Solution for a)

This function is to rotate a matrix clockwise by 90 degrees.

The first parameter is the size of array, the second parameter is the original matrix with 2d array, the third parameter is the new matrix with 2d array.

For example, if n=4, the a is as follows:

11 12 13 14

15 16 17 18

19 20 21 22

23 24 25 26

After this function, the b will be as follows:

23 19 15 11

24 20 16 12

25 21 17 13

26 22 18 14

b[j][n-1-i] = a[i][j]; this sentence means that put the first row of a to the last column of b, and then do the same rotations.

Solution for b)

This function is to find the minimum value in an array.

The first parameter is the size of array, the second parameter is an integer array.

This is a recursive method, the base case is when n equals 1, the recursive case is n>1.

For example, if n=5, and a= 11,12,3,4,5

After this function, the 3 will be returned.

if (m > a[n-1])

return a[n-1]

else

return m;

the above sentences ensure to return the minimum value of this array.

Solution for c)

This function is a program segment to reverse an array.

This first parameter is size of array, the second parameter is this array.

In the function f\_c, there is an incursive function: f\_c1. There are three parameters: the front position, the rear position and the integer array.

The basic idea is the front position: i move from 0 to the direction of end. The rear position: j moves from the end to the direction of beginning. In each turn, swap the value of i and j.

+-------------------+

| f\_c1(i=3,j=2, a={5,4,3,2,1}) |

| return |

+-------------------+

| f\_c1(i=2,j=3, a={5,4,3,2,1}) |

+-------------------+

| f\_c1(i=1,j=3, a={5,2,3,4,1}) |

+-------------------+

| f\_c1(i=0,j=4, a={1,2,3,4,5}) |

+-------------------+

| f\_c(n=5, a={1,2,3,4,5}) |

+-------------------+

| main |

+-------------------+