

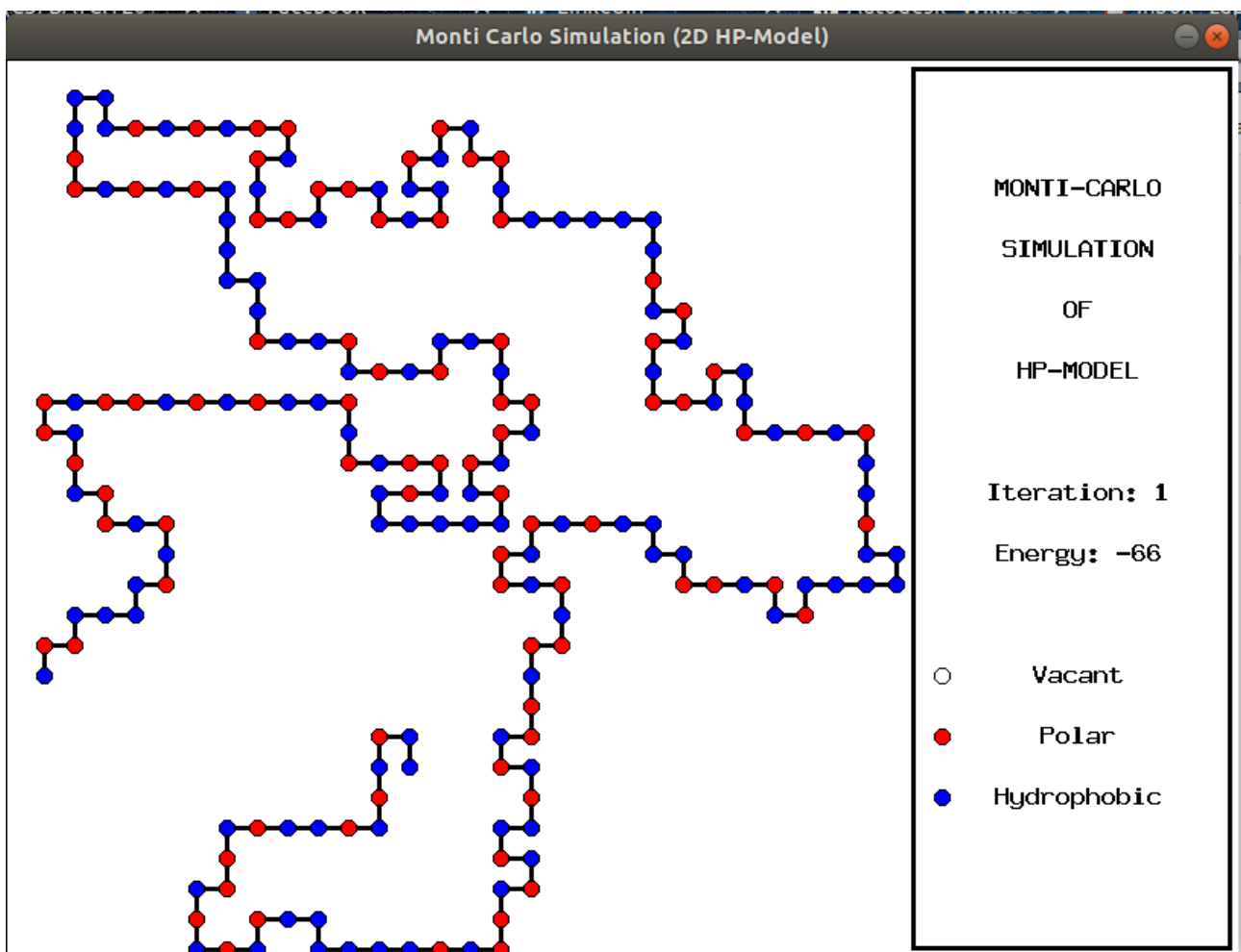
3D Modeling & Simulations in Bioinformatics (CS06F)

ASSIGNMENT: 01

METROPOLIS MONTI CARLO SIMULATION

OF

2D Hydrophobic-Polar model of Proteins



Name: ZEESHAN ALI, Roll No: I14-1623

Instructor: PROF. HAMMAD NAVEED

Main Block

```

probability = 0.5
NUM_OF_ITERATIONS = 1000
hyper_parameter = 6

energy = calculate_Energy(residue_position)

energies = []
rpositions = []
mutation_type = []

for iteration in range(NUM_OF_ITERATIONS):
    rp_, m_type = random_mutation(residue_position)
    residue_position_updated = np.copy(rp_)

    if (residue_position_updated != []):
        energy_updated = calculate_Energy(residue_position_updated)
        if (energy_updated < energy):
            energy = energy_updated
            residue_position = np.copy(residue_position_updated)
        elif ((energy_updated-energy) < hyper_parameter*probability):
            energy = energy_updated
            residue_position = np.copy(residue_position_updated)

    energies.append(energy)
    rpositions.append(residue_position)
    mutation_type.append(m_type)

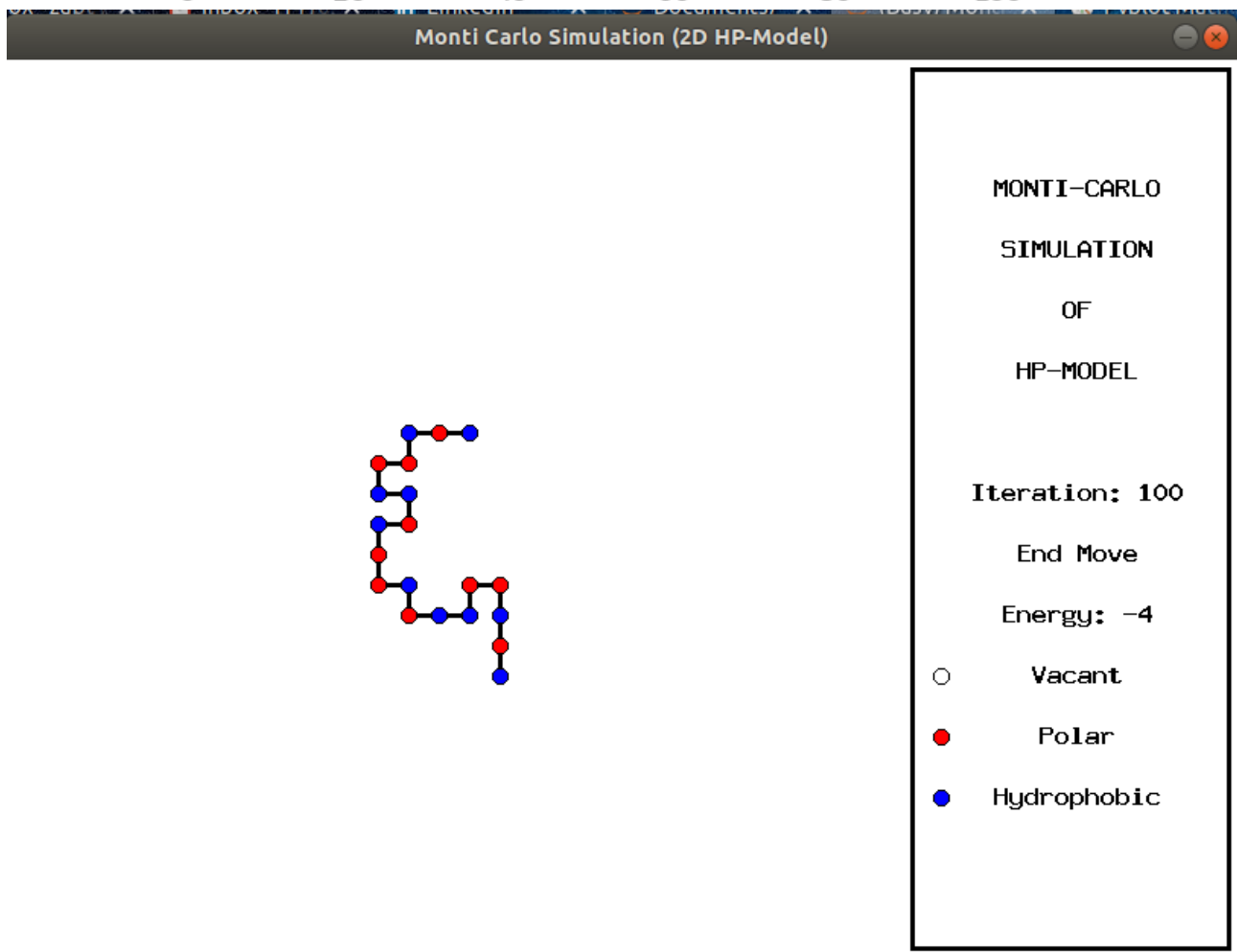
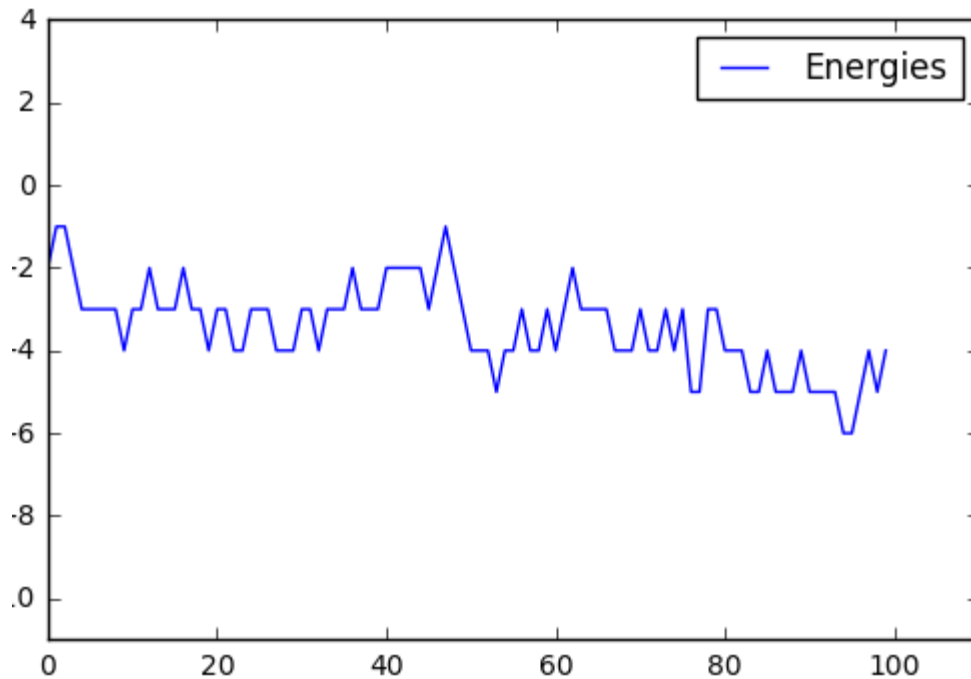
```

