COMPETITIVE CODING
with
RAMAN CLASSES

... An interactive series for those willing to learn and code

Problem :Beautiful And Ugly

Topic	: **Arrays**
Difficulty	: **EASY**
Programming Language	: **C++**
Time to Spend	: **25 min**

Problem Statement

Problem:

In a stock market, there is a product with its infinite stocks. The stock prices are given for n days, where a[i] denotes the price of the stock on the ith day. There is a rule that a customer can buy at most i stock on the ith day. If the customer has an amount of k dollars initially, find out the maximum number of stocks they can buy?

For example, for 3 days the price of a stock is given as 7,10,4. You can buy 1 stock worth 7$ on day 1,2 stocks worth 10$ each on day 2 and 3 stocks worth 4$ each on day 3. If k=100$, you can buy all the stocks (total 6) for 39$

Problem Statement

Input :

```
3
10 7 19
45
```

Output :

```
4
```

Input Description :

The first line contains an integer n denoting the number of days.

The next line contains n space-separated integers where ith integer denotes the price of the stock on the ith day. Next line contains a positive integer k which is the initial amount with the customer.

Problem Statement

Output Description :

Print the maximum number of stock that a customer can buy.

Let Us Revise

In order to solve this problem, go through the following concepts.

1. Usage of Vectors and pairs.
2. Greedy algorithm.

Problem Description

We are given a stock prices of an infinite stock for the next n days and we have k amount of money. So the problem wants us to find out the total number of stocks we can buy.

Remember we can buy only i number of stock on ith day.

For example: if for next 3 days the stocks prices are:
[10, 7, 19] and we have $45 in hands.
we can purchase 1 stock on day 1,2 stock on day 2 and 1 stock day 3 for 10,7*2=14 and 19 respectively.
Hence, total amount is10+14+19=43 and number of stocks purchased is 4.

Output:4

Let Us Think

In order to solve this problem, let us think and analyse how to get started with this problem.

Try to draw different examples on paper. On careful observation, you can see its not necessary to buy the stocks continuously starting from day 1. We can buy stocks which ever day we want. So the best case scenario is buying stocks on the day its price is low.

So, problem boils down to sorting the prices and calculate the total number of stocks accordingly.

Let Us Think

Now you know the logic. Lets proceed with the code.

Let Us Code

Now you know the logic. Lets proceed with the code.

Things we need to do for this problem:

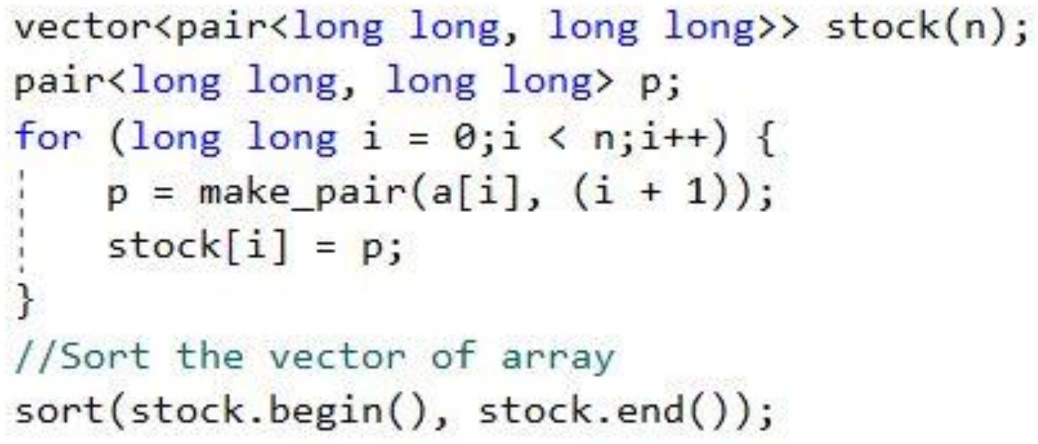
1. Find a sorted vector pair of price and the number of items we can buy on that day.

