Diffie-Hellman Key Exchange Demonstration

First Alice and Bob publicly agree to use (p, g), where g is a primitive root of p, and p should be some integer power of a odd prime number

```
In[76]:= p = Prime[10]
             素数
Out[76]= 29
```

```
In[77]:= g = PrimitiveRootList[p][[1]]
             原根列表
Out[77]= 2
```

Then Alice choose a random number, say a, she computes $A = g^a \bmod p$, and sends A to Bob

```
In[78]:= SeedRandom[2020]
         随机种子
```

```
In[79]:= a = RandomInteger[{Prime[10], Prime[100]}]
             伪随机整数       素数         素数
Out[79]= 32
```

```
In[80]:= bigA = Mod[g^a, p]
                模余
Out[80]= 16
```

Meanwhile, Bob computes $B = g^b \bmod p$, and sends B to Alice

```
In[81]:= b = RandomInteger[{Prime[10], Prime[100]}]
             伪随机整数       素数         素数
Out[81]= 219
```

```
In[82]:= bigB = Mod[g^b, p]
                模余
Out[82]= 10
```

Once Alice received B, which is the number sends from Bob, she will compute the shared secret s, where $s = B^a \bmod p$

```
In[83]:= sA = Mod[bigB^a, p]
              模余
Out[83]= 24
```

Also, right after Bob received A, which is the number sends from Alice, he will compute the shared secret s, where $s = A^b \bmod p$

In[84]:=
```
sB = Mod[bigA^b, p]
```
模余

Out[84]=
```
24
```

As we can see, the shared secret computed by Alice is sA, which equals to the shared secret computed by Bob, the sB, Alice and Bob now may use s=sA=sB as a shared secret for symmetric encrypted communication.