Results of the algorithm on different protein sequences

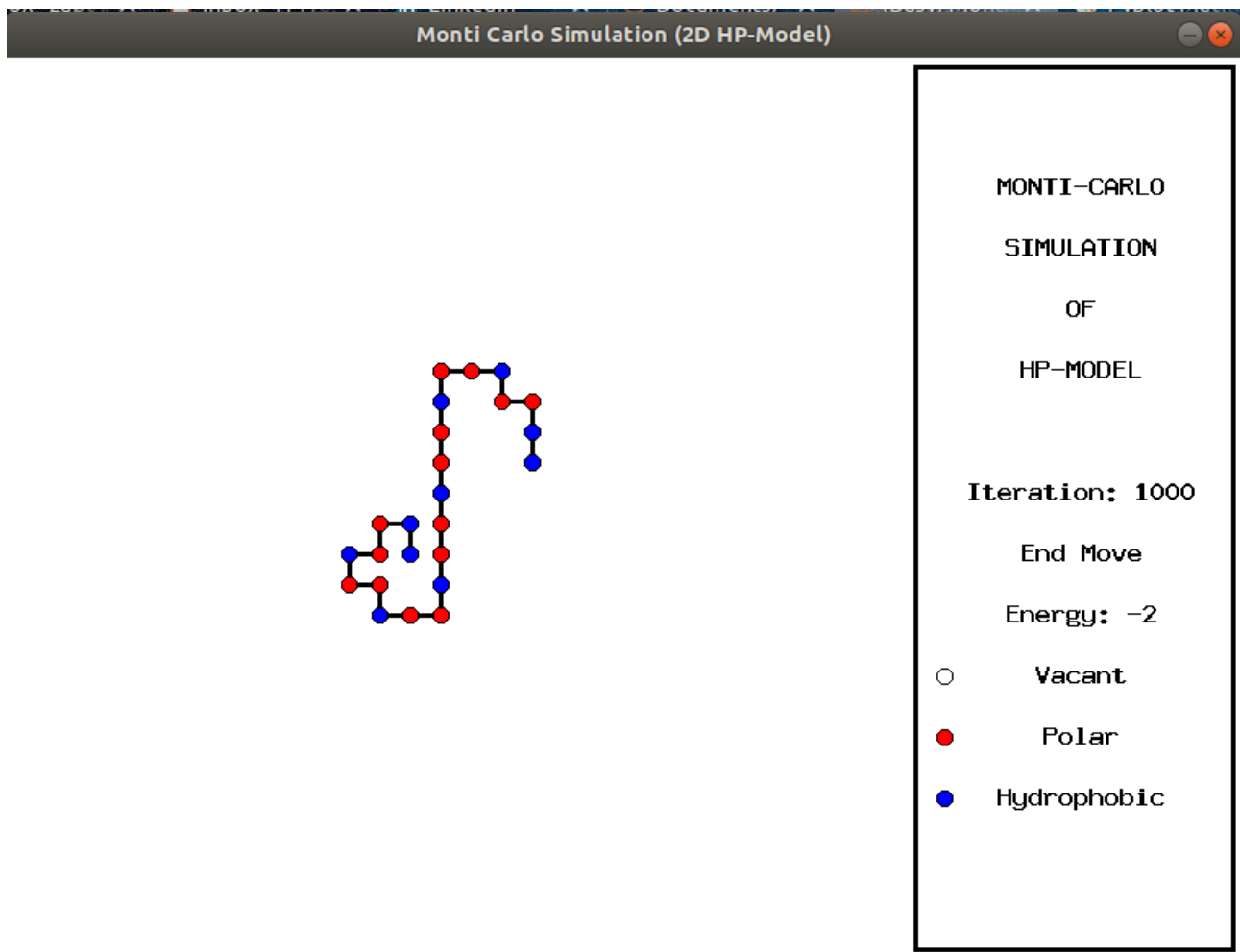
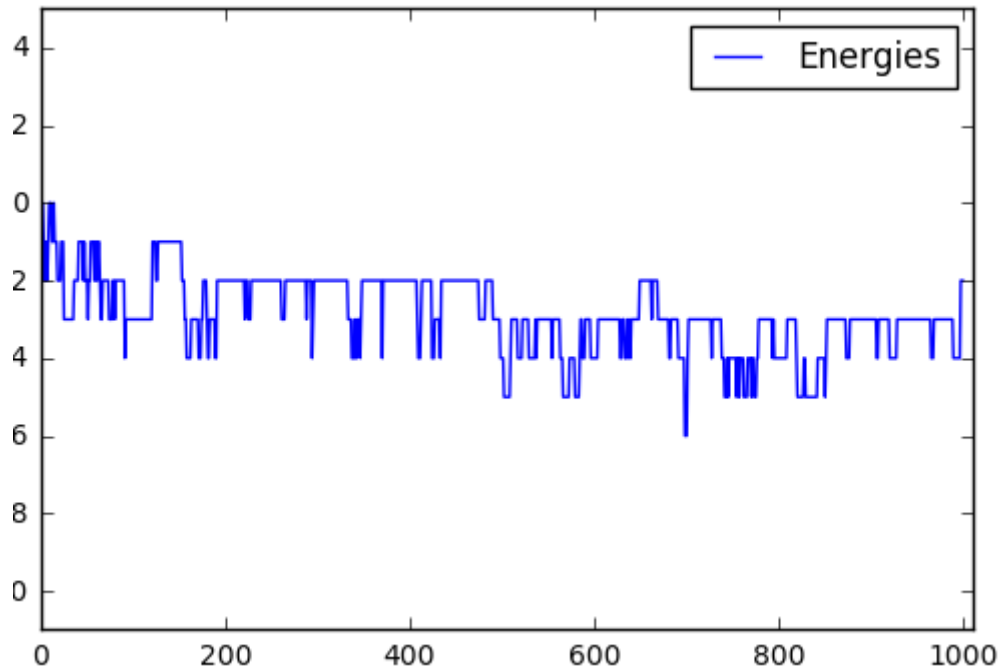
(Observed that the performance was better for longer sequences)

