```
#!/usr/bin/env python2
# -*- coding: utf-8 -*-
"""
Created on Fri Feb 17 16:41:07 2017

@author: Armin Galetzka, Denys Bast

Creates and returns the MLBS signal for a specified number of bits.
Standard: bits=9
Input:  bits          ------  Number of bits for register length
Output: output        ------  MLBS Signal
        seedRandom    ------  Used seed for random number generator
"""

import numpy as np
from scipy import stats

def get(bits=10):
    # ------ create random start register ------
    seedRandom = np.random.randint(2**31)
    np.random.seed(seed=579946590)
    xk = np.arange(2)
    pk = (0.5,0.5)
    custm = stats.rv_discrete(name='custm', values=(xk, pk))
    # ------ create Signal ------
    register = np.zeros(bits)
    while (np.sum(register)==0):
        register = np.array(custm.rvs(size=bits))
    N = 2**bits-1
    output = np.zeros(N)

    for i in range(0,N):
        output[i] = register[bits-1]
        if bits==6 or bits==7:
            r = np.logical_xor(register[0],register[bits-1])
        if bits==8:
            r = np.logical_xor(np.logical_xor(np.logical_xor(register[0],\
                               register[1]),register[6]),register[7])
        if bits==9:
            r = np.logical_xor(register[3],register[8])
        if bits==10:
            r = np.logical_xor(register[2],register[9])
        if bits==11:
            r = np.logical_xor(register[1],register[10])
        if r:
            r=1
        else:
            r=0
        register = np.append(r, register[0:bits-1])

    output = output - 0.5
    return (output, seedRandom)
```