2");

return 0}

**5**

**OUTPUT 5:**





**Fig 5 : C programme output**

**6**