S#	Length	Sequence	Energy
01	20	HRHRPRHHPRHRPRHHPRHRH	-4
02	24	PRHRPRHHPPRRHHPRPPRHPPRRHH	-2
03	25	PRHRPRHHPPRRHHPRPPRHPPRRHH	-2
04	36	PPRRHHRRHHPPRRPRHHHHHHHHPRHHPPRRHHRRHRPP	-5
05	48	PRHRPRHHPRHHPPRRPRHHHHHHHHHHHHPPRRPPRRHHPRHHHHPRHHHHHH	-16
06	51	HHHRHRHRHRHHHHHRHRPRHRPPRRHRPPRRHRPRHRPRHHHHHHHRHRHRHH	-16
07	60	PRHHHHRRHHHHHHHHHRPPRRHHHHHHHHHHHHHRPRHHHHHHHHHHHHHHPPRRHHHHHHHHHRHRHRP	-31
08	64	HHHHHHHHHHHHHHHRHRHRPRHHRRHHRRHRPRHHRRHHRRHRPRHHRRHHRRHRHRHHHHHHHHHHHHHH	-40
09	85	HHHHHPPRRHHHHHHHHHHHHHHHHPPRRPRHHHHHHHHHHHHHHHHPPRRHHHHHHHHHHHHHHPPRRHHHHHHHHHHHHRRHRH HRRHHRRHRH	-59
10	100	PPRRHHRRHHHHRRHHHHRRHRHRHHHHHHPPRRPPRRHHHHHHHHPPRRPPRRHRHHRRHHHHHHHHHHHHHR RHHHRHHRRHRHRHHHHPPRRPRHHH	-33
11	100	PPRRPPRRHRHHRRPPRRHHHHRRHHHHHHRRHRPPRRHHRRHRHHHHHHRRHHHHHHHHHHHHRRHHRRHHHHHHHHPPRRPPRRPP RHHHHHHHHRRHRHHHHPPRRPPRRHRHH	-51

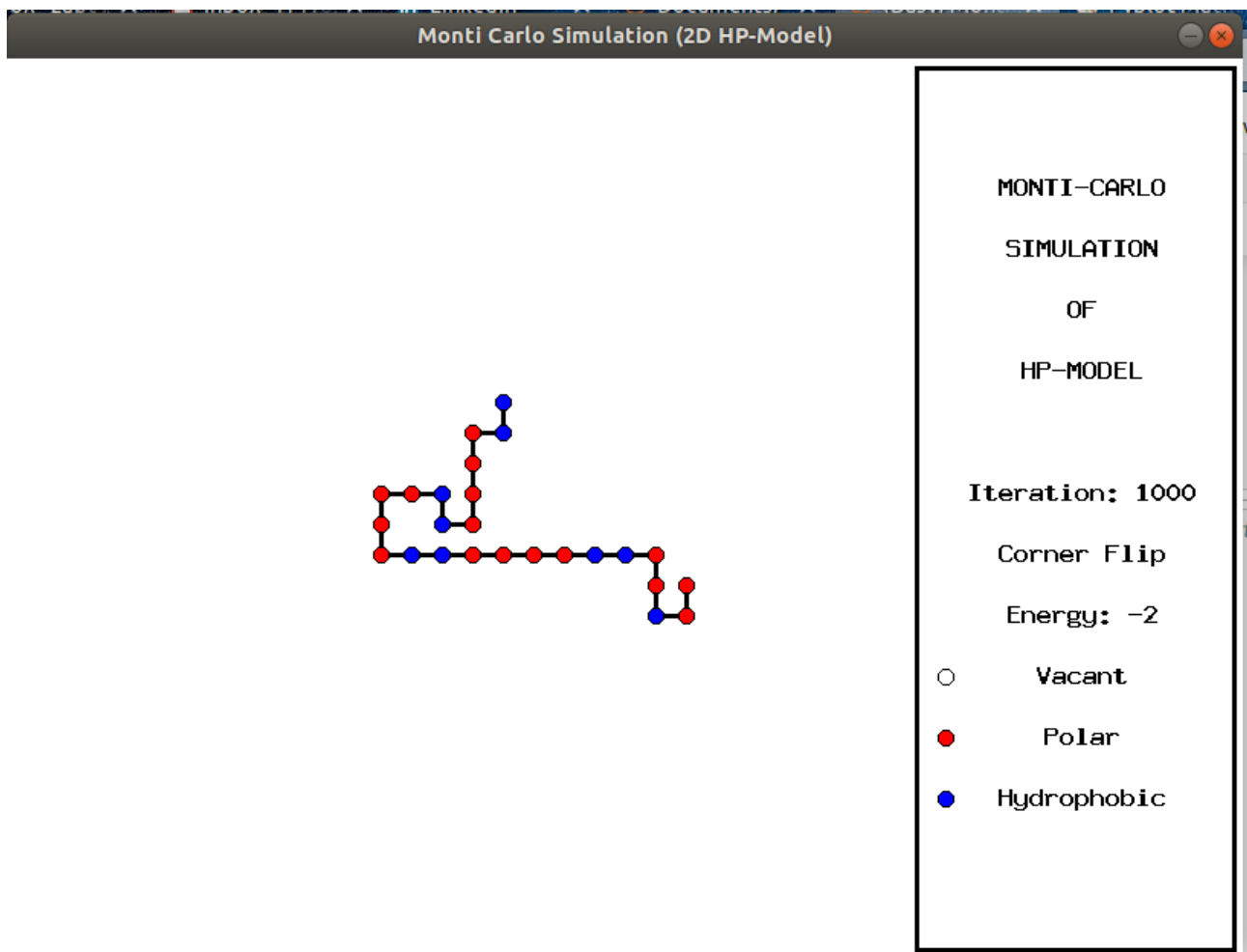
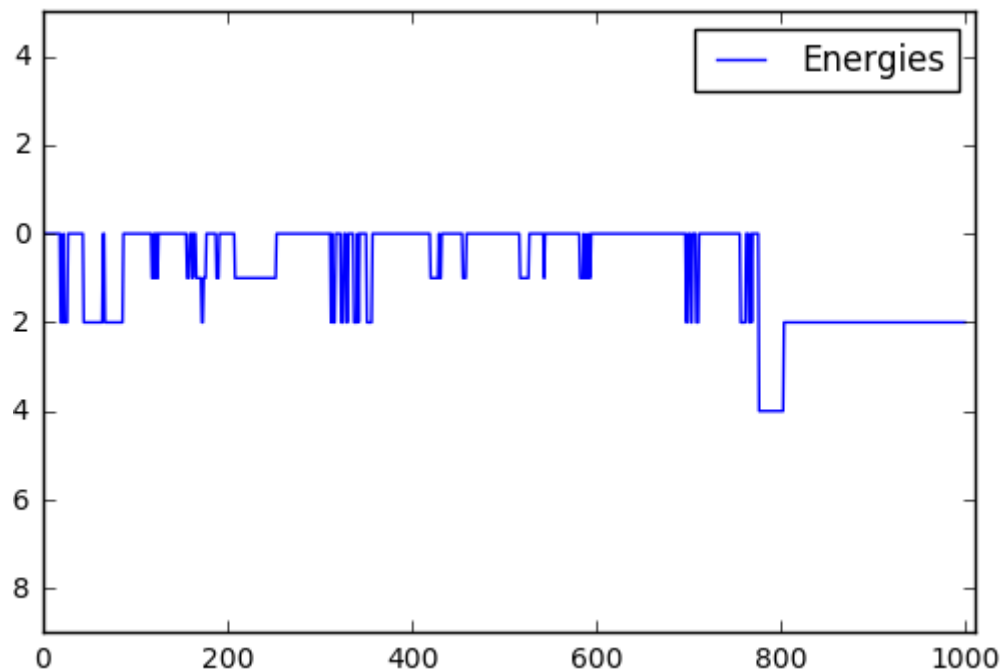
➔ STRING_01 = "HP"*2+"P"+"H"*2+"P"+"H"+"P"*2+"H"+"P"+"H"*2+"P"*2+"HPH"



