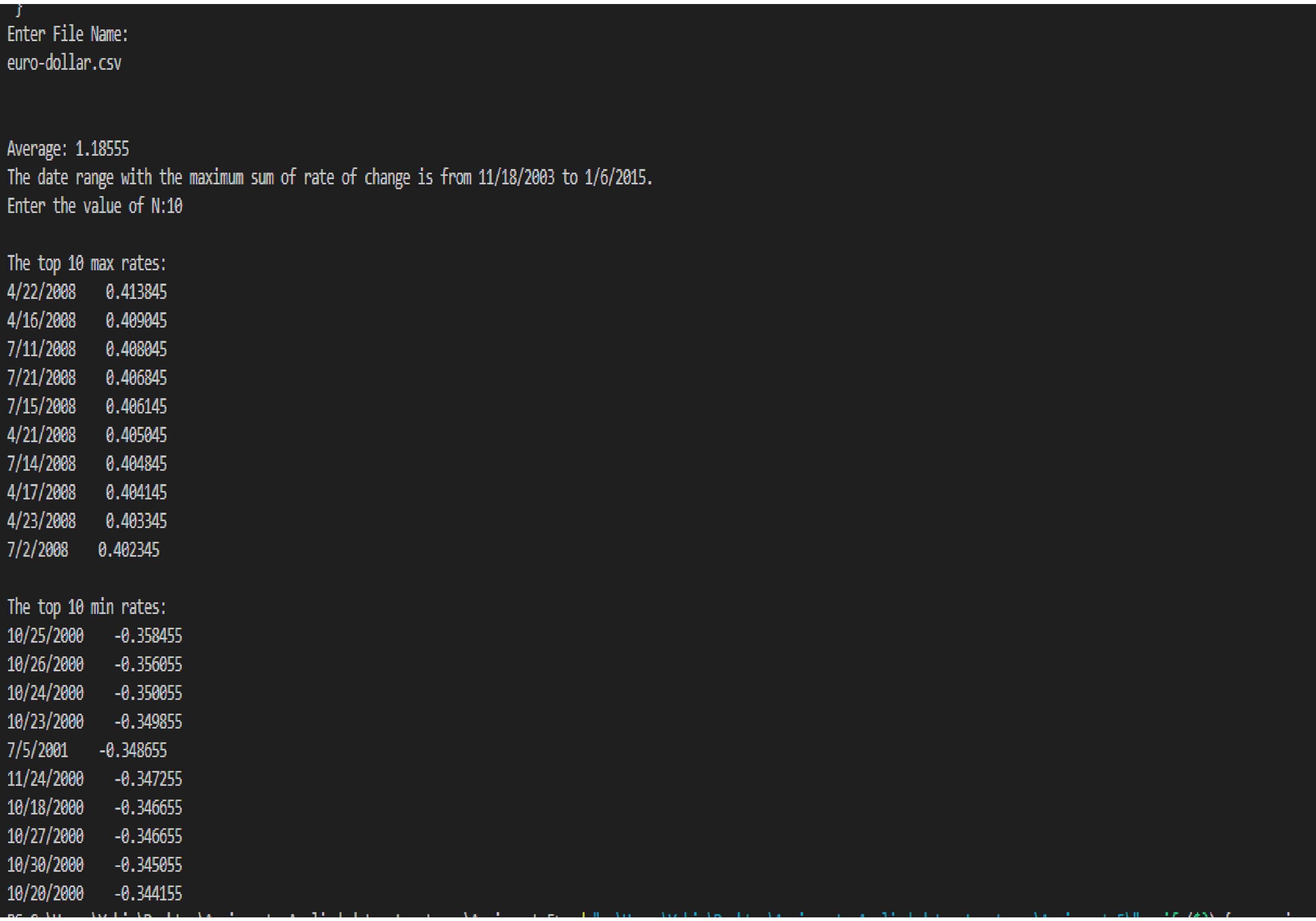
Output



```
}
Enter File Name:
euro-dollar.csv



Average: 1.18555
The date range with the maximum sum of rate of change is from 11/18/2003 to 1/6/2015.
Enter the value of N:10

The top 10 max rates:
4/22/2008    0.413845
4/16/2008    0.409045
7/11/2008    0.408045
7/21/2008    0.406845
7/15/2008    0.406145
4/21/2008    0.405045
7/14/2008    0.404845
4/17/2008    0.404145
4/23/2008    0.403345
7/2/2008    0.402345

The top 10 min rates:
10/25/2000    -0.358455
10/26/2000    -0.356055
10/24/2000    -0.350055
10/23/2000    -0.349855
7/5/2001    -0.348655
11/24/2000    -0.347255
10/18/2000    -0.346655
10/27/2000    -0.346655
10/30/2000    -0.345055
10/20/2000    -0.344155
```

Explanation

The program relies on a structure called *Exchange_Data*, which stores information about each exchange. It includes the date of the exchange, the value of the exchange, and the rate of change of that value compared to the average value.

When the program runs, it first asks the user to provide the name of a CSV file. This file contains the exchange data, with each line representing a different exchange. The program reads this file and creates *Exchange_Data* objects for each exchange, storing them in a vector called data.

After reading the data, the program calculates and displays the average value of all exchanges. It also finds the date range with the highest sum of the rate of changes and displays this information to the user.

Next, the program asks the user to enter a value for N. This N represents the number of maximum or minimum rate of change values the user wants to find.

Then the program rearranges the *data* vector into two different structures: a Maximum Heap and a Minimum Heap. These heaps are used to facilitate finding the N maximum and minimum values, respectively. Then, based on the user's input, the program extracts and displays the top N maximum or minimum rate of change values from the appropriate heap. This rearrangement ensures efficient extraction of the top N values by utilizing the properties of heaps.

Overall, the program reads exchange data from a CSV file, calculates relevant statistics, and allows the user to find the top N maximum or minimum rate of change values.

Analysis

1. Reading Data from File
   - The function read_data reads data from a file and stores it in a vector.
   - It reads each line of the file, so the time complexity is O(n), where n is the number of lines in the file.
2. Calculating Rate of Change
   - The function CalculateChange calculates the rate of change for each data point.
   - It loops through each data point once, so the time complexity is O(n), where n is the number of data points.
3. Building Heap
   - The functions MaxHeapify, Build_Max_Heap, MinHeapify, and Build_Min_Heap are used to build max and min heaps.
   - Both MaxHeapify and Build_Max_Heap have a time complexity of O(log n), where n is the number of elements in the heap. Which is the same as MinHeapify and Build_Min_Heap.

4. Finding Top N Max or Min Rates
   - The function Top_rates finds the top N max or min rates.
   - It repeatedly extracts the maximum (or minimum) element N times and performs heapify operations.
   - The time complexity of extracting an element and heapify operation is O(log n), where n is the number of elements in the heap.
   - Thus, the overall time complexity is O(N log n), where N is an input from the user which is the number of top rates required and n is the number of elements in the heap.
5. Finding Maximum Sum Subarray of Rate of Change
   - The function findMaxRate finds the subarray with the maximum sum of the rate of change.
   - It uses Kadane's algorithm to find the maximum sum subarray.
   - Kadane's algorithm has a time complexity of O(n), where n is the number of elements in the array.