

Universidad de Guanajuato División de Ingenierías  
Campus Irapuato Salamanca (DICIS)

Algoritmos y estructura de datos  
Carlos Hugo García Capulín

Tarea No. 4 Problemas con arreglos  
Reporte

Jair Chávez Islas  
24/Agosto/2021

# Problema

¿Qué es un arreglo?

Un arreglo es un conjunto de posiciones de datos, donde cada uno tiene el mismo tipo de dato y el mismo nombre.

`float grupo[10];`



Tipo de  
dato



Nombre  
del arr.



Cantida  
de elem.

En este ejemplo se tiene un arreglo de tipo flotante llamado grupo, que contiene 10 datos.

**Programa 7: Sacar el promedio de los números de un vector.**

Lo que queremos lograr con este programa es que con un arreglo podamos ingresar el número de datos que formará el arreglo y posteriormente ingresar el valor de cada dato para al final poder mostrar el promedio de todos los datos ingresados; El promedio es la suma de los datos dividido entre la cantidad de datos, así que, por ejemplo, podemos poner 8 datos que sean: 1,2,3,5,6,7,4 y 9. Entonces,  $1+2+3+5+6+7+4+9 = 37$ .

$37/8 = 4.625$ .

**Programa 8: Sacar el resultado del producto y de la suma de 2 matrices.**

¿Qué es una matriz?

Una matriz es un conjunto de números ordenados en una estructura de filas y columnas, por ejemplo.

$$B = \begin{pmatrix} 1 & 5 & 2.3 \\ -2 & 0 & 0 \\ 3 & -1.1 & 2 \end{pmatrix}$$

---

## Suma de matrices

La suma de matrices no tiene mayor ciencia de lo que nos dice su propio nombre, es la suma lineal de una matriz A y una matriz B para tener como resultado una matriz C.

Hay una regla muy importante que hay que tomar en cuenta a la hora de sumar matrices y es que, para que dos matrices se puedan sumar deben ser de las mismas dimensiones (misma cantidad de filas y renglones). Ya que, de no ser así, tenemos lo siguiente

Matriz A:	Matriz B:
$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$	$\begin{pmatrix} 1 & 2 & \\ 3 & 4 & \\ & & \end{pmatrix}$

Al no ser de las mismas dimensiones, hay números que no pueden sumarse porque quedan solos o vacíos, por decirlo de alguna manera, así que no se puede hacer una suma; Tomando en cuenta esta regla, tenemos el siguiente ejemplo.

Matriz A:	Matriz B:
$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$	$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$
$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} + \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} = \begin{pmatrix} 2 & 4 & 6 \\ 8 & 10 & 12 \\ 14 & 16 & 18 \end{pmatrix}$	

Pude sacar este resultado gracias a una calculadora de matrices en línea.

## Multiplicación de matrices

Para que se pueda dar la multiplicación de matrices también se necesita seguir una regla y es que, el número de filas de la matriz A, debe ser igual al número de columnas de la matriz B, con el siguiente ejemplo tenemos que,

Matriz A:	Matriz B:
$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$	$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$

La columna de la matriz A {1,2,3} se multiplica con la fila de la matriz B {1,4,7} de la siguiente manera  $1*1$ ;  $2*4$ ;  $3*7$ ; los resultados se suman y en la matriz C, se pone el resultado en la casilla en la que coincidan la fila y la columna si es que estuviesen en la misma matriz, en este mismo ejemplo sería  $1+8+21=30$  y se pone en la primera casilla, y así con todos, hasta tener la matriz completa.

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} = \begin{pmatrix} 30 & 36 & 42 \\ 66 & 81 & 96 \\ 102 & 126 & 150 \end{pmatrix}$$

De ser distinto, tenemos el siguiente ejemplo

Matriz A:	Matriz B:
<div><div></div><div>1</div><div>2</div><div>3</div></div>	<div><div></div><div>1</div><div>2</div><div>3</div></div>
<div><div></div><div>4</div><div>5</div><div>6</div></div>	<div><div></div><div>4</div><div>5</div><div>6</div></div>
<div><div></div><div>7</div><div>8</div><div>9</div></div>	<div><div></div><div></div><div></div><div></div></div>

Queda un número sin multiplicarse y es por eso que no se podría llevar a cabo esta multiplicación.

