1)

Students attendance :

write a program to read the attendance states for 20 students in 15 weeks.

	W0	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
S0	**3**	**0**	**0**	**2**	**0**	**0**	**0**	**0**	**2**	**1**	**1**	**0**	**1**	**2**	**2**
S1	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**0**
S2	**1**	**1**	**1**	**2**	**2**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**5**	**4**
…	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**
..	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**
..	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**
S19	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**

No absents during week 9

This 2 times absents during week 5

Then display the following :

- Average absents for each student.
- The student with the maximum absents.
- The student with minimum absents.

2)

Identity matrix :

In this program you need to read a matrix from the user (assume with size 100 x 100).

The program will notify the user if the matrix is identity or not.

1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

Identity

1	0	0	0	0
0	1	0	0	0
0	0	5	0	0
0	0	0	1	0
0	0	0	0	1

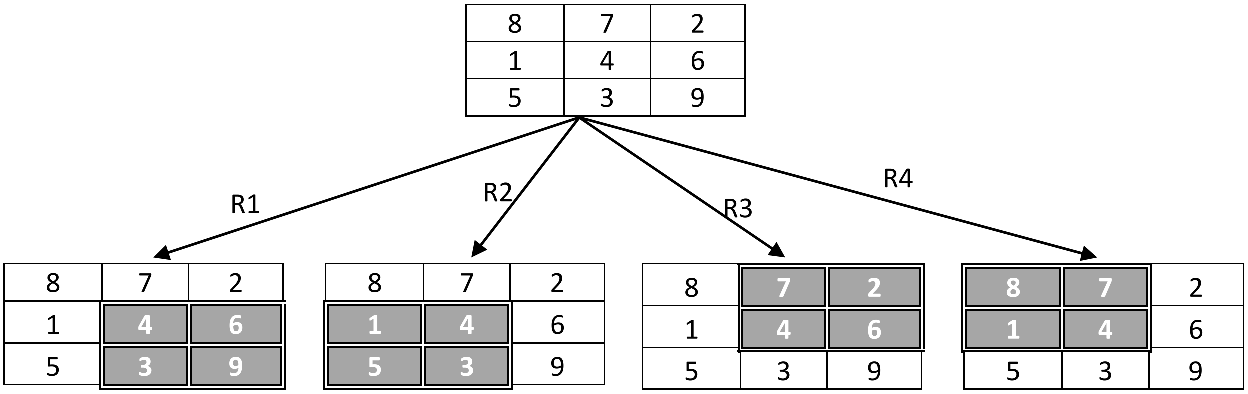
Not Identity

3)Rotate Game :

in this game the board (3x3) which started by unsorted numbers (1:9)

the board logically divided into 4 regions

R1 → bottom-right
R2 → bottom-left
R3 → top-right
R4 → top-left



8	7	2
1	4	6
5	3	9

R1

8	7	2
1	**4**	**6**
5	**3**	**9**

R2

8	7	2
1	**4**	6
5	**3**	9

R3

8	**7**	**2**
1	**4**	**6**
5	3	9

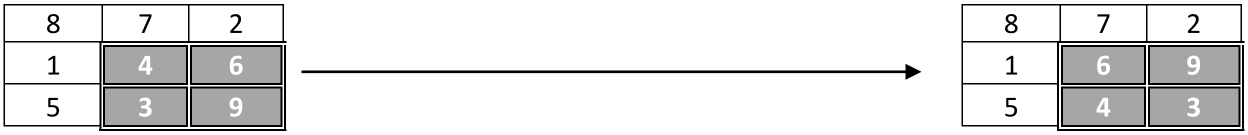
R4

8	**7**	2
1	**4**	6
5	3	9

The user will select both : The region & the direction.

e.g.

R1 & Left direction



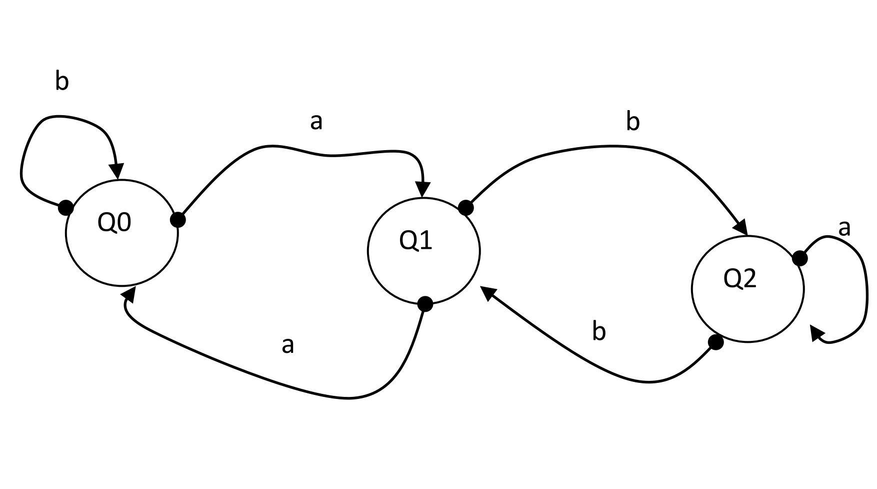
8	7	2
1	**4**	**6**
5	**3**	**9**

8	7	2
1	**6**	**9**
5	**4**	**3**

The user will repeat the steps till the puzzle becomes sorted.

