```python
import random
import numpy as np

# CONSTANTS
p = np.array([12, 24, 48, 70, 100, 120])
N = 120
n_trials = 100000


# FUNCTIONS
def generate_random_matrix(n):
    r = np.random.randint(2, size=(n, N))
    r[r == 0] = -1
    return r


def perform_one_trial(pattern_n):
    # Generate weight matrix
    patterns = generate_random_matrix(pattern_n)
    W = (1/N) * np.matmul(patterns.T, patterns)
    # np.fill_diagonal(W, 0)

    # Choose random pattern
    nu_index = random.randint(0, pattern_n - 1)
    chosen_pattern = patterns[nu_index]
    # Choose random neuron
    neuron_index = random.randint(0, N - 1)
    target_neuron_value = chosen_pattern[neuron_index]

    # Feed chosen pattern to network
    selected_weights = np.matrix(W[neuron_index]).T
    new_neuron_value = np.sign(np.dot(chosen_pattern, selected_weights))
    new_neuron_value = new_neuron_value.item()
    if new_neuron_value == 0:
        new_neuron_value = 1
    return new_neuron_value == target_neuron_value


def perform_n_trials(p, n):
    success = 0
    for i in range(n):
        success += perform_one_trial(p)
```

```
43      print("Error probability for " + str(p) + "
   patterns: " + str(1 - success / n))
44      return 1 - success / n
45
46
47  # MAIN CODE
48  pnt_vec = np.vectorize(perform_n_trials)
49  print(pnt_vec(p, n_trials))
50
```