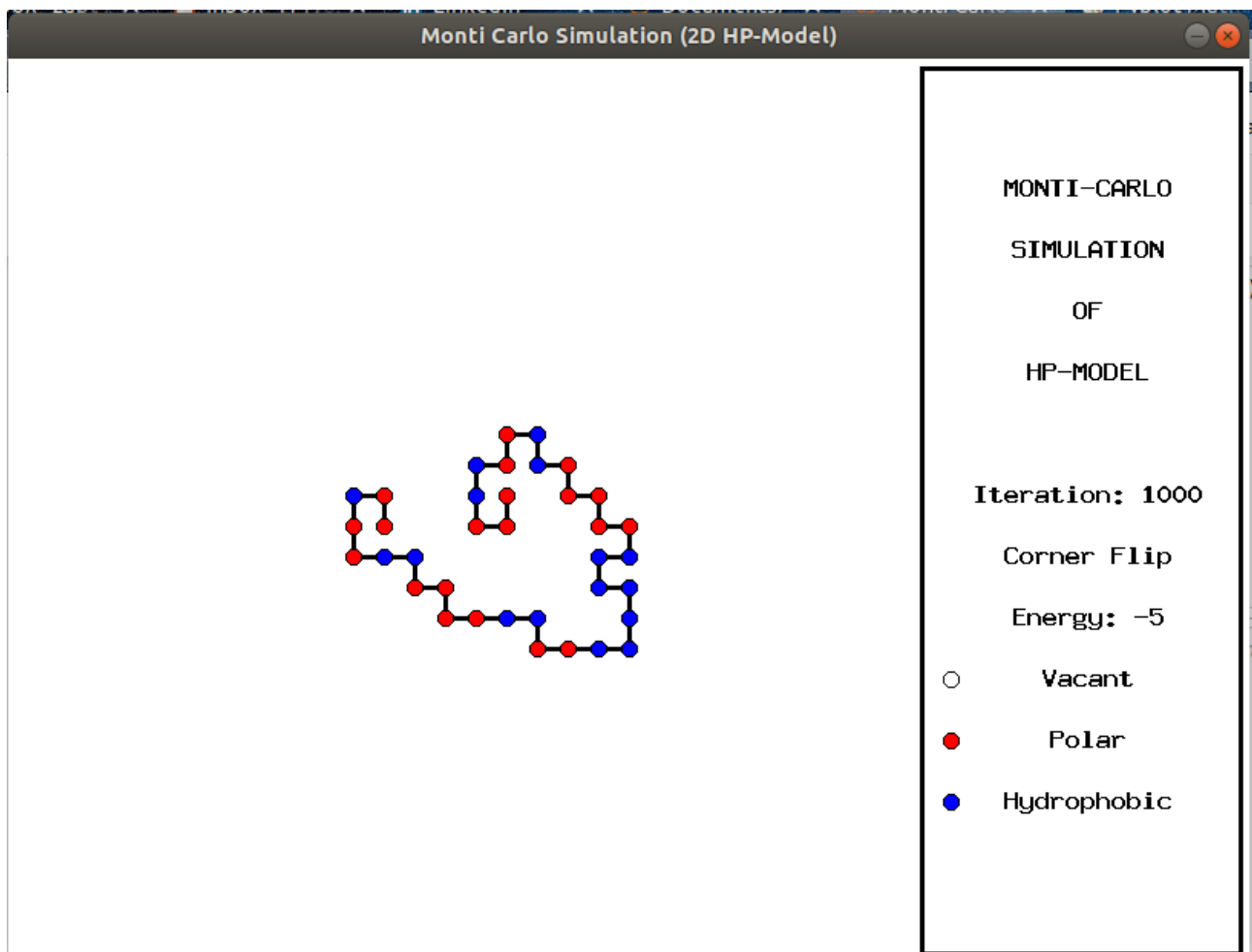
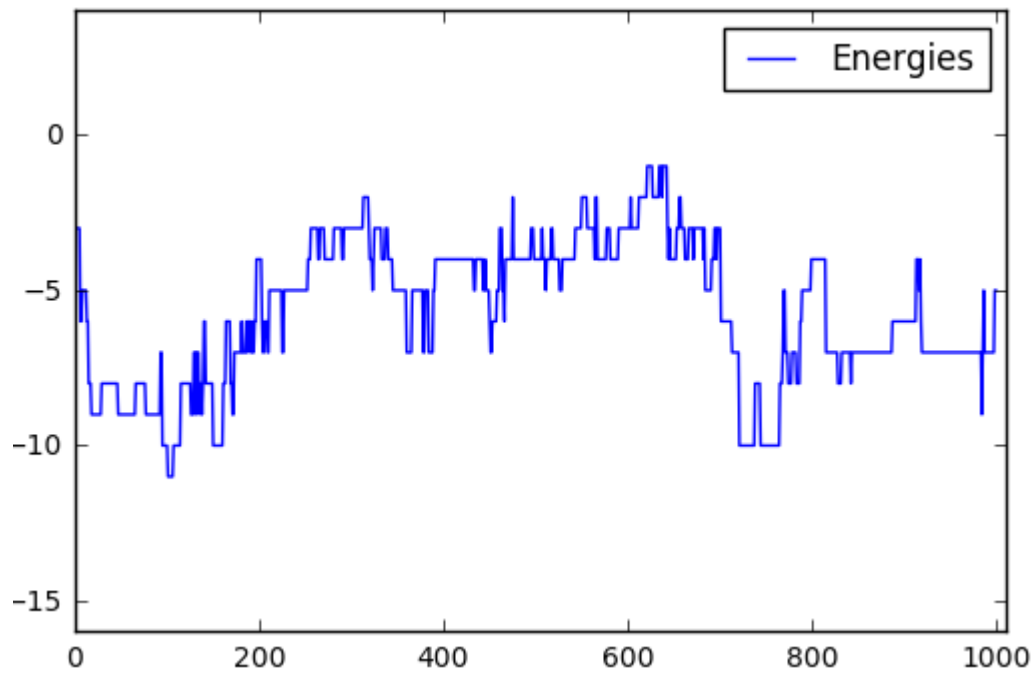
→ `STRING_02 = "H"*2+("P"*2+"H")*7+"H"`



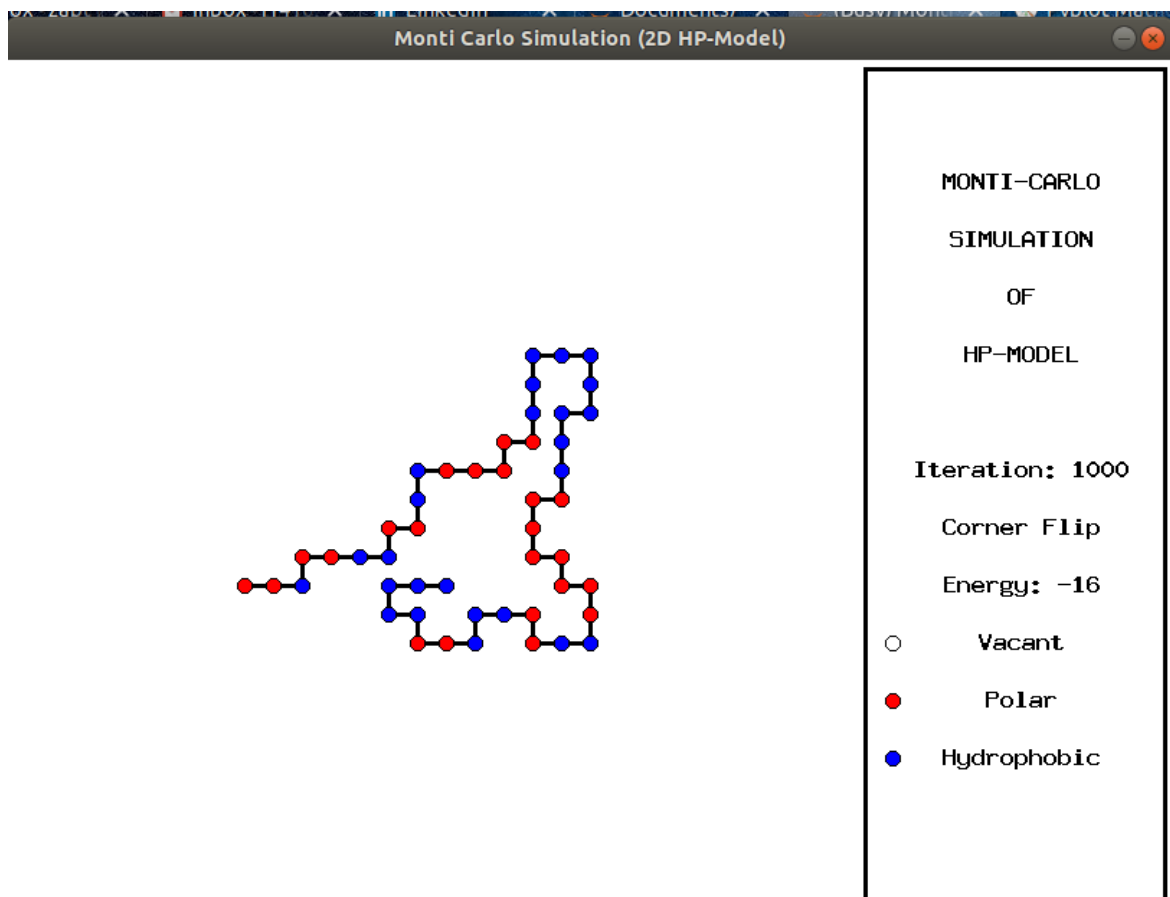
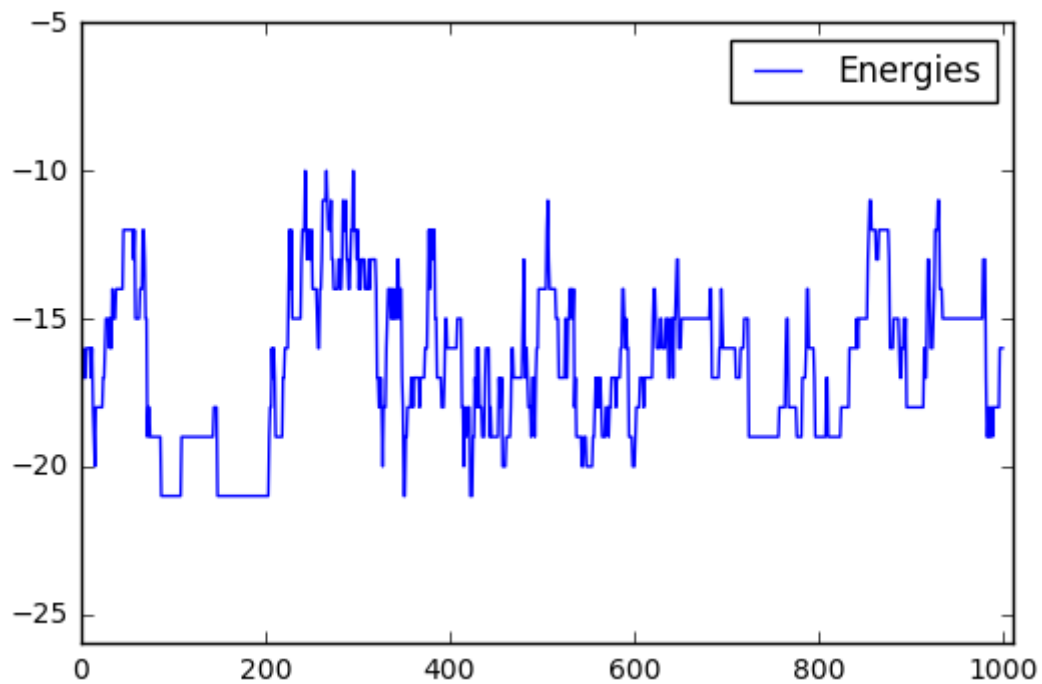
→ `STRING_03 = "P"*2+"H"+"P"*2+("H"*2+"P"*4)*3+"H"*2`



