Description of the problem:
1. Here, the Monte Carlo method generally relies on the repeated random sampling to obtain numerical results.
2. In the problem,we should take some random variables to calculate the entire time and also should take the sum of different values to make the total time taken.
3. After summing we get a normal curve to plot a graph. In order to do algorithm, we should take all the random values as input and proceed to do the code.
4. Given the problem,Able(A) is experienced in his work and can do more work as compared to Baker(B) and we need to distribute the customers among these two persons.
5. Consider we can take 1000 simulations and:
Per 1 call we get different no.of time(it can be sec,min,hrs)
6. There can be no.of different possibilities for each person to take the calls. After taking these calls we should calculate the time taken for each person individually and sum all the values.
7. If Able is busy with his calls then the next customer will be given to Baker ,if Baker is busy then the next customer will be assigned to the person who completes his work in a shorter amount of time.
8. The waiting time, time spent for each customer,service time provided by both Able and Baker is noted.
9. The time will be calculated as per random values which are taken.
10.In the last ,if Able is busy with his calls then it goes to Baker ,even if Baker is busy with his calls it goes to Able,if both of them are busy with their calls then it goes to other person “C”,or if “C” is busy then the customer will be in a waiting list.

Approach:
Here,this code is done based on the Monte Carlo method, because this method involves multiple random trials.
As because, we’ve taken some random values according to the input and summing the approximate values it could lead to the approximate answer.
In this algorithm, the resultant values will be used to find a mean and standard deviation of final approximation to the total server time.

Algorithm:
Assume: A as Able
B as Baker
1. Let us initialise call centre timings(100)[for example]
Assume Range:0-1000 seconds
For example:Take [5,40,58,320…….]

2. Let us initialise time gap between the calls

Assume Range: 0-500 seconds
Example: Take [15,56,118,450……]

```
3. {
for(1000 total runs):
generate call centre timings ------ 1
generate time gap between calls ---------2
i=0
address.time = A[i]
time gap to next call = t[i]
if(A is available):
Allot A ------ A[i]
A---------- busy
wait for t[i] seconds
else if(B is available):
Allot B ---------A[i]
B--------- busy
wait for t[i] seconds
else:
wait for A or B to be available
return 0;
}
```

4.Calculate total time for all the calls taking random variable

5.Calculate total time for 1000 different calls

6.Plot it in a graph and find mean ,standard deviation

7.Time taken by A [T(A)],current time [T(now)],Time taken by B [T(B)],waiting time [T(w)],Caller's time[T(C)].

Complexity:
Consider,
The total method is run for n number of iterations
In each iteration there are m number of calls,
Then the complexity of the algorithm is $O(n \times m)$.