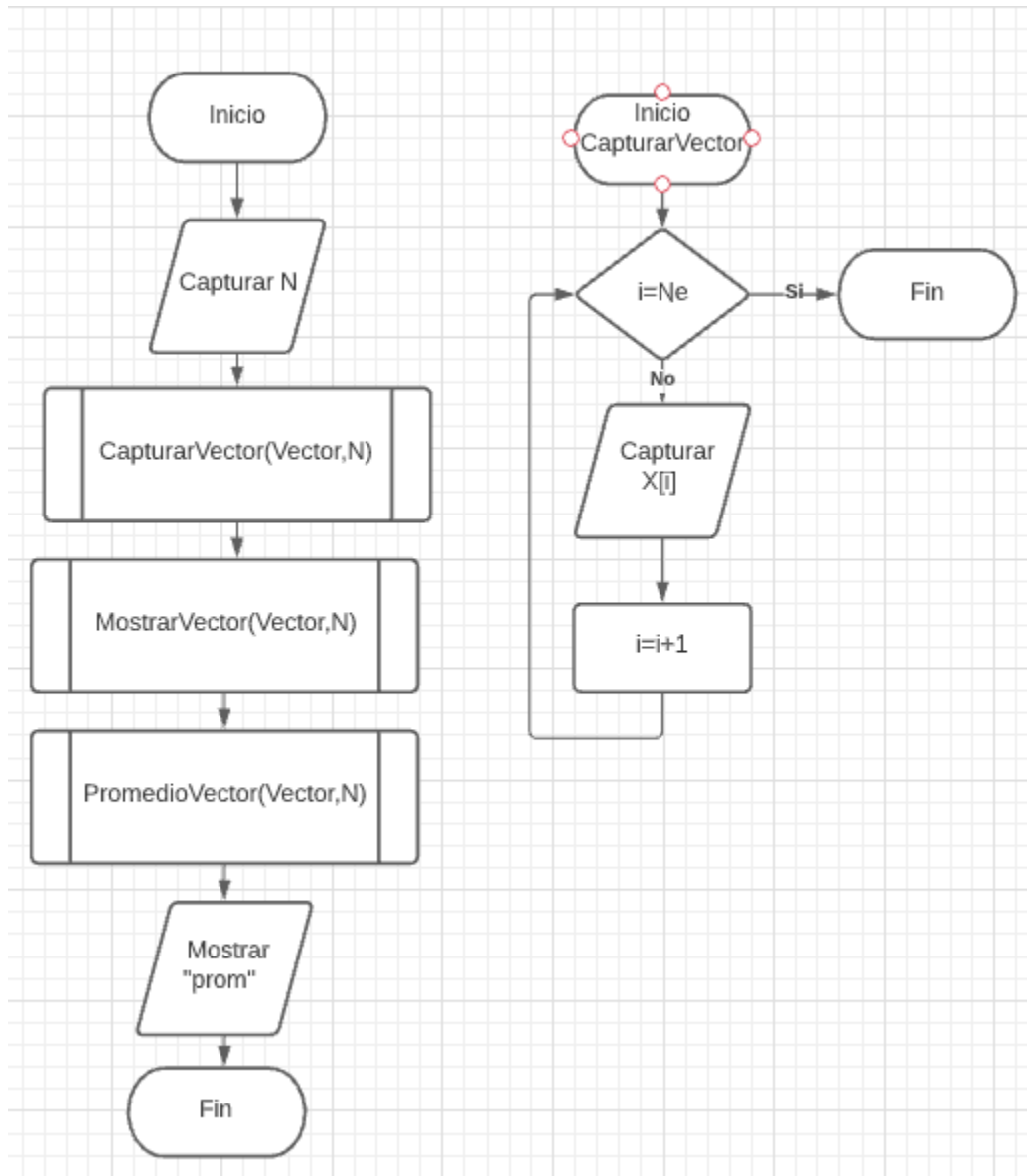
Matriz A:	Matriz B:
<div><div></div><div>1</div><div>2</div><div>3</div></div>	<div><div></div><div>1</div><div>2</div><div></div></div>
<div><div></div><div>4</div><div>5</div><div>6</div></div>	<div><div></div><div>4</div><div>5</div><div></div></div>
<div><div></div><div>7</div><div>8</div><div>9</div></div>	<div><div></div><div>6</div><div>7</div><div></div></div>

