Step 1)
Input Matrix:

```
A =   [12, 1, 5, 3, 16],      // Stock 1
      [4, 4, 13, 4, 9],       // Stock 2
      [6, 8, 6, 1, 2],        // Stock 3
      [14, 3, 4, 8, 10]       // Stock 4
```

Step 2)
STOCK 1:

BUY DAY	SELL DAY	PROFIT
1	2	-11
1	3	-7
1	4	-9
1	5	4
2	3	4
2	4	2
2	5	15
3	4	-2
3	5	11
4	5	13

STOCK 2:

BUY DAY	SELL DAY	PROFIT
1	2	0
1	3	9
1	4	0
1	5	5
2	3	9
2	4	0
2	5	5
3	4	-9
3	5	-4
4	5	5

STOCK 3:

BUY DAY	SELL DAY	PROFIT
1	2	2
1	3	0
1	4	-5
1	5	-4
2	3	-2
2	4	-7
2	5	-6
3	4	-5
3	5	-4
4	5	1

STOCK 4:

BUY DAY	SELL DAY	PROFIT
1	2	-11
1	3	10
1	4	-6
1	5	-4
2	3	1
2	4	5
2	5	7
3	4	4
3	5	6
4	5	2

Step 3)
Stock 1 most profitable transaction:
Buy at day 2 for 1 and sell at day 5 for 16. Profit = 15

Stock 2 most profitable transaction:
Buy at day 1 for 4 and sell at day 3 for 13. Profit = 9

Stock 3 most profitable transaction:
Buy at day 1 for 6 and sell at day 2 for 8. Profit = 2

Stock 4 most profitable transaction:
Buy at day 2 for 3 and sell at day 5 for 10. Profit = 7

Step 4)
Maximum profit is 15 from Stock 1 – Buy day 2 and sell day 5
OUTPUT: (1, 2, 5, 15)

Problem Statement 2

You are given a matrix A of dimensions m × n, where each element represents the predicted prices of m different stocks for n consecutive days. Additionally, you are given an integer k (1 ≤ k ≤ n). Your task is to manually find a sequence of at most k transactions, each involving the purchase and sale of a single stock, that yields the maximum profit.

Step 1)

Input Matrix:

A = [[25, 30, 15, 40, 50], // Stock 1
[10, 20, 30, 25, 5], // Stock 2
[30, 45, 35, 10, 15], // Stock 3
[5, 50, 35, 25, 45]] // Stock 4

Step 2)

STOCK 1:

BUY DAY	SELL DAY	PROFIT
1	2	5
1	3	-10
1	4	15
1	5	25
2	3	-15

2	4	10
2	5	20
3	4	25
3	5	35
4	5	10

STOCK 2:

BUY DAY	SELL DAY	PROFIT
1	2	10
1	3	20
1	4	15
1	5	-5
2	3	10
2	4	5
2	5	-15
3	4	-5
3	5	-25
4	5	-20

STOCK 3:

BUY DAY	SELL DAY	PROFIT
1	2	15
1	3	5
1	4	-20
1	5	-15
2	3	-10
2	4	-35
2	5	-30
3	4	-25
3	5	-20
4	5	5

STOCK 4:

BUY DAY	SELL DAY	PROFIT
1	2	45
1	3	30
1	4	20
1	5	40
2	3	-15

2	4	-25
2	5	-5
3	4	-10
3	5	10
4	5	20

Step 3)

Finding optimal non-overlapping transactions (k=3):

Best profitable transactions:

1. Stock 4: (1,2) → Profit = 45
2. Stock 1: (3,5) → Profit = 35
3. Stock 4: (1,5) → Profit = 40
4. Stock 1: (1,5) → Profit = 25
5. Stock 1: (3,4) → Profit = 25

Selected non-overlapping transactions:

- Transaction 1: Stock 4, Buy Day 1, Sell Day 2 → Profit = 45
- Transaction 2: Stock 1, Buy Day 3, Sell Day 5 → Profit = 35

Total Maximum Profit = 80

Output: [(4,1,2), (1,3,5)]

Problem 3

Problem Statement

You are given a matrix A of dimensions m × n, where each element represents the predicted prices of m different stocks for n consecutive days. Additionally, you are given an integer c ($1 \leq c \leq n - 2$). Your task is to determine the maximum profit achievable under the given trading restrictions, where you cannot buy any stock for c days after selling any stock. If you sell a stock on day i, you are not allowed to buy any stock until day i + c + 1.

Input

	Day 1	**Day 2**	**Day 3**	**Day 4**	**Day 5**	**Day 6**	**Day 7**
Stock 1	7	1	5	3	6	8	9
Stock 2	2	4	3	7	9	1	8
Stock 3	5	8	9	1	2	3	10
Stock 4	9	3	4	8	7	4	1
Stock 5	3	1	5	8	9	6	4

Cooldown: c = 2

First let's define best(t) as the maximum profit achievable with one buy, sell between days t and 7.

t	Window	Best trade	$best_1(t)$
1–4	Days 1 to 7	buy Stock 3 on day 4 at 1 then sell on day 7 at 10	9
5	Days 5 to 7	buy Stock 3 on day 5 at 2 then sell on day 7 at 10	8
6	Days 6 to 7	buy Stock 2 on day 6 at 1 then sell on day 7 at 8	7
7	day 7	no transaction	0

Now for each possible first trade (i, j, l), let's compute $Profit_1 = P[\, i \,, l \,] - P[\, i \,, j \,]$, earliest next buy $B = l + 3$, $Profit_2 = best_1(B)$, and $Total = Profit_1 + Profit_2$. Here are the top contenders:

Buy j	First trade (i , l)	$Profit_1$	Next B=l +3	$Profit_2$	Total
j = 1	Stock 2: buy	7	8 to 0	0	7

	1, sell 5				
j = 2	Stock 1: buy 2, sell 3	4	6 to 7	7	11
j = 3	Stock 2: buy 3, sell 5	6	8 to 0	0	6
j = 4	Stock 3: buy 4, sell 7	9	10 to 0	0	9
j = 5	Stock 3: buy 5, sell 7	8	10 to 0	0	8
j = 6	Stock 2: buy 6, sell 7	7	10 to 0	0	7

The best total (11) come from:

1. Stock 1: buy on day 2 at 1 then sell on day 3 at 5 ($Profit_1 = 4$) with the cooldown until day 6 (3 + 2 + 1)

2. Stock 2: buy on day 6 at 1 then sell on day 7 at 8 ($Profit_2 = 7$)

Total profit = 4 + 7 = 11.

Final Answer

Maximum profit: 11

Trades: (i = 1, j = 2, l = 3) and (i = 2, j = 6, l = 7)