Ex2:

It means getting a pointer that contains the adress of that imaginary location. It causes undefined behavior and cause errors if the memory is segmented. Its behavior can change from compiler to compiler.

p = pointer pointed at the first element of p..

\*p = value at the first element of p.

&p = adress of the first element.

\*(p+1) = value at second element of p

\*p + 1 = pointer to the second element of p

q[0] = p[ – 1] is undefined , p[9], p[10] which is beyond memory so undefined behavior

M is a pointer pointing to a pointer pointing to an integer.

\*M is a pointer to the first slice of M.

\*\*M is a value, first element of the first slice of M.

M[1][3] is 8.

\*(M[0]+1) is 1.

\*(\*(M+1)+3) is 8.

It depends.

#include <iostream>

using namespace std;

int main()

{

int \*\* M = new int \*[2];

M[0] = new int[5]{0,1,2,3,4};

M[1] = new int[5]{5,6,7,8,9};

int temp;

for (int i=0; i<5; i++){

temp = \*(\*M + i);

\*(\*M + i) = \*(\*(M + 1) + 4 - i);

\*(\*(M + 1) + 4 - i) = temp;

}

for (int i=0; i<5; i++){

printf("%d ",M[0][i]);

}

for (int i=0; i<5; i++){

printf("%d ",M[1][i]);

}

return 0;

}