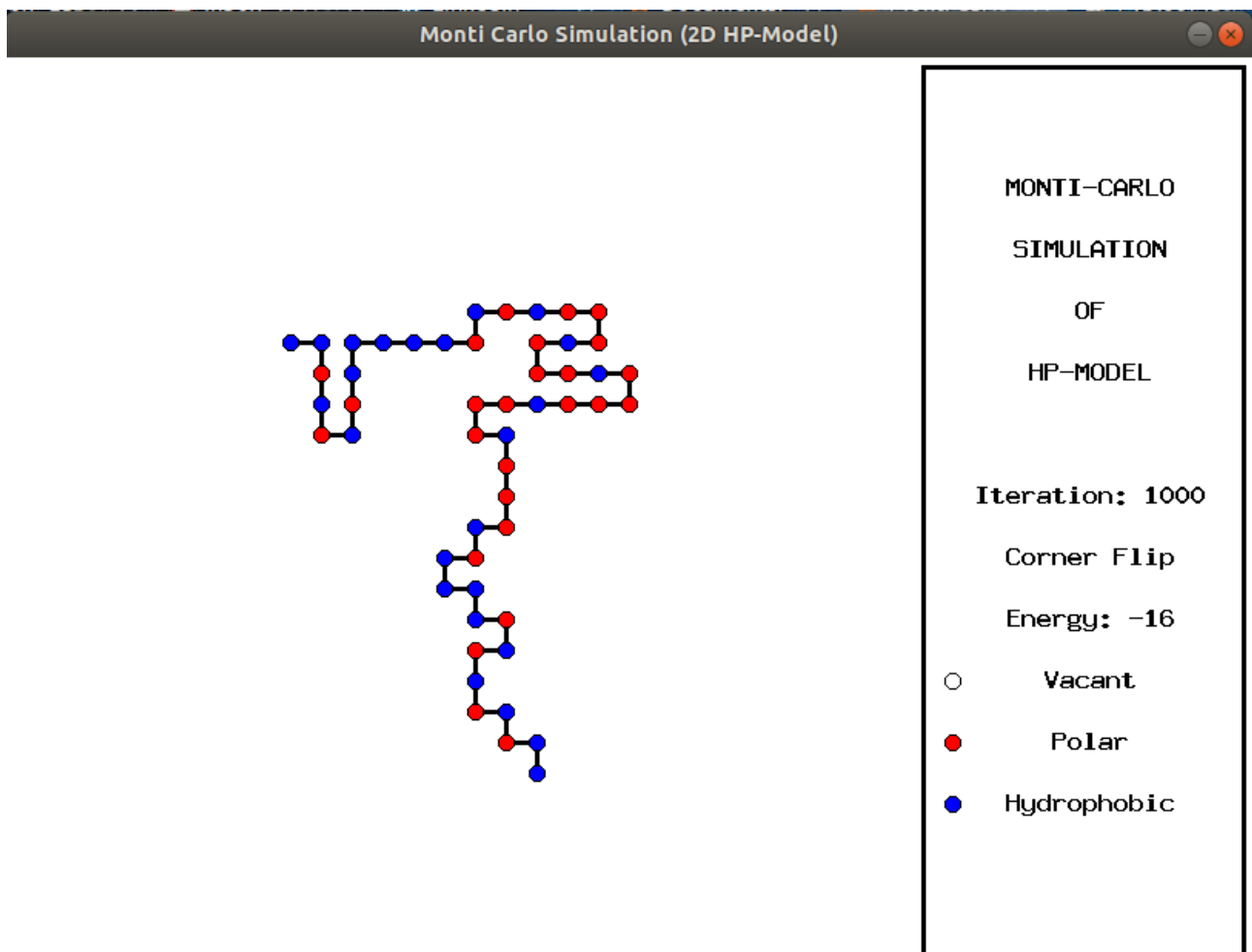
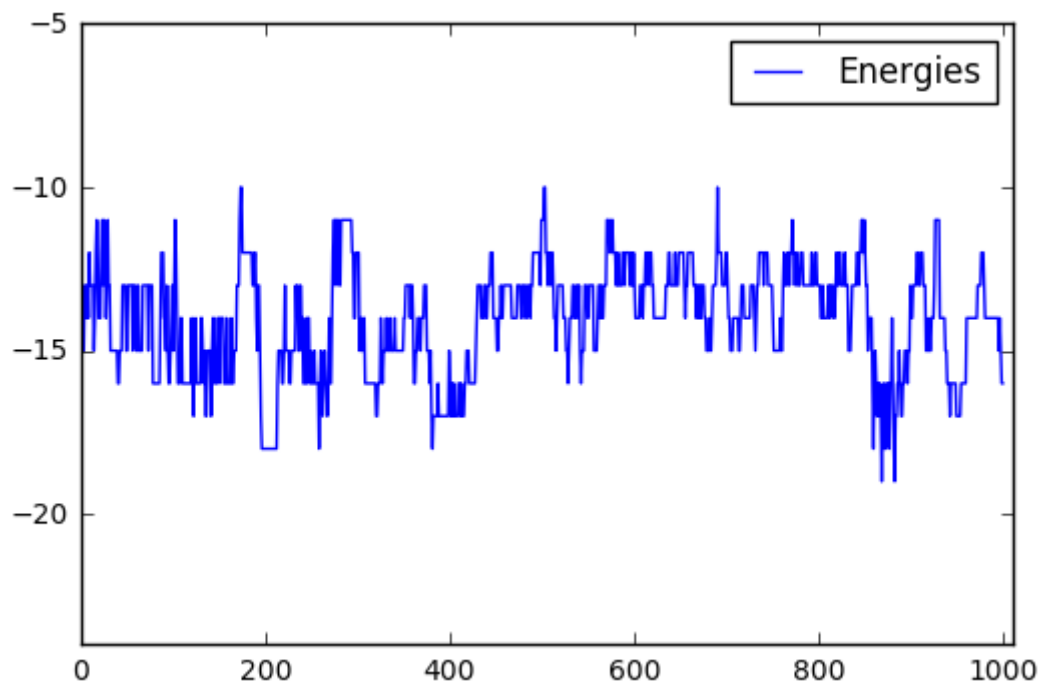
➔ STRING_04 = "PPPHH"+"P"*2+"H"*2+"P"*5+"H"*7+"P"*2+"H"*2+"P"*4+"HHPPHPP"



