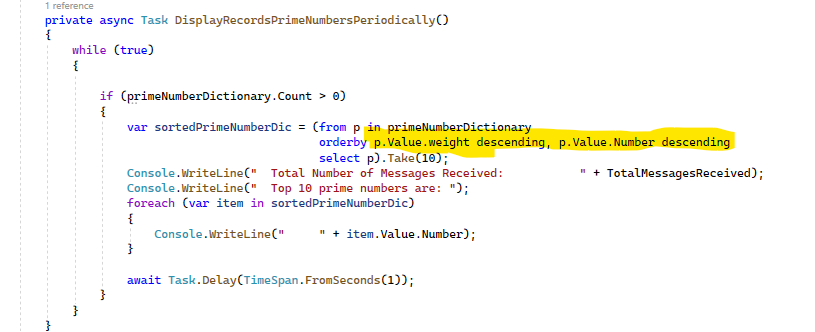
**gRPC Server + client side implementation detailed document for the written code.**

**Server Side:**  
1. New ASP.Net core project created.  
2. Structural analysis of the generated code by the .NET  
3. To implement server for the gRPC we must create a structure of file which the name as proto. I have created a new proto file name as primenumber.proto.   
  
gRPC procedure call are of 4 types  
1. Unary (1-1)  
2. Client streaming  
3. Server streaming  
4. Bi-Directional streaming  
  
As per the requirement in the assignment I decided to use Bi-Directional streaming because we are sending streams of 10000 numbers from client side each second and we want the result of each message from the server side. On the server side I have also implemented the streaming of gRPC. So that we take stream of request from the client side and process each stream and return the response from the server. All the communication from the client and server is async that’s why I have to implement each function using async, await.   
Keeping track of all the valid prime numbers I first created a simple dictionary but after releasing the fact that calls from the client side are concurrent and this will lead to errors on server side I decided to use the ConcurrentDictionary for adding all the valid prime numbers. This dictionary helps me for tracking all the valid prime numbers along with displaying top 10 prime numbers in descending order.   
For displaying top 10 max called valid prime numbers. I have created a weight field for keeping track of max valid called prime number. I used C# query syntax for displaying max weight if the max weight is same then get the max number.  


**Client Side:**1. Referencing the proto file from the server project by using linking in the csproj.  
2. Channel established with the server by providing service hosted ip.  
3. ConcurentDictionary created for tracking all the messages which are processed or not.  
4. For all unprocessed message, I have added a bit to retry sending messages to the server for processing and updating the dictionary accordingly.  
5. Class with name PrimeNumber is created with 3 fields (number, IsPrime, IsMessageProcessed) this class is used in the concurrent dictionary for tracking the records.  
  
I have also explained each step on my github repo.

**Note:**  
I have added bat files for project server + client. Just execute those files and binaries will be generated respectively.   
If you are unable to execute those bat files then follow below steps:  
1. Open cmd and navigate to root directory of the project GrpcPrimeNumberService  
2. Write dotnet run and press enter  
3. This command will starting serving the service  
4. For client code, go to client code root directory ConsoleApp  
5. Write dotnet run and press enter  
6. It will the client and trying to send messages to the server.