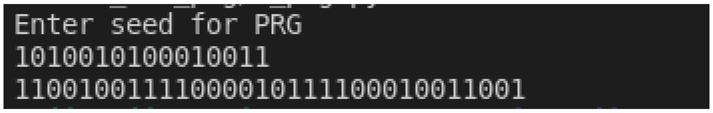
Pseudo random generator code explanation:

- It has three global variables SEED_SIZE, GENERATOR, MODULUS.
- SEED_SIZE is 16, GENERATOR is 2462 which is the primitive root for MODULUS and MODULUS is 18443 which is a prime number.
- Expansion factor of PRG is defined by L(x)=2*x, here it is 32 bits means random number length is 32 bits.
- Initially the user needs to enter a seed and it will be passed to PRG () function and SEED will be divided into two equal parts x,r and we pass this x,r to OWF () function.
- In this OWF () function using x, we calculate DLP value and using r we calculate hardcore predicate and returns value of DLP, r and hard-core predicate bit.
- The DLP,r value is used as SEED for next iteration and in each iteration, we append hard-core predicate bit to our pseudo random number get one bit which is hard-core predicate.
- At the end we combine all hard-core predicate bits and that is generated pseudo random number.

```
def OWF(x,r):
   mod_exp = bin(pow(GENERATOR, int(x,2), MODULUS)).replace('0b',
'').zfill(SEED_SIZE)
   hc=0
   for i in range(len(x)):
       hc=(hc^(int(x[i])&int(r[i])))%2
   return mod_exp+r+str(hc)
```

Output:



```
Enter seed for PRG
1010010100010011
11001001111000010111100010011001
```