**Stock Prediction**

from \_\_future\_\_ import division

from math import sqrt

from operator import add

from heapq import heappush, heappop

def printTransactions(money, k, d, name, owned, prices):

def mean(nums):

return sum(nums) / len(nums)

def sd(nums):

average = mean(nums)

return sqrt(sum([(x - average) \*\* 2 for x in nums]) / len(nums))

def info(price):

cc, sigma, acc = 0, 0.0, 0

for i in range(1, 5):

if price[i] > price[i - 1]: cc += 1

sigma = sd(price)

mu = mean(price)

c1, c2, c3 = mean(price[0:3]), mean(price[1:4]), mean(price[2:5])

return (price[-1] - price[-2]) / price[-2]

infos = map(info, prices)

res = []

drop = []

for i in range(k):

cur\_info = info(prices[i])

if cur\_info > 0 and owned[i] > 0:

res.append((name[i], 'SELL', str(owned[i])))

elif cur\_info < 0:

heappush(drop, (cur\_info, i, name[i]))

while money > 0.0 and drop:

rate, idx, n = heappop(drop)

amount = int(money / prices[idx][-1]

if amount > 0:

res.append((n, 'BUY', str(amount)))

money -= amount \* prices[idx][-1]

print(len(res))

for r in res:

print(' '.join(r))

if \_\_name\_\_ == '\_\_main\_\_':

m, k, d = [float(i) for i in input().strip().split()]

k = int(k)

d = int(d)

names = []

owned = []

prices = []

for data in range(k):

temp = input().strip().split()

names.append(temp[0])

owned.append(int(temp[1]))

prices.append([float(i) for i in temp[2:7]])

printTransactions(m, k, d, names, owned, prices)

**Output**

