DIFFIE HELLMAN

Approach Note

**Steps of Approach**

1. I will open a project in Visual Studio, add header files and other libraries like Open SSL3, LibreSSL 3.
2. Once a successful project is created, I will create a variable *p* which will contain a random prime number.
3. We will find the primitive roots of prime number *p* which will be assigned to variable *q*.
   1. To find the primitive root of *p* we will use a loop which will check each and every value which comes before *p* in the whole number series.
   2. To check; we will calculate the mod of *q1* to *qp-1* with respect to *p*.
   3. If the values received after mod are different, then that value *q* is the primitive root of prime number *p*.
4. If more than one primitive roots are found, I will select the primitive root which is largest out of all the roots and assign that value to a variable *q*.
5. In this step I will generate two random numbers which will be assigned to variables like *a* and *b* which will be the private keys of the clients who are trying to interact.
6. With the help of these private keys a and b we will create public keys *pa* and *pb* with the help of this formula . Similarly, .
7. This public key *pa* and *pb* will be exchanged between the clients meaning **clientB** will use public key *A* and **clientA** will use public key *B*.
   1. Now **clientA** will use public key B in the formula - and **clientB** will use public key A in the formula - .
8. Numbers *xA* and *xB* are compared with each other. If both of them are equal to each other this means that **clientA** and **clientB** are connected securely and they can start communicating with each other.

Step 8 is the end result to this algorithm.