NumPy basics

Use NumPy to efficiently solve the following problem. Do not use pure Python, instead mimic the approach on Page 109 of the PDA text.

You are playing the roulette wheel at the Mirage in Vegas. Your friend has rigged it so that the ball has a19/37 probability of landing in a red slot. (Ordinarily, the odds are 18/37.) The payoff is twice your bet.

You begin with $20. You can play only 100 times before the casino manager will catch on. Of course, you will bet on red, but the question is how much you should bet. If you bet the full amount, you have a18/37 of losing the very first time.

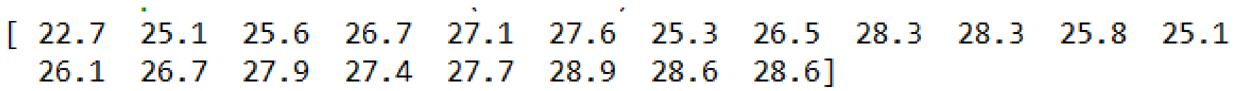
Compute the average payoff when you play with a 1$, 2$, 3$, etc. bet each time. Compute the average by performing 10000 simulations. Remember, when you bust, you cannot continue.

Solution:

Python Program

```
import numpy as np
import random
random.seed(1)

numTrials = 10000
trialLength = 100
startingStake = 20

np.set_printoptions(precision=1)

unifProbs = np.random.rand(numTrials,20,trialLength)
winProbs = unifProbs > (18.0/37.0)

state = np.zeros((numTrials,20))
state[:] = startingStake # initial values

newState = np.zeros((numTrials,20))
average = []

for i in xrange(10000):
    for j in xrange(20):
        for k in xrange(100):
            if winProbs[i][j][k]:
                if k == 0:
                    newState[i][j] = state[i][j] + (j + 1)
                else:
                    newState[i][j] = newState[i][j] + (j + 1)
            else:
                if k == 0:
                    newState[i][j] = state[i][j] - (j + 1)
                else:
                    newState[i][j] = newState[i][j] - (j + 1)

            if newState[i][j] > j:
                continue
            else:
                break

print newState.mean(axis = 0)
```

<u>Output</u>



```
[ 22.7  25.1  25.6  26.7  27.1  27.6  25.3  26.5  28.3  28.3  25.8  25.1
  26.1  26.7  27.9  27.4  27.7  28.9  28.6  28.6]
```