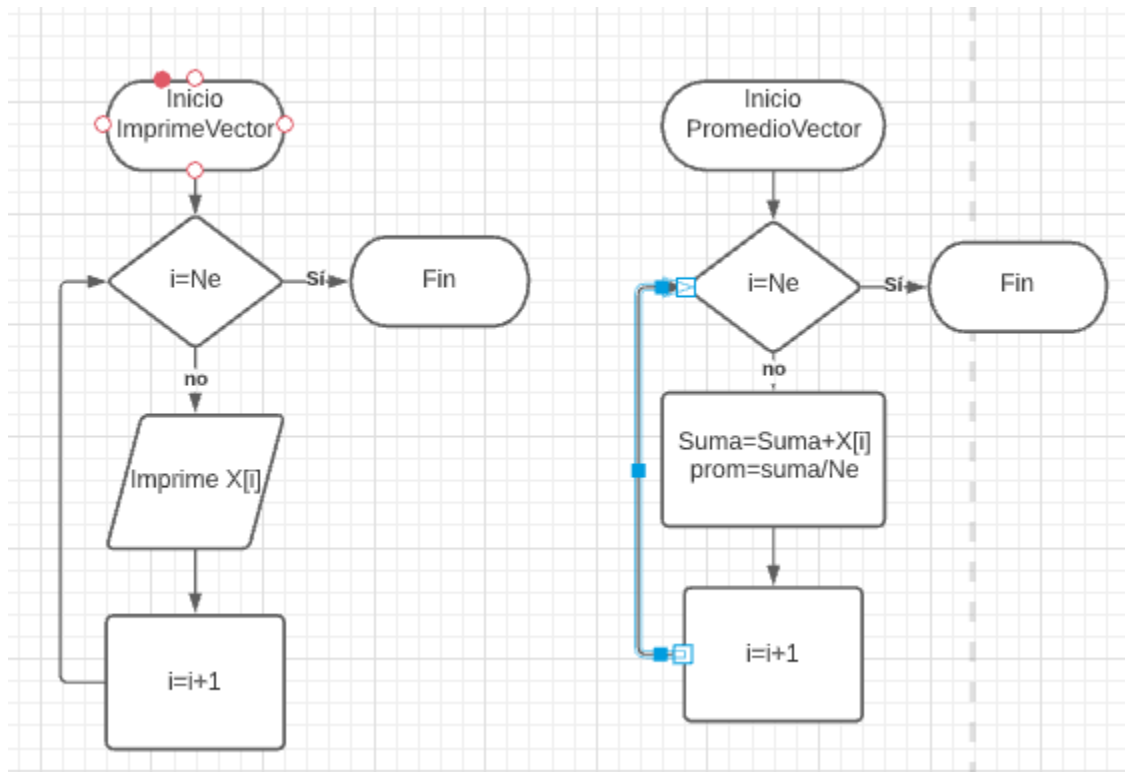
En este ejemplo sí que se puede multiplicar, dando como resultado la siguiente matriz.

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 \\ 4 & 5 \\ 6 & 7 \end{pmatrix} = \begin{pmatrix} 27 & 33 \\ 60 & 75 \\ 93 & 117 \end{pmatrix}$$

