

Compito 3.2 di Polizzi Lucrezia fatto in gruppo con Fazio Aurora. Implementazione del problema dell'accordo bizantino che segue il protocollo Las Vegas. Questa volta i processi affidabili sono 8 e l'impostore rimane 1, l'implementazione è molto simile a quelle che segue il protocollo Montecarlo. Veniva inoltre richiesto di calcolare il valor medio che è circa 2.

```
from random import randint

def ProbMoneta():
    valore = randint(0,1)
    return valore

def bitB (b):
    for i in range (0,8):
        valore = randint(0,1)
        b.append(valore)
    return b

def infingardo (inaffidabile,b): # do il valore inaffidabile al
traditore
    for i in range(8):
        inaffidabile = 1 - b[i]
        b.append (inaffidabile)
    return b

def ValMed (R, r):
    aux = 0
    for i in range (R):
        aux += r
    return aux/R
#L = 5t+1 = 6; H = 6t+1= 7 soglie che garantiscono la validità

def LVByzGen (all, inaffidabile, b, L, H): # metto in atto il problema
dei Generali Bizantini
    moneta = ProbMoneta() #calcolo la probabilità della moneta
    check = True
    r = 0
    b = bitB(b)
    while (check == True):#fino a quando non sono d'accordo li faccio
"parlare"
        if (b[0]==b[1] and b[1] == b[2] and b[2]==b[3] and b[3]==b[4]
and b[4]==b[5] and b[5]==b[6] and b[6]==b[7]) :
            check = False
            r += 1
            all = infingardo (inaffidabile,b)
            print (all)
            uno = all.count (1) #calcolo maggioranza
```

```

zero = all.count (0)
if uno > zero: #in base a chi vince do il valore a tally e maj
    maj = 1
    tally = uno
    print ("maj=", maj, "tally=", tally)
else:
    maj = 0
    tally = zero
    print ("maj=", maj, "tally=", tally)

if moneta == 1:
    soglia = L
else:
    soglia = H

if tally >= soglia :
    valore = maj
    b.clear()
    for i in range (0,8):
        b.append(valore)
else:
    valore = 0
    b.clear()
    for i in range (0,8):
        b.append(valore)

if tally >= 8:
    valore = maj
    b.clear()
    for i in range (0,8):
        b.append(valore)

return r,check

if __name__ == "__main__":
    R = pow(2,10)
    b = [] # salvo 8 processi onesti
    all = [] # salvo tutti i processi onesti + 1 traditore
    L = 6
    H = 7
    for i in range(R):
        print("\nRun n°:", i, "\n")
        inaffidabile = 0

        r,check = LVByzGen(all, inaffidabile, b,L,H)
        print ("I round necessari per raggiungere l'accordo sono:", r)
        b.clear()

    vm = ValMed(R,r)

```

```
print("\nIl valor medio è: ", vm)
```