4)

Finite State Machine:
Represented either graphically or as matrix



a	b
1	0
0	2
2	1

Your program will read the matrix that represents a given finite state machine,
Also you have to read a string that contains (a’s & b’s).
Finally your program will display (accepted in case the machine ended at the final state).

Example:

aabbbab	→	accepted
aabbbabaab	→	not accepted

5)

Club rank:
Write a program to read the rank of 20 clubs in the previous 100 years.
Then display the following :

- Display the maximum rank of the club , which selected by the user.
- Display each club , which get the 1st rank.
- Display the club of the century.

6)

- Write a program to read a matrix (100 x 100) from the user.
- Ask the user to select a row.
- Ask the user to select either
  - The right diagonal or
  - The left diagonal.
- Then , swap the selected row with the selected diagonal

e.g.

3	**4**	**5**	**6**
2	**4**	**66**	**78**
90	**34**	**56**	**79**
12	**43**	**67**	**68**

The selected row is : 2

The selected diagonal is: the right

3	**4**	**5**	**79**
2	**4**	**56**	**78**
12	**34**	**66**	**6**
90	**43**	**67**	**68**

7)

Puzzle: In this game your board contains (4 x 4) , that include unsorted values from (1 to 15)

Set only one cell by zero (the empty cell).

Let the player select cell to move , (the selected cell should be one of the 4 cells around the empty cell)

1	**5**	**10**	**15**
2	**9**		**14**
13	**8**	**11**	**7**
12	**4**	**3**	**6**

Selected cell
(2 , 2)

→

1	**5**	**10**	**15**
2	**9**	**11**	**14**
13	**8**		**7**
12	**4**	**3**	**6**

repeat the task till, the user make the puzzle sorted.

8 –)

255	0	0	255	128	128
255	255	0	128	128	218
180	180	200	255	128	128
128	128	128	128	128	128
255	255	0	0	0	0
0	0	128	128	0	128

85	85	85	170	170	170
170	170	170	158	158	158
186	186	186	170	170	170
…	…	…	…	…	…
…	…	…	…	…	…
…	…	…	…	…	…

Read a matrix (99 x 99) , then for each 3 contiguous cells , replace them to the their average values.

(255 + 0 + 0) / 3 → 85

(255 + 128 + 128) / 3 → 170

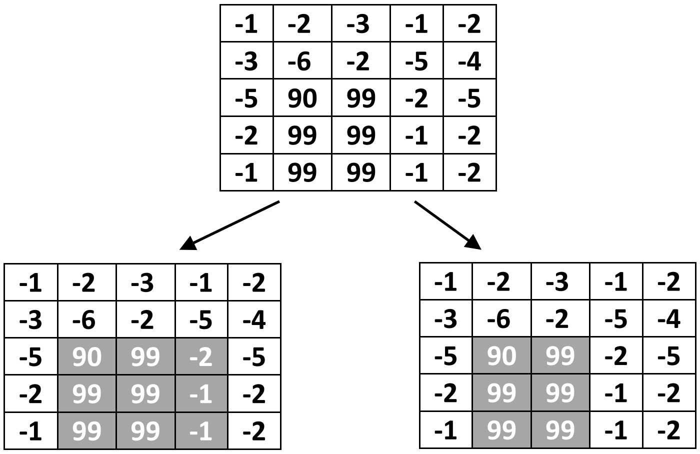
…

…

9)

Write a program to read a matrix (20x20) from the user and then do the following:

- **Determine the square which includes the maximum sum.**
- **Determine the rectangle which includes the maximum sum.**



-1	-2	-3	-1	-2
-3	-6	-2	-5	-4
-5	90	99	-2	-5
-2	99	99	-1	-2
-1	99	99	-1	-2

-1	-2	-3	-1	-2
-3	-6	-2	-5	-4
-5	90	99	-2	-5
-2	99	99	-1	-2
-1	99	99	-1	-2

-1	-2	-3	-1	-2
-3	-6	-2	-5	-4
-5	90	99	-2	-5
-2	99	99	-1	-2
-1	99	99	-1	-2

Largest square:
[(2,1) → (4,3)]
90+99+ -2
99+99+ -1
99+99+ -1

Largest rectangle:
[(2,1) → (4,2)]
90+99
99+99
99+99

10)

Write a program to read a matrix (20x20) from the user and then do the following:

- Display the triangle with the largest sum.

