Programming Problem

The StampDispenser class represents a postage stamp vending machine. The machine contains stamps of different values. The machine will always have a stamp with a value of 1 cent and the machine will never run out of any type of stamp. The machine should allow a consumer of the class to calculate the minimum number of stamps that the machine can dispense to fill a given request.

*Your task is to complete one of the provided implementations of the StampDispenser class*: *C++, C#,* or Java.

/\*\*

\* Facilitates dispensing stamps for a postage stamp machine.

\*/

public class StampDispenser

{

/\*\*

\* Constructs a new StampDispenser that will be able to dispense the given

\* types of stamps.

\*

\* @param stampDenominations The values of the types of stamps that the

\* machine should have. Should be sorted in descending order and

\* contain at least a 1.

\*/

public StampDispenser(int[] stampDenominations)

{

}

/\*\*

\* Returns the minimum number of stamps that the machine can dispense to

\* fill the given request.

\*

\* @param request The total value of the stamps to be dispensed.

\*/

public int calcMinNumStampsToFillRequest(int request)

{

return 0;

}

}

As an example, suppose an instance of StampDispenser was created with stampDenominations, {90, 30, 24, 10, 6, 2, 1}, and calcMinNumStampsToFillRequest(int) was called with request, 34. The call should return 2, as 34 cents can best be filled by one 24 cent stamp and one 10 cent stamp.

Things to keep in mind:

1. Assume that a junior programmer is going to read your code. You should include comments and any other aides that you use to communicate your code to other developers.
2. Optimize the code for speed.
3. The code should compile and work.
4. The code should work for countries with high denomination values where stamp values of 1000 or 9000 are common.