→ `STRING_05 = "P"*2+"H"+("P"*2+"H"*2)*2+"P"*5+"H"*10+"P"*6+
 ("P"*2+"H"*2)*2+"H"+"P"*2+"H"*5`

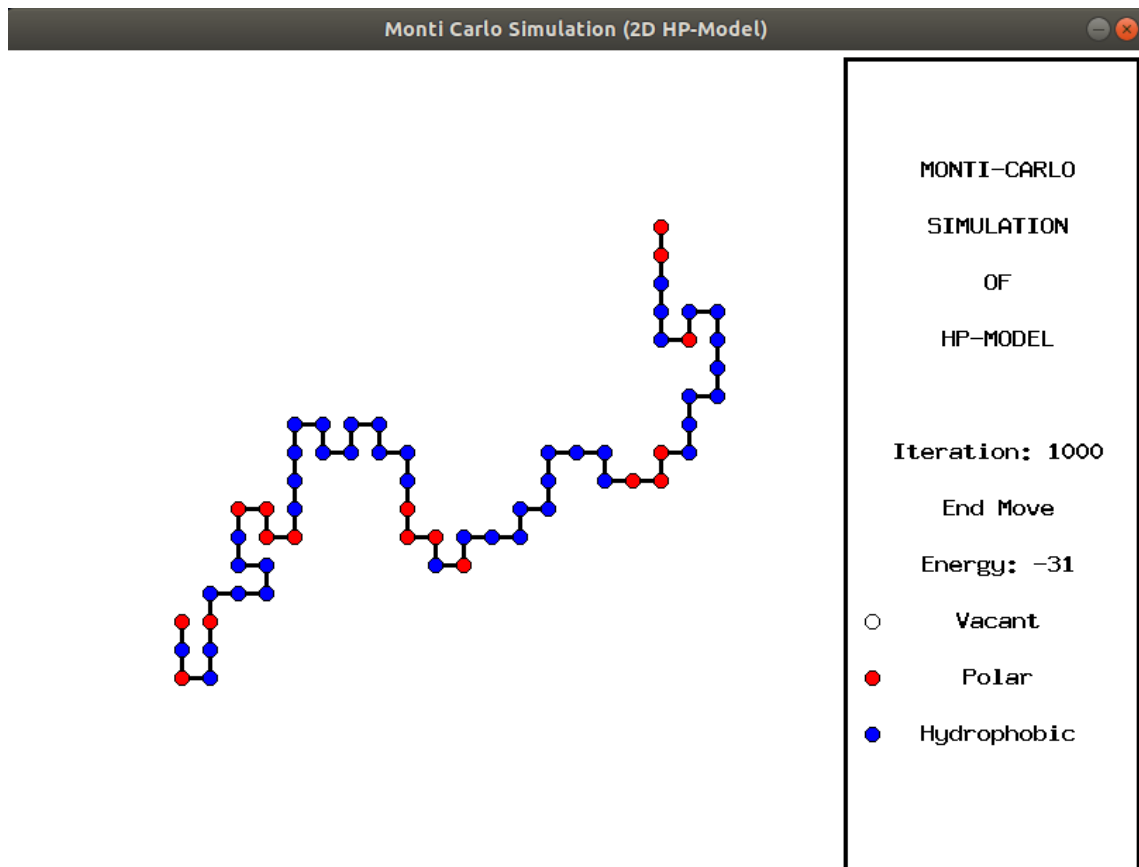
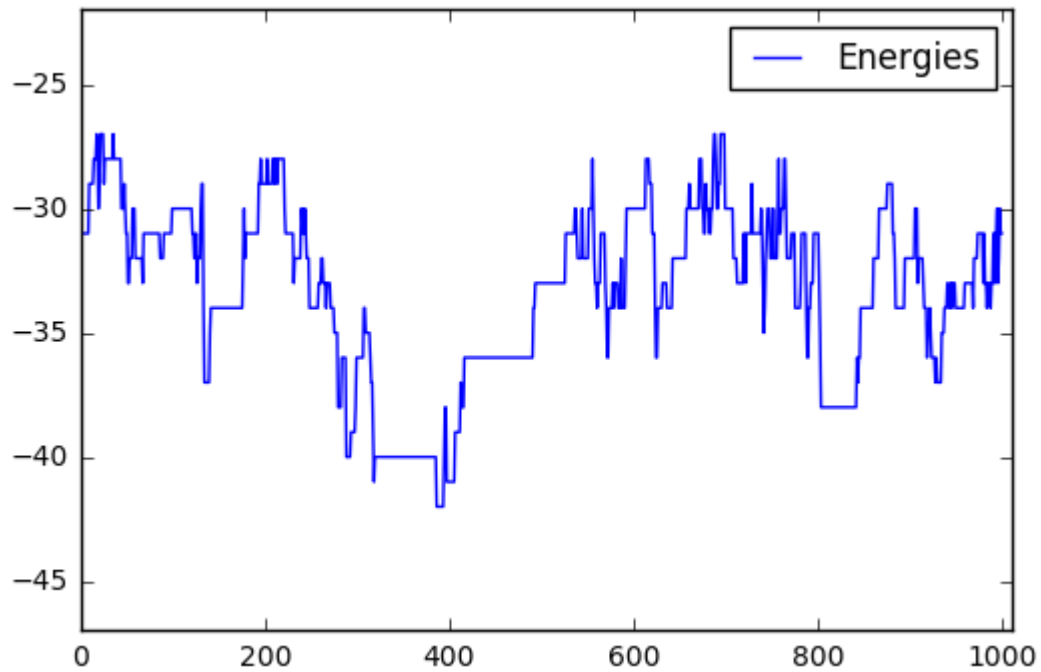


