|  |  |  |  |
| --- | --- | --- | --- |
| **SVKM's-IOT, Dhule**Shri Vile Parle Kelavani Mandal's  **INSTITUTE OF TECHNOLOGY**  **DHULE (M.S.)**  **DEPARMENT OF COMPUTER ENGINEERING** | | | |
| **Subject:** Competitive Programming Lab | | | Remark |
| **Name: Jaykishan Natwar Varma** | | **Roll No.:** 68 |
| **Class:** TY. Comp. Engg. | **Batch:** T4 | **Division:** |
| **Expt. No.:** | **Date :** | | Signature |
| **Title:**  Vito’s Family | | |
|  | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **ASSIGNMENT/EXPERIMENT: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | | |
| **Date of Performance:**  **Date of Submission:** | | |
| **Marks Split Up** | **Maximum Marks** | **Marks Obtained** |
| **Performance/Conduction** | **3** |  |
| **Report Writing** | **3** |  |
| **Attendance** | **2** |  |
| **Viva/Oral** | **2** |  |
| **Total Marks** | **10** |  |
| **Signature of Subject Teacher** | | |

**Title:** Vito’s Family

**Aim:** The world-known gangster Vito Deadstone is moving to New York. He has a very big family there, all of them living in Lamafia Avenue. Since he will visit all his relatives very often, he is trying to find a house close to them. Vito wants to minimize the total distance to all of them and has blackmailed you to write a program that solves his problem.

**Language used: Python**

**Platform Used: Pycharm, VS code etc.,**

**Sample Input:** The input consists of several test cases. The first line contains the number of test cases. For each test case you will be given the integer number of relatives r (0 < r < 500) and the street numbers (also integers) s1, s2, . . . , si , . . . , sr where they live (0 < si < 30000 ). Note that several relatives could live in the same street number.

**Sample Output:** For each test case your program must write the minimal sum of distances from the optimal Vito’s house to each one of his relatives. The distance between two street numbers si and sj is dij = |si − sj |.

**Example:**

***Sample Input:***

2

2 2 4

3 2 4 6

***Sample Output:***

2

4

**Algorithm/Flowchart:**

function vitos\_family(locations):

// Sort the locations array

sort(locations)

// Calculate the median (optimal location)

n = length(locations)

if n % 2 == 1:

median\_index = n // 2

optimal\_location = locations[median\_index]

else:

median\_index = n // 2

// For even number of locations, choose either of the two middle points (e.g., the lower one)

optimal\_location = locations[median\_index - 1]

// Calculate the total distance to the optimal location

total\_distance = 0

for location in locations:

total\_distance += abs(location - optimal\_location)

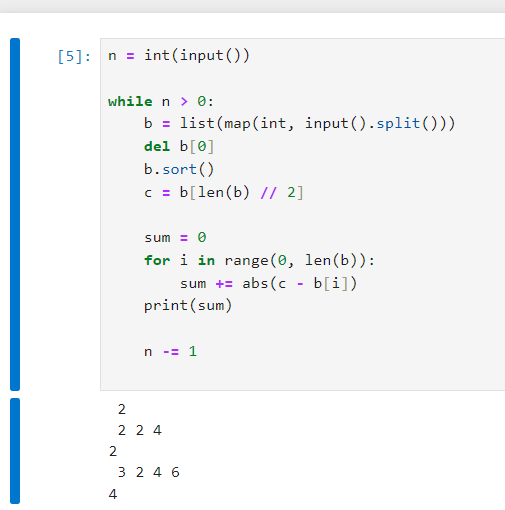
return total\_distance

// Example usage:

locations = [2, 4, 6, 8, 10]

result = vitos\_family(locations)

print("Minimum total distance:", result)

**Code:**

**Conclusion:**

In this way we can implement the vito’s family using dictionary, loops and conditional statements.