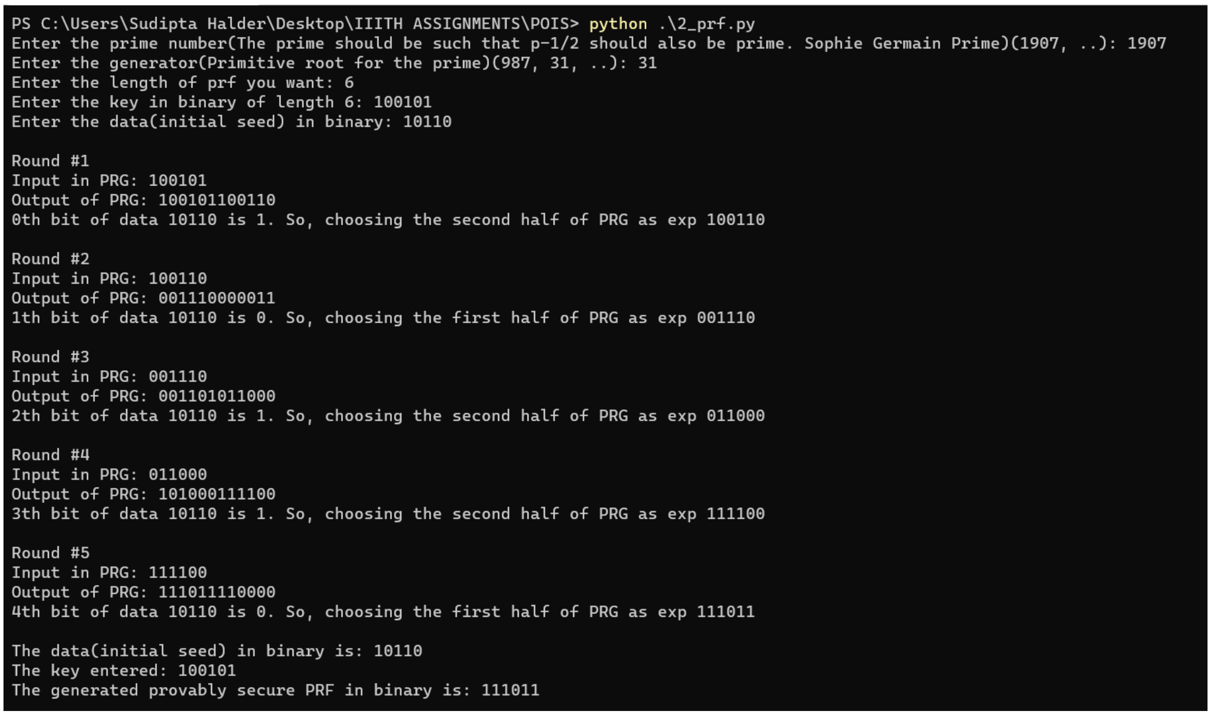
Pseudorandom Function

Input format:

1. It will ask for a safe prime number p in integer format.
2. It will ask for a generator (primitive root) of that prime in integer format.
3. It will ask for the expected output length of the PRF.
4. It will then ask for a key in binary format of the same length.
5. It will then ask to input data.

Output Format:

1. It will return the PRF in binary.



```
PS C:\Users\Sudipta Halder\Desktop\IIITH ASSIGNMENTS\POIS> python .\2_prf.py
Enter the prime number(The prime should be such that p-1/2 should also be prime. Sophie Germain Prime)(1907, ..): 1907
Enter the generator(Primitive root for the prime)(987, 31, ..): 31
Enter the length of prf you want: 6
Enter the key in binary of length 6: 100101
Enter the data(initial seed) in binary: 10110

Round #1
Input in PRG: 100101
Output of PRG: 100101100110
0th bit of data 10110 is 1. So, choosing the second half of PRG as exp 100110

Round #2
Input in PRG: 100110
Output of PRG: 001110000011
1th bit of data 10110 is 0. So, choosing the first half of PRG as exp 001110

Round #3
Input in PRG: 001110
Output of PRG: 001101011000
2th bit of data 10110 is 1. So, choosing the second half of PRG as exp 011000

Round #4
Input in PRG: 011000
Output of PRG: 101000111100
3th bit of data 10110 is 1. So, choosing the second half of PRG as exp 111100

Round #5
Input in PRG: 111100
Output of PRG: 111011110000
4th bit of data 10110 is 0. So, choosing the first half of PRG as exp 111011

The data(initial seed) in binary is: 10110
The key entered: 100101
The generated provably secure PRF in binary is: 111011
```

Working Flow:

1. Upon getting the input, the function ` generate_prf(prime, generator, initial_seed, str(key))` will be called.
2. Inside that function, a loop will run n times, where n = length of the input data in binary.
3. For each iteration, the key will go inside the length doubling PRG, and a 2x length pseudorandom number comes out from PRG where x = length of the key. Now, it will check the bit of the data. If it's 0, it'll choose first x bits (left half), or else, it'll choose the next x bits (right half). And it will be sent as input for the next iteration in PRG. For better understanding, refer to the above pic.
4. Finally, the PRF is given as output.