Step 1: Start

Step 2: Define Power Function:

2.1:Define a function power(base, expo, mod

2.2:Initialize result to 1.

2.3 Use a for loop to multiply result by base, taking the modulus with each multiplication.

```
function power(base, expo, mod):
result = 1
for i = 0 to expo-1:
result = (result * base) % mod
return result
```

Step 3: Input Prime Number and Base:

3.1:Prompt "Enter a prime number" and store it as prime.

3.2:Prompt "Enter a base (primitive root modulo prime)" and store it as gpowermod.

```
print("Enter a prime number: ")
prime = user input
print("Enter a base (primitive root modulo prime): ")
gpowermod = user input
```

Step 4: Input Alice’s and Bob’s Secret Keys:

4.1:Prompt "Enter Alice's secret key" and store it as aseca.

4.2:Prompt "Enter Bob's secret key" and store it as bseca.

```
print("Enter Alice's secret key: ")
aseca = user input
print("Enter Bob's secret key: ")
bseca = user input
```

Step 5: Calculate Public Keys:

5.1:Calculate Alice’s public key A

5.2:Calculate Bob’s public key B

```
A = power(gpowermod, aseca, prime)
```

```
        B = power(gpowermod, bseca, prime)

        print("The public key of Alice is: ", A)

        print("The public key of Bob is: ", B)

Step 6: Calculate Secret Keys:

    6.1:Calculate Alice’s secret key

    6.2:Calculate Bob’s secret key

        calseca = power(B, aseca, prime)

        calsecb = power(A, bseca, prime)

        print("Calculated Secret key of Alice is: ", calseca)

        print("Calculated Secret key of Bob is: ", calsecb)

Step 7: Display Result:

    7.1:Print calseca and calsecb as the shared secret keys for Alice and Bob.

        print("Calculated Secret key of Alice is: ", calseca)

        print("Calculated Secret key of Bob is: ", calsecb)

Step 8: Stop
```