2. Calculate the total till money is less than the total amount need to buy every stock on that day.

3. Find out the number of stocks which can be bought using the money left on that day and add to total.

Let Us Code

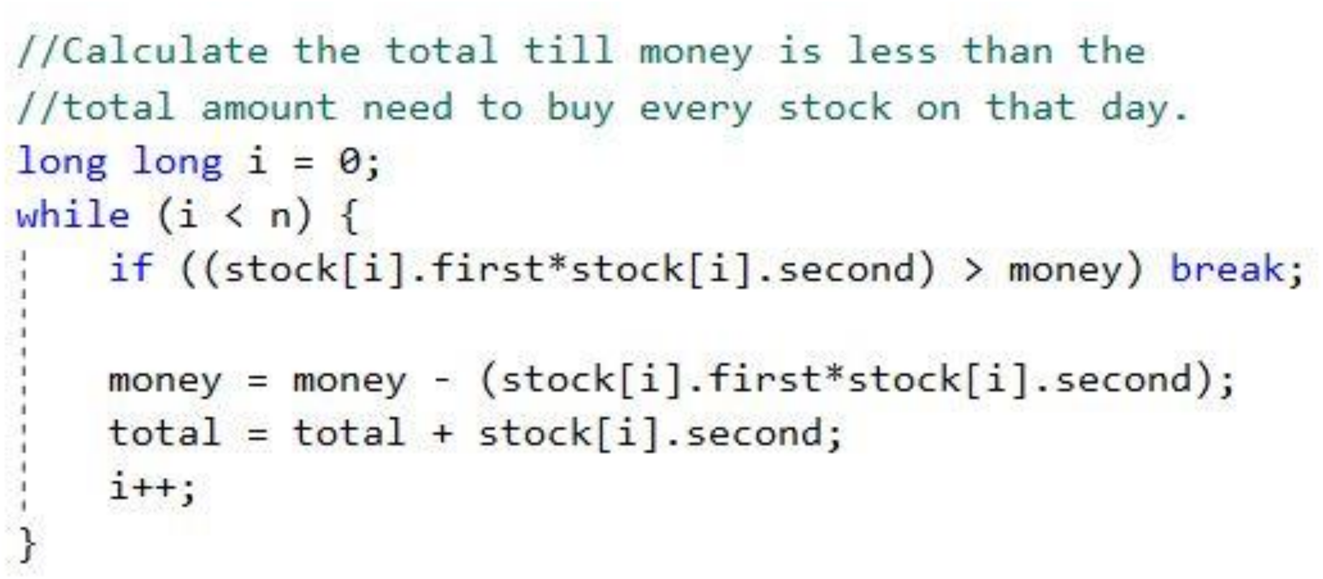
1. Find a sorted vector pair of price and the number of items we can buy on that day.



```
vector<pair<long long, long long>> stock(n);
pair<long long, long long> p;
for (long long i = 0;i < n;i++) {
    p = make_pair(a[i], (i + 1));
    stock[i] = p;
}
//Sort the vector of array
sort(stock.begin(), stock.end());
```

Let Us Code

2. Calculate the total till money is less than the total amount need to buy every stock on that day.

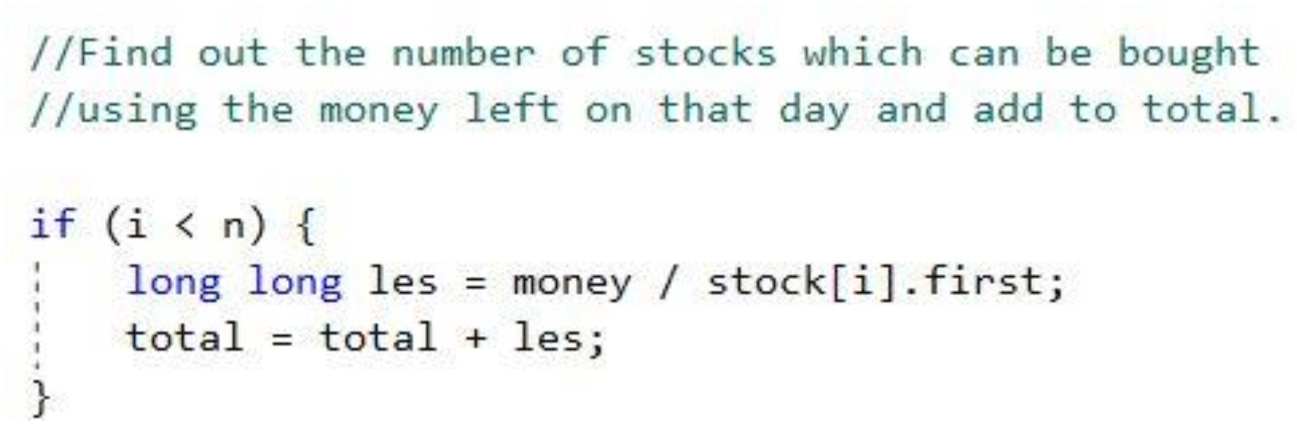


```
//Calculate the total till money is less than the
//total amount need to buy every stock on that day.
long long i = 0;
while (i < n) {
    if ((stock[i].first*stock[i].second) > money) break;

    money = money - (stock[i].first*stock[i].second);
    total = total + stock[i].second;
    i++;
}
```

Let Us Code

3. Find out the number of stocks which can be bought using the money left on that day and add to total.



```
//Find out the number of stocks which can be bought
//using the money left on that day and add to total.

if (i < n) {
    long long les = money / stock[i].first;
    total = total + les;
}
```

Thank You !

... brought to you by

RAMAN CLASSES

FULFILL YOUR DREAMS OF SUDYING IN IIT's

www.ramanclasses.in