Buy Maximum Stocks if i stocks can be bought on i-th day

> 给定一个股票价格数组price和总资金K，在第i天最多购买价格为price[i]的股票i支，计算K资金所能够获得的最多的股票数量

1. 创建map<股票价格，可购买数量>
2. 从前往后遍历map，对于当前的<股票价格，可购买数量>pair来说，可获得的股票数量最多为min(可购买数量, K / 股票价格)

```
#include <bits/stdc++.h>

using namespace std;

int maxStocks(vector<int> &price, int &K) {
    map<int, int> m;
    int count = 0;
    for (int i = 0; i < price.size(); ++i) m[price[i]] = i + 1;
    for (auto it = m.begin(); it != m.end() && K > 0; ++it) {
        count += min(it->second, K / it->first);
        K -= it->first * min(it->second, K / it->first);
    }
    return count;
}

int main() {
    int T;
    scanf("%d", &T);
    while (T--) {
        int N, K, num;
        scanf("%d %d", &N, &K);
        vector<int> price;
        for (int i = 0; i < N; ++i) {
            scanf("%d", &num);
            price.push_back(num);
        }
        printf("%d\n", maxStocks(price, K));
    }
    return 0;
}
```