**Question 1  
 C Program**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include <stdbool.h>

#define MAX\_DATA\_SIZE 1000

#define PLOT\_WIDTH 70

#define PLOT\_HEIGHT 20

void load\_data(const char \*filename, double \*data, int \*size) {

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

if (file == NULL) {

perror("Error opening file");

exit(EXIT\_FAILURE);

}

\*size = 0;

while (fscanf(file, "%lf", &data[(\*size)++]) != EOF);

fclose(file);

}

void find\_peaks(double \*data, int size, int \*peaks, int \*num\_peaks, int distance, double prominence) {

\*num\_peaks = 0;

for (int i = distance; i < size - distance; i++) {

bool is\_peak = true;

for (int j = 1; j <= distance; j++) {

if (data[i] <= data[i - j] || data[i] <= data[i + j]) {

is\_peak = false;

break;

}

}

if (is\_peak && data[i] > prominence) {

peaks[(\*num\_peaks)++] = i;

}

}

}

void plot\_ascii(double \*data, int size, int \*peaks, int num\_peaks, int \*minima, int num\_minima, const char \*label) {

double max\_val = data[0], min\_val = data[0];

for (int i = 1; i < size; i++) {

if (data[i] > max\_val) max\_val = data[i];

if (data[i] < min\_val) min\_val = data[i];

}

char plot[PLOT\_HEIGHT][PLOT\_WIDTH];

for (int i = 0; i < PLOT\_HEIGHT; i++) {

for (int j = 0; j < PLOT\_WIDTH; j++) {

plot[i][j] = ' ';

}

}

for (int i = 0; i < size; i++) {

int x = (i \* PLOT\_WIDTH) / size;

int y = (int)((data[i] - min\_val) / (max\_val - min\_val) \* (PLOT\_HEIGHT - 1));

plot[PLOT\_HEIGHT - 1 - y][x] = '\*';

}

for (int i = 0; i < num\_peaks; i++) {

int x = (peaks[i] \* PLOT\_WIDTH) / size;

int y = (int)((data[peaks[i]] - min\_val) / (max\_val - min\_val) \* (PLOT\_HEIGHT - 1));

plot[PLOT\_HEIGHT - 1 - y][x] = 'M';

}

for (int i = 0; i < num\_minima; i++) {

int x = (minima[i] \* PLOT\_WIDTH) / size;

int y = (int)((data[minima[i]] - min\_val) / (max\_val - min\_val) \* (PLOT\_HEIGHT - 1));

plot[PLOT\_HEIGHT - 1 - y][x] = 'm';

}

printf("\n%s\n", label);

for (int i = 0; i < PLOT\_HEIGHT; i++) {

for (int j = 0; j < PLOT\_WIDTH; j++) {

printf("%c", plot[i][j]);

}

printf("\n");

}

}

void find\_and\_plot\_peaks(double \*data, int size, const char \*label) {

int peaks[MAX\_DATA\_SIZE], num\_peaks;

double inversed\_data[MAX\_DATA\_SIZE];

find\_peaks(data, size, peaks, &num\_peaks, 20, 0.5);

for (int i = 0; i < size; i++) {

inversed\_data[i] = -data[i];

}

int minima[MAX\_DATA\_SIZE], num\_minima;

find\_peaks(inversed\_data, size, minima, &num\_minima, 20, 0.5);

// Plot the data, peaks, and minima using ASCII art

plot\_ascii(data, size, peaks, num\_peaks, minima, num\_minima, label);

}

int main() {

double data1[MAX\_DATA\_SIZE], data2[MAX\_DATA\_SIZE];

int size1, size2;

load\_data("Data\_1.txt", data1, &size1);

load\_data("Data\_2.txt", data2, &size2);

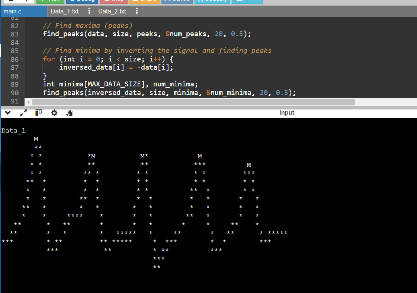
find\_and\_plot\_peaks(data1, size1, "Data\_1");

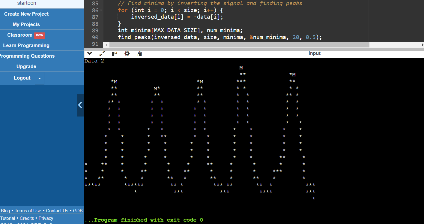
find\_and\_plot\_peaks(data2, size2, "Data\_2");

return 0;

}

**Output**

****

****

**Python code:**

import numpy as np

import matplotlib.pyplot as plt

from scipy.signal import find\_peaks

*# Load data*

data1 = np.loadtxt('/content/Data\_1.txt')

*# Define a function to find and plot peaks*

def find\_and\_plot\_peaks(data, label):

*# Find maxima (peaks)*

peaks, \_ = find\_peaks(data, distance=20, prominence=0.5)

*# Find minima by inverting the signal and finding peaks*

inversed\_data = -data

minima, = find\_peaks(inversed\_data, distance=20, prominence=0.5)

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

plt.plot(data, label=f'{label} Signal', color='black')

plt.plot(peaks, data[peaks], 'ro', label='Maxima')

plt.plot(minima, data[minima], 'bo', label='Minima')

plt.title(f'{label} Signal Peaks')

plt.xlabel('Index')

plt.ylabel('Amplitude')

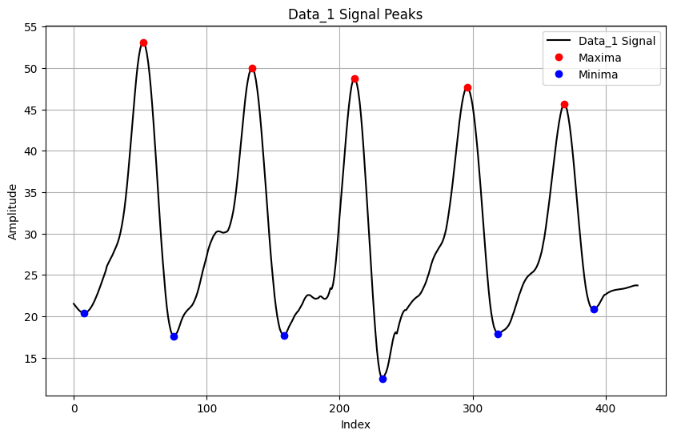
plt.legend()

plt.grid(True)

plt.show()

*# Analyze and plot for both datasets*

find\_and\_plot\_peaks(data1, 'Data\_1')



import numpy as np

import matplotlib.pyplot as plt

from scipy.signal import find\_peaks

*# Load data*

data1 = np.loadtxt('/content/Data\_2.txt')

*# Define a function to find and plot peaks*

def find\_and\_plot\_peaks(data, label):

*# Find maxima (peaks)*

peaks, \_ = find\_peaks(data, distance=20, prominence=0.5)

*# Find minima by inverting the signal and finding peaks*

inversed\_data = -data

minima, \_ = find\_peaks(inversed\_data, distance=20, prominence=0.5)

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

plt.plot(data, label=f'{label} Signal', color='black')

plt.plot(peaks, data[peaks], 'ro', label='Maxima')

plt.plot(minima, data[minima], 'bo', label='Minima')

plt.title(f'{label} Signal Peaks')

plt.xlabel('Index')

plt.ylabel('Amplitude')

plt.legend(loc='center left', bbox\_to\_anchor=(-0.15, 0.5))

plt.grid(True)

plt.show()