→ `STRING_06 = "H"*2+"PH"*3+"P"+"H"*4+"PH"+"("P"*3+"H")*2+"P"*4+"H"+`
`("P"*3+"H")*2+"PHP"+"H"*4+"HP"*3+"H"*2`



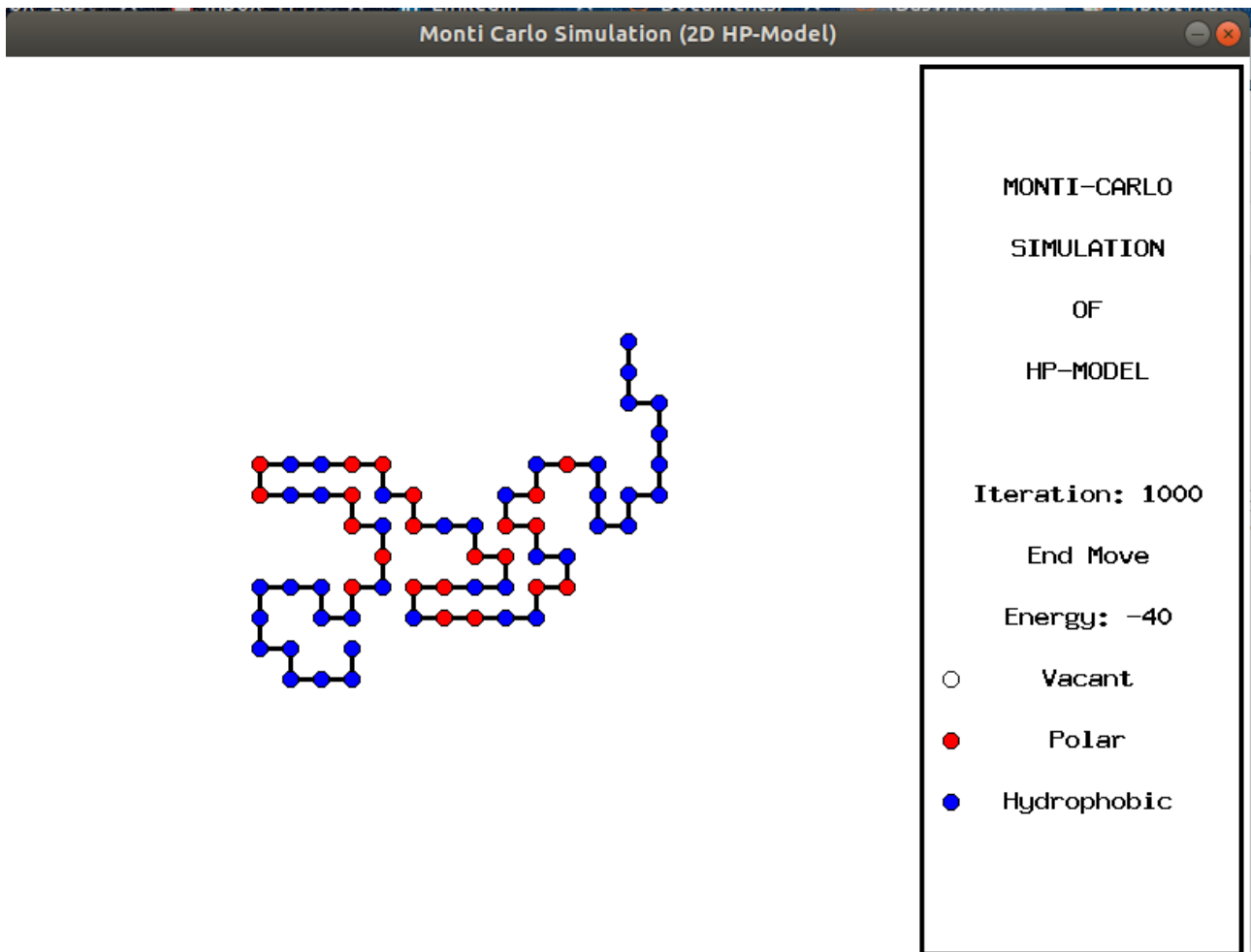
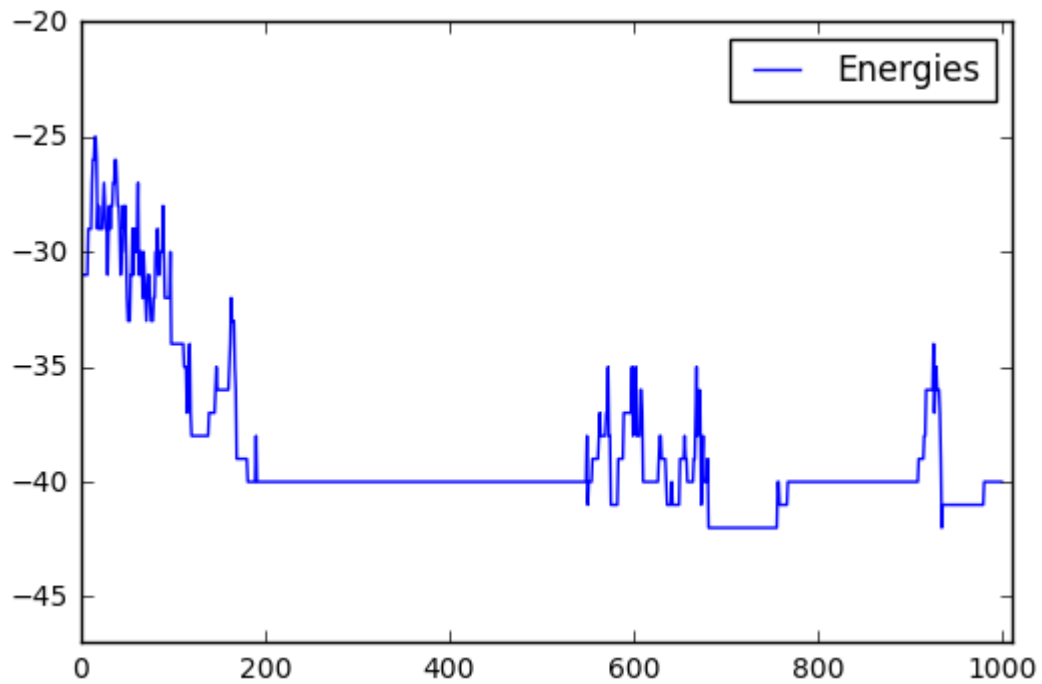
→ STRING_07 =

"PPHHHPHHHHHHHHHHPPPPHHHHHHHHHHHHHPHPPPHHHHHHHHHHHHHHHPPPPHHHHHHH
PHHHPH"

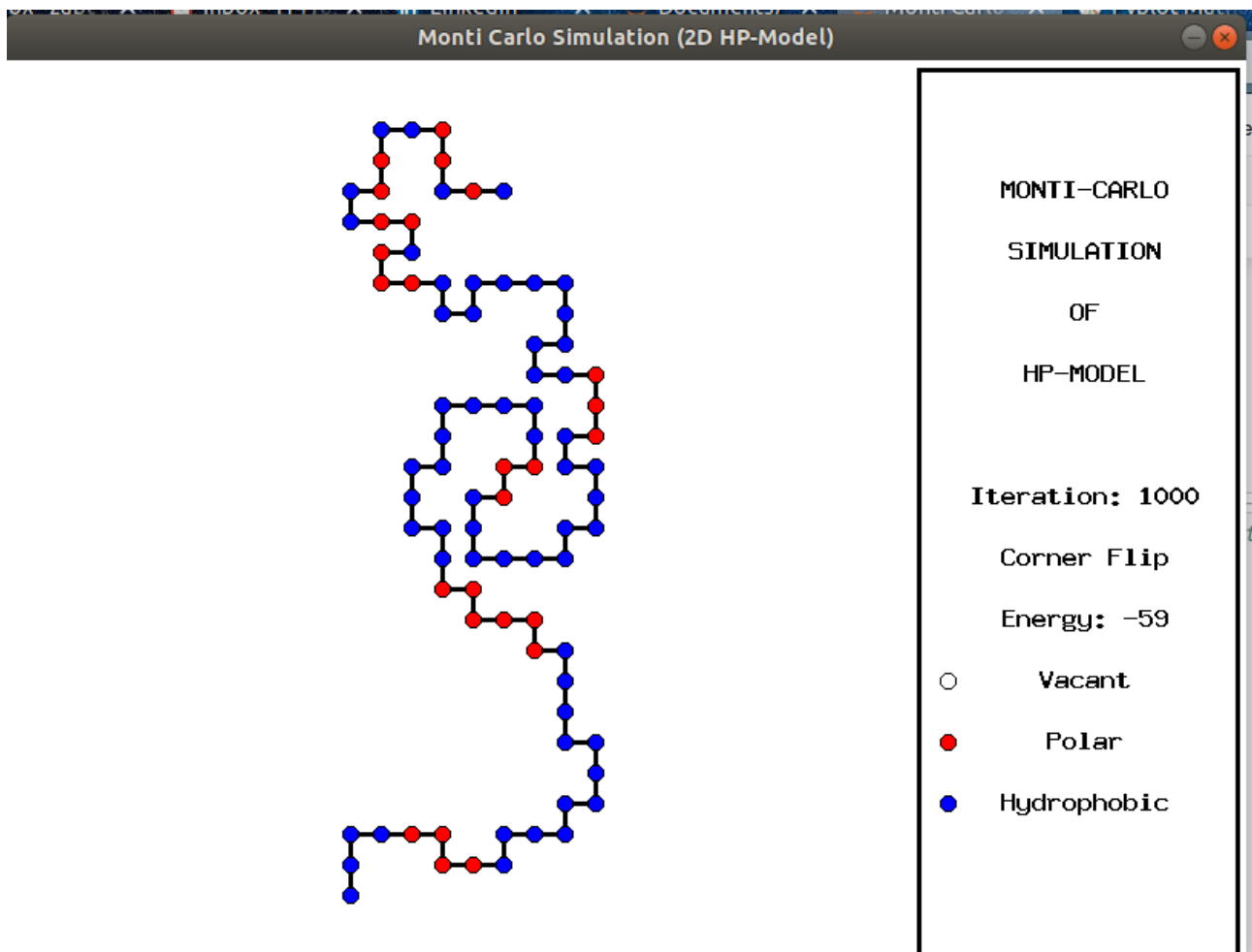
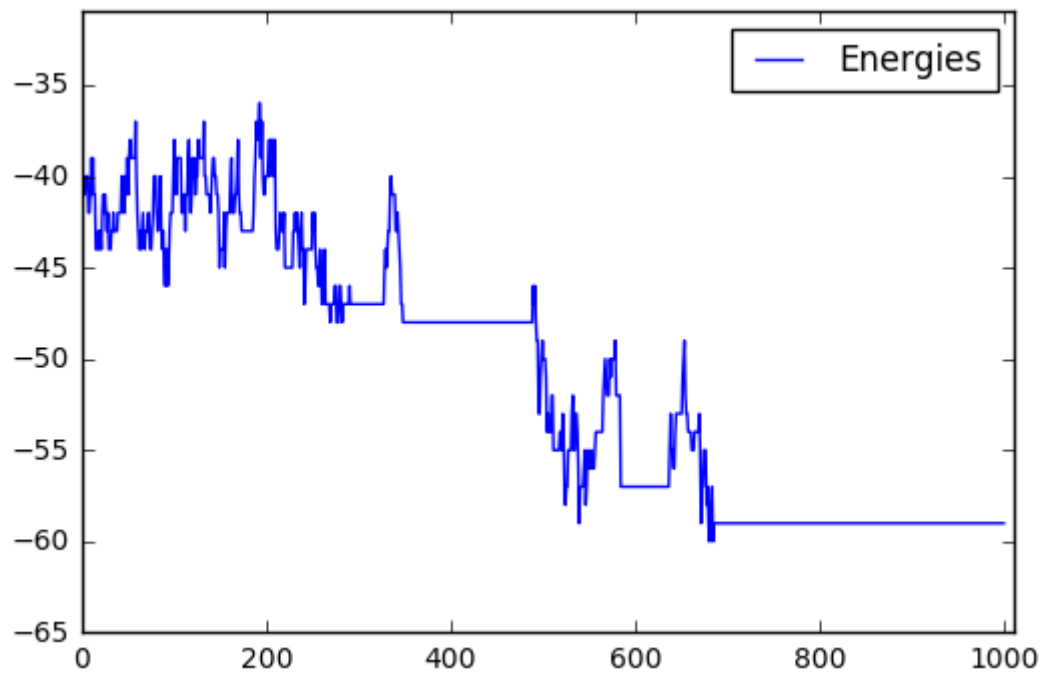


