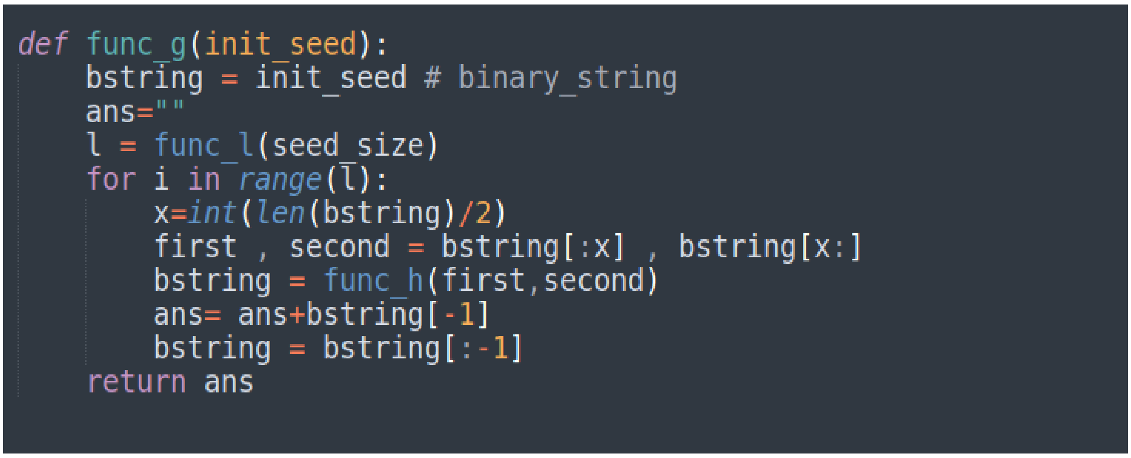
POIS ASSIGNMENT 1

TASK 1

BUILD A PROVABLY SECURE PRG (PSEUDO-RANDOM NUMBER GENERATOR)

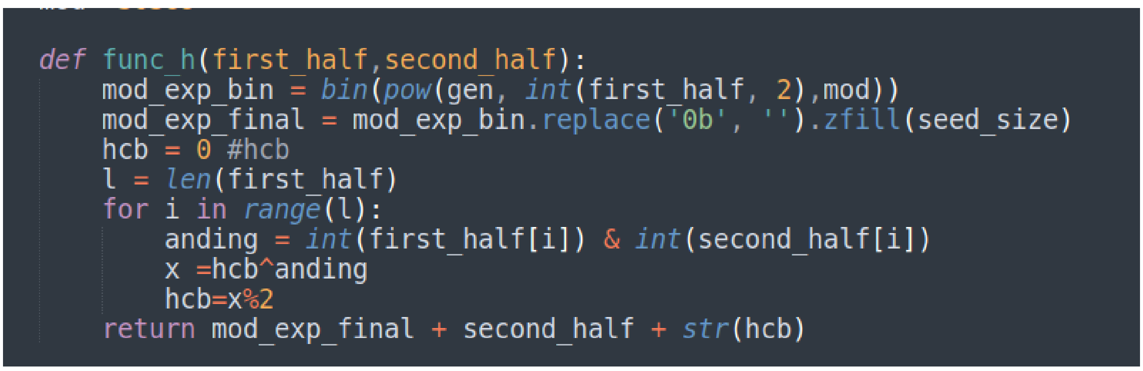
CODE EXPLANATION

G : Function (the PRG itself) that, given a k bit-string in input, outputs a l(k) bit-string and no randomized algorithm can say if the string produced is generated by a real random source or not.



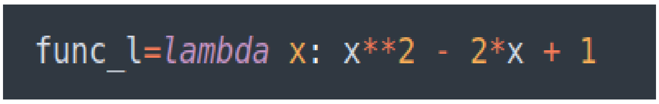
```
def func_g(init_seed):
    bstring = init_seed # binary_string
    ans=""
    l = func_l(seed_size)
    for i in range(l):
        x=int(len(bstring)/2)
        first , second = bstring[:x] , bstring[x:]
        bstring = func_h(first,second)
        ans= ans+bstring[-1]
        bstring = bstring[:-1]
    return ans
```

H : Function that helps to find those pseudorandom bit. Since the input of H is a bit string that will be split into two halves, the length of the initial seed must be even.



```
def func_h(first_half,second_half):
    mod_exp_bin = bin(pow(gen, int(first_half, 2),mod))
    mod_exp_final = mod_exp_bin.replace('0b', '').zfill(seed_size)
    hcb = 0 #hcb
    l = len(first_half)
    for i in range(l):
        anding = int(first_half[i]) & int(second_half[i])
        x =hcb^anding
        hcb=x%2
    return mod_exp_final + second_half + str(hcb)
```

L : Function can be any polynomial function. Used the modular exponentiation as one-way permutation : l(k) = k^2-2k+1.

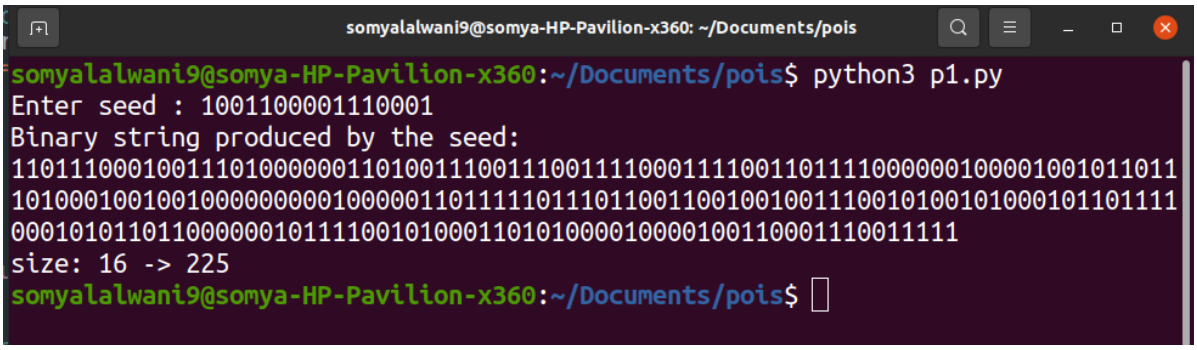


```
func_l=lambda x: x**2 - 2*x + 1
```

Assumptions : Assumed the generator g = 2.

The only argument taken is the initial seed that must be a binary string and longer no more than the value of the variable SEED_SIZE.

OUTPUT :



```
somyalalwani9@somya-HP-Pavilion-x360:~/Documents/pois$ python3 p1.py
Enter seed : 1001100001110001
Binary string produced by the seed:
110111000100111010000000110100111001110011110001111001101111000000100001001011011
10100010010010000000000010000011011111101110110011001001001110010100101000101101111
00010101101100000001011110010100011010100000100000100110001110011111
size: 16 -> 225
somyalalwani9@somya-HP-Pavilion-x360:~/Documents/pois$ 
```