-1	-2	-3	-1	-2
-3	-6	-2	-5	-4
-5	90	99	-2	-5
-2	99	99	-1	-2
-1	99	99	-1	-2

-1	*-2*	-3	-1	-2
-3	*-6*	*-2*	-5	-4
-5	90	99	-2	-5
-2	99	99	-1	-2
-1	99	99	-1	-2

-1	-2	*-3*	-1	-2
-3	*-6*	*-2*	*-5*	-4
-5	90	99	-2	-5
-2	99	99	-1	-2
-1	99	99	-1	-2

-1	-2	-3	*-1*	-2
-3	-6	*-2*	*-5*	*-4*
-5	90	99	-2	-5
-2	99	99	-1	-2
-1	99	99	-1	-2

-1	-2	*-3*	-1	-2
-3	*-6*	*-2*	*-5*	-4
-5	*90*	*99*	*-2*	*-5*
-2	99	99	-1	-2
-1	99	99	-1	-2

-1	-2	-3	-1	-2
-3	*-6*	-2	-5	-4
-5	*90*	*99*	-2	-5
-2	99	99	-1	-2
-1	99	99	-1	-2

-1	-2	-3	-1	-2
-3	-6	*-2*	-5	-4
-5	*90*	*99*	*-2*	-5
-2	99	99	-1	-2
-1	99	99	-1	-2

-1	-2	-3	-1	-2
-3	-6	-2	*-5*	-4
-5	90	*99*	*-2*	*-5*
-2	99	99	-1	-2
-1	99	99	-1	-2

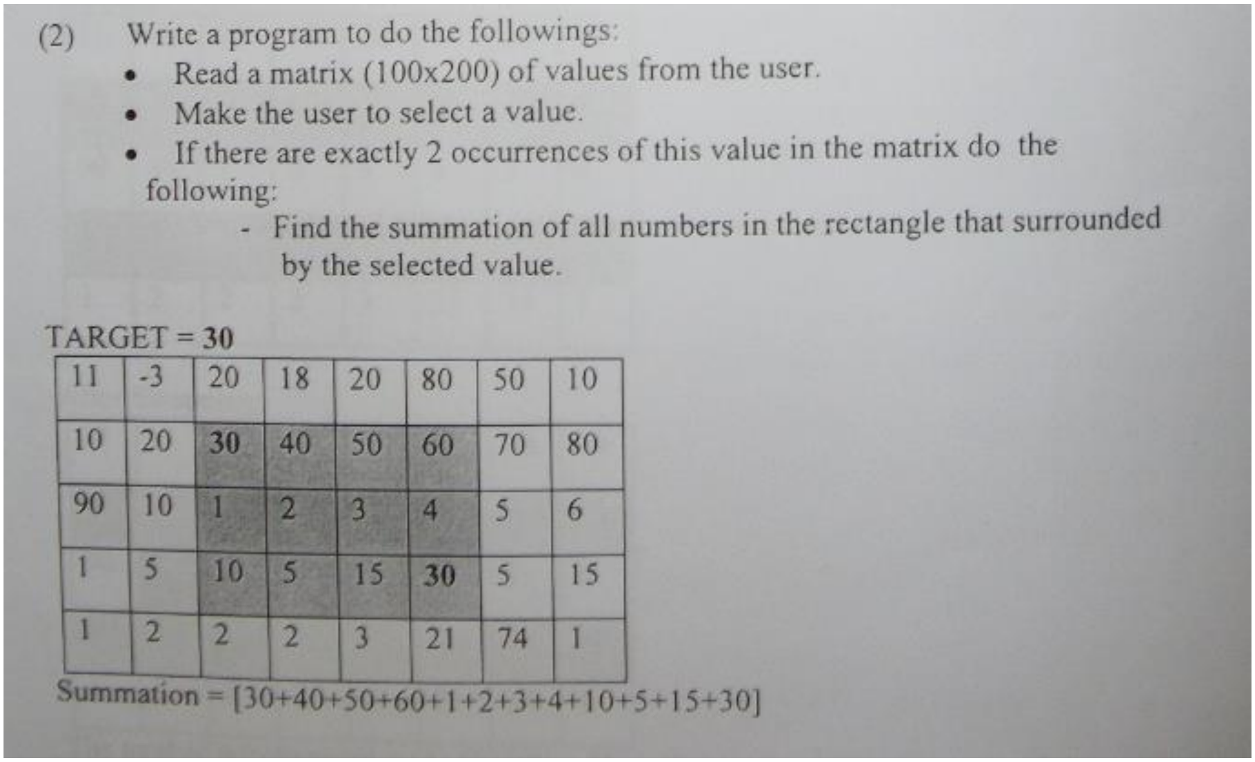
-1	-2	-3	-1	-2
-3	-6	*-2*	-5	-4
-5	*90*	*99*	*-2*	-5
-2	*99*	*99*	*-1*	*-2*
-1	99	99	-1	-2

…………….

…………….

…………….

11)



(2) Write a program to do the followings:

- Read a matrix (100x200) of values from the user.
- Make the user to select a value.
- If there are exactly 2 occurrences of this value in the matrix do the following:
  - Find the summation of all numbers in the rectangle that surrounded by the selected value.

TARGET = **30**

11	-3	20	18	20	80	50	10
10	20	**30**	40	50	60	70	80
90	10	1	2	3	4	5	6
1	5	10	5	15	**30**	5	15
1	2	2	2	3	21	74	1

Summation = [30+40+50+60+1+2+3+4+10+5+15+30]