# Solución implementada

Diagrama de flujo del primer problema





**Diagrama de flujo del segundo problema**

**Código comentado del primer problema**

```

1 #include<stdio.h>//Agregamos la librería stdio.h que contiene las funciones básicas del lenguaje de programación C.
2
3 float PromedioVector(int X[], unsigned int Ne);//Aquí Se ponen los prototipos de las funciones que vayamos utilizando, cada uno conforme a su respectiva función.
4 void CapturarVector(int X[], unsigned int Ne);
5 void ImprimeVector(int X[], unsigned int Ne);
6
7
8 int main();//Inicia función Principal
9 { int Vector[100]; //Declaración de las variables, en el caso de esta línea de código, tenemos una cadena.
10   unsigned int N;
11   float promedio;
12
13   printf("\nNumero de elementos que va a capturar: "); //pedimos un numero mediante un texto que se muestra al usuario.
14   scanf("%u",&N); //guardamos el numero en la variable ya declarada anteriormente "N".
15
16   CapturarVector(Vector,N); //Aplicamos la función "CapturarVector" al número N y a la cadena Vector.
17   printf("\n El vector capturado es: "); //Aquí imprimimos un breve mensaje al usuario previo de mandar el resultado.
18   ImprimeVector(Vector,N); //Aplicamos la función "ImprimeVector" al número N y a la cadena "Vector".
19
20   promedio=PromedioVector(Vector,N); //Aplicamos la función "PromedioVector" al número N y a la cadena Vector y lo guardamos en la variable promedio.
21
22   printf("\n El promedio es: %f",promedio); //Imprimimos el resultado del promedio.
23   printf("\n"); //Aquí solo se imprime un salto de línea.
24   return 0; //se retorna el valor de 0 para verificar que haya finalizado sin problemas
25 }
26
27 void ImprimeVector(int X[], unsigned int Ne) //Empieza la función "ImprimeVector".
28 { unsigned int i; //Se declaran las variables locales de esta función
29   for(i=0; i<Ne; i++) //en este ciclo el contador va imprimiendo los números, hasta llegar a el número de números que tendrá el vector, digitado anteriormente por el usuario.
30     printf("\nV[%i]-%i",i,X[i]); //Se imprime el número digitado por el usuario con su respectiva posición según el contador.
31
32 }
33
34 void CapturarVector(int X[], unsigned int Ne) //Empieza la función "CapturarVector".
35 { unsigned int i; //Se declara la variable local de esta función.
36
37   for(i=0; i<Ne; i++) //En este ciclo el contador va capturando los números, hasta llegar a el número de números que tendrá el vector, digitado anteriormente por el usuario.
38   {
39     printf("\nX[%i]=","i"); //Se imprime el número del vector, indicando a el usuario que capture el número como tal.
40     scanf("%i",&X[i]); //Se guarda el valor del número.
41   }
42 }
43
44
45 float PromedioVector(int X[], unsigned int Ne) //Empieza la función "PromedioVector".
46 { unsigned int i; //Se declaran las variables locales de esta función.
47   int suma=0;
48   float prom;
49
50   for(i=0; i<Ne; i++) //Se inicia un ciclo que irá sumando los números del vector ya antes digitados.
51     suma+=X[i]; //Se van sumando todos los números del vector y se guarda en la variable suma.
52   prom=suma/(float)Ne; //Se saca el promedio de la suma y se guarda en prom. Hace falta cambiar la variable Ne de int a float.
53   return prom; //Se retorna la variable prom.
54 }
55
56

```

## Código comentado del segundo problema

```

1 #include <stdio.h> //Agregamos la librería stdio.h que contiene las funciones básicas del lenguaje de programación C.
2
3 void CapturaMatriz(int Ma[][10], unsigned int M, unsigned int N); //Aquí Se ponen los prototipos de las funciones que vayamos utilizando, cada uno conforme a su respectiva función.
4 void ImprimeMatriz(int Ma[][10], unsigned int M, unsigned int N);
5 char SumaMatrices(int A[][10], unsigned int Ma, unsigned int Nb, int B[][10], unsigned int Mb, unsigned int Nb, int C[][10], unsigned int Mc, unsigned int Nc);
6 char ProductoMatrices(int A[][10], unsigned int Ma, unsigned int Nb, int B[][10], unsigned int Mb, unsigned int Nb, int C[][10], unsigned int Mc, unsigned int Nc);
7
8
9 int main() //Inicia función Principal
10 { int A[10][10]; //Se declaran las variables.
11   int B[10][10];
12   int C[10][10];
13   unsigned int Ma, Na, Mb, Nb, Mc, Nc;
14   char flag; //flag se utiliza para almacenar un valor binario
15
16   //Para poder imprimir nuestra matriz necesitamos filas y columnas, así que le pedimos al usuario la cantidad de cada una que tendrá la matriz.
17   printf("\nIngrese el numero de filas de la matriz A: ");
18   scanf("%u",&Ma); //El número de filas se guarda en Ma.
19   printf("\nIngrese el numero de columnas de la matriz A: ");
20   scanf("%u",&Na); //El número de columnas se guarda en Na.
21
22   CapturaMatriz(A, Ma, Na); //Se aplica la función "CapturaMatriz" para las variables A, Ma, Na.
23   ImprimeMatriz(A, Ma, Na); //Se aplica la función "ImprimeMatriz" para las variables A, Ma, Na.
24
25   //Para poder hacer sumar o multiplicar matrices, se necesitan al menos 2 matrices, entonces aquí al igual que con la primera, pedimos los datos.
26   printf("\nIngrese el numero de filas de la matriz B: ");
27   scanf("%u",&Mb); //El número de filas se guarda en Mb.
28   printf("\nIngrese el numero de columnas de la matriz B: ");
29   scanf("%u",&Nb); //El número de columnas se guarda en Nb.
30
31   CapturaMatriz(B, Mb, Nb); //Se aplica la función "CapturaMatriz" para las variables B, Mb, Nb.
32   ImprimeMatriz(B, Mb, Nb); //Se aplica la función "ImprimeMatriz" para las variables B, Mb, Nb.
33
34   flag = SumaMatrices(A, Ma, Na, B, Mb, Nb, C, Mc, Nc); //se iguala flag y la función.
35
36   printf("\n"); //Aquí solo se imprime un salto de línea.
37
38   if(flag) //De ser que no haya habido errores en cuanto a la suma, continua con la impresión de la matriz resultante.
39   {
40     Mc=Ma;
41     Nc=Na;
42     printf("\nA+B=");
43     ImprimeMatriz(C, Mc, Nc); //Utiliza la función para imprimir la matriz C.
44   }
45   else //Si hubo algún proble, imprime un mensaje de que no se pudo hacer la operación.
46   {
47     printf("La suma de las matrices no es posible porque no tiene las mismas dimensiones");
48     Mc=Ma;
49     Nc=Nb;
50     printf("\n"); //Aquí solo se imprime un salto de línea.
51   }
52 }
53
54

