```cpp
#include <iostream>

#include <bits/stdc++.h>

using namespace std;



// Profit for each type of stock


int KProfit( vector<vector<int>> & prices , int type, int k  , int fee) {


   int n = prices[type].size();


   if(n < 2) return 0;


 // for a single transaction to complete buy and sell  must be completed

 // for k transactions -  k buys and k selling must be completed.


    vector<int> after(2*k+ 1 , 0);

    vector<int> curr(2*k+ 1 , 0);


for (int ind = n - 1; ind >= 0; ind--) {

     for (int transac = 2*k-1; transac >= 0; transac--) {


            if(fee>=1){  //for transactions with fee


                if (transac % 2 == 0) { // We can buy the stock


              curr[transac] = max( after[transac],   -prices[type][ind] + after[transac+1]);

                }


            else { // We can sell the stock

             curr[transac] = max(after[transac], prices[type][ind] -fee + after[transac+ 1]);
```

```
          }

       }

        else{

          if (transac % 2 == 0) { // We can buy the stock

            curr[transac] = max(0 + after[transac],   -prices[type][ind]  + after[transac+1]);
          }

          else { // We can sell the stock
            curr[transac] = max(0 + after[transac], prices[type][ind] + after[transac+ 1]);
          }

       }

    }
    after = curr;
 }

// The result is stored in after[0] which represents maximum profit after the final transaction.
return after[0];
}
```

```
   // calling for each type of stock &&  adding to Final profit

   int kbuys(vector<vector<int>> & prices , int k, int fee){
   int profit = 0;

   for(int i = 0 ; i< prices.size(); i++){

     profit = profit + KProfit(prices , i , k , fee);
   }
   return profit;
   }

   // Driver code

int main()
{
 cout <<"Calculating Total profit for n different stocks for m days" << endl;

   cout<< "Enter how many types of stock" << "\n";
   int  stock  ;
   cin>> stock ;
```

```
  if(stock > 10 && stock < 1){
     cout << "maximum type of stocks is 10 "<< "\n" << "please enter again " << "\n";
     cin >>stock;
  }

  cout << "Enter no.of days  "  << "\n";
  int days;
  cin >> days;

  if(days > 10 && days < 1){
     cout << "maximum no of days is 10 "<< "\n" << "please enter again "  << "\n";
     cin >> days ;
  }


 vector<vector<int>> prices(stock,vector<int> (days));

  // Input values for the stock prices from the user

  for (int i = 0; i < stock; ++i) {
     for (int j = 0; j <days; ++j) {
cout << "Enter price of stock-type " << i+1 << " for day " << j+1 << ": ";
        cin >> prices[i][j];
     }
  }




//calling function for single transaction
```

```
 cout << "Total profit for " << "1" << "  transaction" << " " <<"is " << ":" <<kbuys(prices, 1 , 0) <<
endl ;


//calling function for unlimited transactions: maximum unlimited transactions = day/2


 cout << "Total profit for " << " unlimited "<< "  transactions" << " " <<"is " << ":" << kbuys(prices,
days/2, 0) << endl ;




// For a custom number of transactions


cout << "Enter number for a custom no.of transactions to make: "<< endl;


   int transc;
   cin>> transc;


if(transc > 10 && transc != 0){
cout << "transaction number invalid  enter again " << endl;
cin>> transc;
   }


cout << "Total profit for  " <<  transc << "  transactions" << " " <<"is " << " " << kbuys(prices, transc
, 0) << endl ;



// with Transaction fee


 cout << "would you like to try with a transaction fee? " << endl;
 cout << "To skip enter  0 "<< endl;
 cout <<"To proceed enter a value  : " << endl;
```

```
  int fee;

  cin >>fee;


if(fee > 0){


cout << "Net profit for " << "1" << "  transaction" << "  after deducting transaction fee :"
<<kbuys(prices, 1 , fee) << endl ;

cout << "Net profit for " << "unlimited"<< "  transactions" << "   after deducting transaction fee: "
<< kbuys(prices, days/2 , fee) << endl ;

cout << "Net profit for  " <<  transc << "  transactions" << " after deducting transaction fee: "<<
kbuys(prices, transc , fee) << endl ;

}



   return 0;

}
```