13. Implement a program to find out whether there exist M days within the last N(N >= M) trading days that the average closing price of these M days is at most P. Assume we have collected the history of the closing prices of the last N trading days for a stock. Requirements: Inputs are positive integer M and N, M <= N; An array of N float elements containing the closing prices of the last N trading days; And a float P. Please design and implement the program in C, C++, Java or Python to produce the answer in most time/space efficient way.

**Answer:**

Very simple implementation. Not sure about efficiency.

import numpy as np

import heapq

def mnp(m, n, prices, p):

return np.mean(heapq.nlargest(m, prices[-n:])) <= p

**# TODO**: test if the input satisfy requirements first

# (n < len(prices), m <= n, p and prices are numeric, m and n are integer,etc)

**Some test case:**

>>> np.random.seed(123)

>>> prices = 5 \* np.random.randn(100) + 100

>>> mnp(5, 50, prices, 100)

False

>>> mnp(5, 50, prices, 105)

False

>>> mnp(5, 50, prices, 110)

True

>>> mnp(25, 50, prices, 105)

True