→ STRING_08 =

"H"*12+"PH"*2+"PPHH"*2+"PPHPPHHPPHHPPHHPPHHPPHHPPHP"+"H"*12

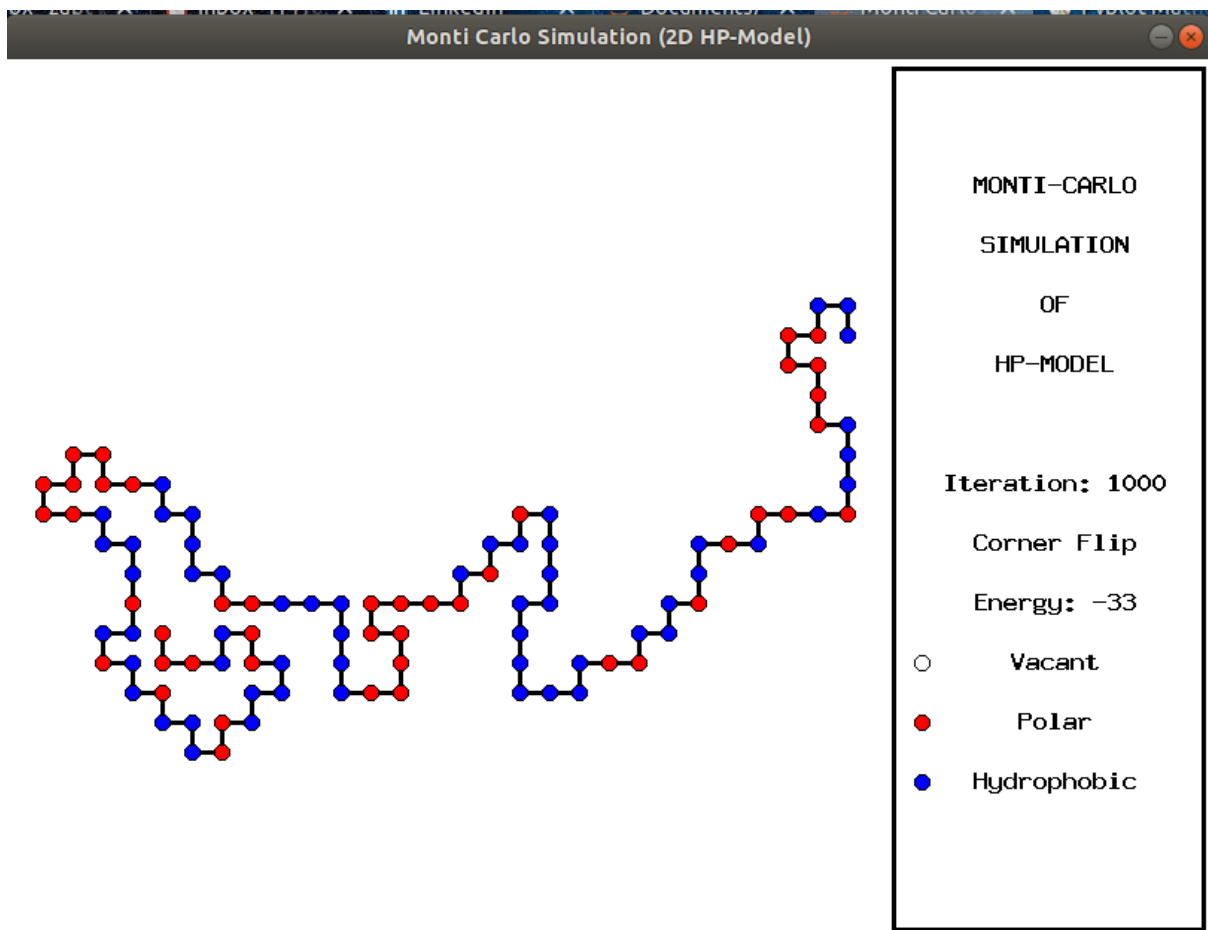
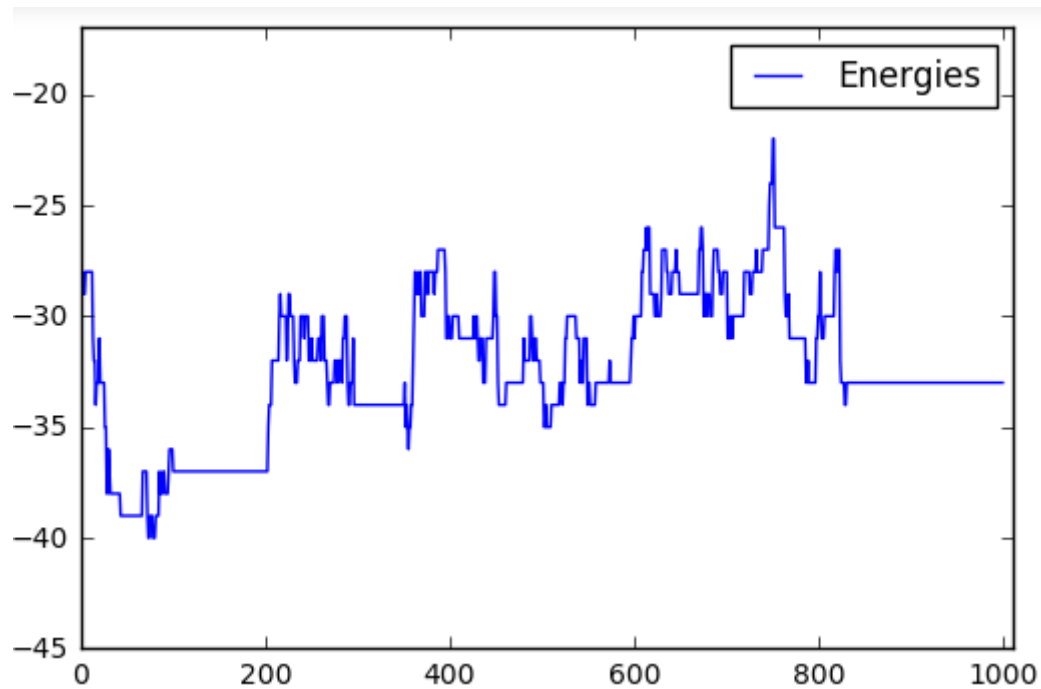


→ `STRING_09 = "HHHHPPPPHHHHHHHHHHHHHHPPPPPP" + ("H"*12+"P"*3)*3 + "HPPHHPPHHPPHPPH"`



→ STRING_10 =

```
"PPPHHPHHHHHPHHHHHPHHHPHHHPHHHHHPPPPPPPHHHHHHHPHHHHHHPPPPPPPP  
RHPHHHPHHHHHHHHHHHHHPHHHHHPHHHPHPHPHHHPHHHPPPPPPPHHH"
```



→ STRING_11 =

"PPPPPPHPPHHPPPPPPHHHHHPHHHHHHHPHHPPPPHPPHHHPHHHHHHHPHHHHHHHHHHHHHPH
HPHHHHHHHHPPPPPPPPPPPPHHHHHHHHHPHHPHHHPPPPPPPHPPHH"

