PRF CODE DOCUMENTATION

A class **PRF** is created to represent the pseudorandom function and the class **PRG** is used to represent the pseudo-random function.

```
class PRF:
    def __init__(self,key,length,func_input):
        self.key = key
        self.length = length
        self.func_input = func_input
        # print("input is",self.func_input)
        self.output = ''
```

In init all the parameters passed to the class PRF are initialized,the parameters are as follows:

i) Key variable stores the key of the PRF.

ii) Length variable stores the length of the function (This is a length conserving prf)

iii) func_input stores the input that is given to the function.

iv) Output final output is stored in this variable.

```
def find_function(self):
    curr_val = self.key
    for i in self.func_input:
        prg = PRG(36389,4,2*self.length,curr_val)
        prg.generate_random_bit()
        output = prg.generated_random_string
        if i=='1':
            curr_val = int(output[:self.length],2)
        else:
            curr_val = int(output[self.length:],2)
    self.output = curr_val
```

find_function
This function consists the main algorithm
we iterate over each bit of the given input and we use pseudo random generator class constructed in the first part with an expansion factor of 2. Initially the value of key is passed to the function and its output is stoed in the output variable, now we iterate over the binary tree structure to reach the leaf node
Since prg has an expansion factor of 2 each of the string produced will have twice the length.
If the current bit is 1 the algorithm is thought to be going on the right leaf of the binary tree and the value with the later half n bit is passed to the pseudorandom function for the next iteration.

Similarly, if the current bit is 0 the algorithm is thought to be going on the left leaf of the binary tree and the value with the first half n bit is passed to the pseudorandom function for the next iteration.

This value is stored in the variable **curr_val** and it is initialized to key value. The output of the function is stored in **output** variable, the value of **curr_val** after the end of for loop is the final output.

```
def generate_key(n):
    key_temp = ""
    for i in range(n):
        key_temp+=str(random.randint(0,1))
    return key_temp
```

The function **generate_key(n)** generates a uniform binary string of length n This function is used to get some random key.

```
if __name__=="__main__":
    print(generate_key(17))
    prf = PRF(int(generate_key(17),2),17,generate_key(17))
    # prf = PRF(int("11110100110100100",2),17,"11110000110111100")
    prf.find_function()
    print(prf.output)
    print(type(prf.output))
```

This is the main section of the code where prf class is initialsed and relevant values are passed,then **find_function()** is called and final output is created.