```

```

51     flag = ProductoMatrices(A,Ma,Na,B,Mb,Nb,C,Mc,Nc);//ahora se iguala flag y la función.
52
53     printf("\n");//Aquí solo se imprime un salto de línea
54     if(flag)//De ser que no haya habido errores en cuanto a la suma, continua con la impresión de la matriz resultante.
55     {
56         printf("\nA*B=C");
57         ImprimeMatriz(C,Mc,Nc);//Utiliza la función para imprimir la matriz C.
58     }
59     else//Si hubo algún proble, imprime un mensaje de que no se pudo hacer la operación.
60     {
61         printf("El producto de las matrices no es posible");
62         printf("\n");//Aquí solo se imprime un salto de línea
63     }
64     return 0;
65 }
66
67 char ProductoMatrices(int A[][10], unsigned int Ma, unsigned int Na, int B[][10], unsigned int Mb, unsigned int Nb, int C[][10], unsigned int Mc, unsigned int Nc)//Se inicia la función "ProductoMatrices".
68 {
69     unsigned int i,j,k;//Se declaran las variables locales de la función.
70     if((Ma-Mb)%(Mc-Na)%(Nc-Nb))//Para poder multiplicar 2 matrices, las filas de A debe ser la misma cantidad que las columnas de B, así que se hace la comparación.
71     {
72         for(i=0; i<Mc; i++)
73         {
74             for(j=0; j<Nc; j++)//Se inician 2 ciclos for para poder multiplicar.
75             {
76                 C[i][j] = 0;
77                 for(k=0; k<Na; k++)
78                     C[i][j] += A[i][k]*B[k][j];//Aquí se hace la multiplicación de las matrices
79             }
80             return 1;//De que la comparación sea correcta, retorna un 1.
81         }
82         return 0;//De no ser así, retorna un 0.
83     }
84 }
85
86 char SumaMatrices(int A[][10], unsigned int Ma, unsigned int Na, int B[][10], unsigned int Mb, unsigned int Nb, int C[][10], unsigned int Mc, unsigned int Nc)//Se inicia la función "SumaMatrices".
87 {
88     unsigned int i,j;//Se declaran las variables locales de la función.
89     if((Ma-Mb)%(Na-Nb))//Para poder sumar 2 matrices, las 2 deben tener las mismas dimensiones, así que aquí se hace la comparación, si las filas y columnas son del mismo tamaño.
90     {
91         for(i=0; i<Na; i++)
92         {
93             for(j=0; j<Nb; j++)//Se inician dos ciclos for para poder ir haciendo la suma de los elementos.
94             {
95                 C[i][j]=A[i][j]+B[i][j];//El resultado de la suma se guarda en la matriz C.
96             }
97             return 1;//De que la comparación sea correcta, retorna un 1.
98         }
99         return 0;//De no ser así, retorna un 0.
100     }
101 }
102
103 void ImprimeMatriz(int Ma[][10], unsigned int M, unsigned int N)//Se inicia la función "ImprimeMatriz".
104 {
105     unsigned int i,j;//Se declaran las variables locales de la función.
106     for(i=0; i<M; i++)
107     {
108         printf("\n");//Aquí solo se imprime un salto de línea
109         for(j=0; j<N; j++)//Se inician dos ciclos for uno dentro de otro, para poder imprimir los datos.
110         {
111             printf("%i, ",Ma[i][j]);//Este es el código que imprime el dato como tal, se agrega una coma para separar los datos de la matriz.
112         }
113     }
114 }

