**Algorithm**

1. Start
2. Declare an variable (say **i ,mul ,n**)
3. Declare a handle (pointer; say **a, b** )
4. Allocate memory for **(n)** elements dynamically for a and b
5. Initialize **i**
6. Get the value for the array **a** from the user
7. Increment **i**  with 1
8. Repeat step 5 until it reaches **n** count
9. Variable I should sum with 1 and store in handle pointer(pa)
10. Initialize **i**
11. Get the value for the array **a** from the user
12. Increment **i**  with 1
13. Repeat step 10 until it reaches **n** count
14. Display first array
15. Initialize i
16. Get the value for the array **b** from the user
17. Increment **i**  with 1
18. Repeat step 15 until it reaches **n** count
19. Variable I should sum with 1 and store in handle pointer(pb)
20. Initialize **i**
21. Get the value for the array **b** from the user
22. Increment **i**  with 1
23. Repeat step 20 until it reaches **n** count
24. Display second array
25. Initialize i
26. Get the value for the array from the user
27. Increment **i**  with 1
28. Repeat step 25 until it reaches **n** count
29. Assign mul=0
30. Multiply a and b matrix and result is added to mul
31. Answer stored in mul
32. Display the result
33. end

**Coding:**

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

void main ()

{

        int\* pa ,\*pb , i;

        int mul, n;

        printf  ("enter the number of elements");

        scanf  ("%d, &n");

        pa = (int\*) malloc (n\* sizeof (int));

        pb = (int\*) malloc (n\* sizeof (int));

        for (i = 0; i < n; i++);

        {

                pa[i] = i + 1;

        }

        printf ("the elements of the first array are");

        for (i = 0; i < n; i++)

        {

                printf ("%d", pa[i]);

        }

        for (i = 0 ; i < n; i++);

        {

                pb[i] = i + 1;

        }

        printf ("the elements of the second array are");

        for (i = 0;i < n; i++)

        {

                printf ("%d",pb[i]);

        }

        for (i = 0; i < n; i++)

        {

        mul=0;

        }

        for (i = 0;i < n; i++)

        {

                mul += (pa[i])\*(pb[i]);

        }

         for(i = 0 ;i < n; i++)

        {

                printf ("%d,mul[i]");

        }

        getch ();

}