```

```

101
102 void CapturaMatriz(int Ma[][10], unsigned int M, unsigned int N)//Se inicia la función "CapturaMatriz".
103 {
104     unsigned int i,j;//Se declaran las variables locales de la función.
105     for(i=0; i<M; i++)
106     {
107         for(j=0; j<N; j++)//Se inician dos ciclos for uno dentro de otro, para poder capturar los datos.
108         {
109             printf("\n[%i][%i]= ",i,j);//Aquí se imprime la posición de la matriz, para que el usuario sepa en que posición de esta se va a guardar el dígito.
110             scanf("%i",&Ma[i][j]);//Se guardan todos los dígitos dentro de la matriz.
111         }
112     }
113 }

```

# Pruebas y resultados

## Evidencia del primer problema

```

C:\Windows\system32\cmd.exe
C:\Users\chama\Documents\algoritmos>p07
Numero de elementos que va a capturar: 8
X[0]=1
X[1]=2
X[2]=3
X[3]=5
X[4]=6
X[5]=7
X[6]=4
X[7]=9
El vector capturado es:
V[0]=1
V[1]=2
V[2]=3
V[3]=5
V[4]=6
V[5]=7
V[6]=4
V[7]=9
El promedio es: 4.625000

```



## Evidencia del segundo problema

```
C:\Users\chama\Documents\algoritmos>p08

Ingrese el numero de filas de la matriz A: 3

Ingrese el numero de columnas de la matriz A: 3

M[0][0]=1
M[0][1]=2
M[0][2]=3
M[1][0]=4
M[1][1]=5
M[1][2]=6
M[2][0]=7
M[2][1]=8
M[2][2]=9

1, 2, 3,
4, 5, 6,
7, 8, 9,
Ingrese el numero de filas de la matriz B: 3

Ingrese el numero de columnas de la matriz B: 3

M[0][0]=1
M[0][1]=2
M[0][2]=3
M[1][0]=4
M[1][1]=5
M[1][2]=6
M[2][0]=7
M[2][1]=8
M[2][2]=9

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

A+B=
2, 4, 6,
8, 10, 12,
14, 16, 18,

A*B=
30, 36, 42,
66, 81, 96,
102, 126, 150,

C:\Users\chama\Documents\algoritmos>
```

```
C:\Users\chama\Documents\algoritmos>p08

Ingrese el numero de filas de la matriz A: 3

Ingrese el numero de columnas de la matriz A: 3

M[0][0]=1
M[0][1]=2
M[0][2]=3
M[1][0]=4
M[1][1]=5
M[1][2]=6
M[2][0]=7
M[2][1]=8
M[2][2]=9

1, 2, 3,
4, 5, 6,
7, 8, 9,
Ingrese el numero de filas de la matriz B: 3

Ingrese el numero de columnas de la matriz B: 2

M[0][0]=1
M[0][1]=2
M[1][0]=4
M[1][1]=5
M[2][0]=6
M[2][1]=7

1, 2,
4, 5,
6, 7,
La suma de las matrices no es posible porque no tiene las mismas dimensiones

A*B=
27, 33,
60, 75,
93, 117,

C:\Users\chama\Documents\algoritmos>
```

### **Problema 1**

Exactamente como lo habíamos dicho en el problema, tomamos un vector de 8 números dándole los siguientes valores {1,2,3,5,6,7,4,9}, el programa hizo la suma y al dividir también dio 4.625

### **Problema 2**

En el problema vimos el resultado que daría si sumamos y multiplicamos dos matrices de 3x3 exactamente iguales y en el programa nos dieron los mismos resultados.

Y también tenemos el segundo ejemplo del segundo problema, en el que, como vemos, el número de columnas de la matriz A, coincide con el número de filas de la matriz B, entonces si se pueden multiplicar, pero no como no tienen exactamente las mismas dimensiones, no se puede sumar.

**Conclusión:** No es la primera vez que tomo algún curso sobre arreglos, pero siempre me había quedado con la duda de exactamente cómo es que funcionan, en este reporte pude reforzar lo poco que ya sabía y comprender lo que antes no me había quedado del todo claro; los arreglos son muy importantes a la hora de programar, se llegan a utilizar muchas veces y ahora sé cómo aplicarlo.