

Practica de Vectores

Desarrolle un programa que sume dos vectores de enteros y posteriormente calcule el máximo, mínimo y promedio del vector resultante.

Pasos:

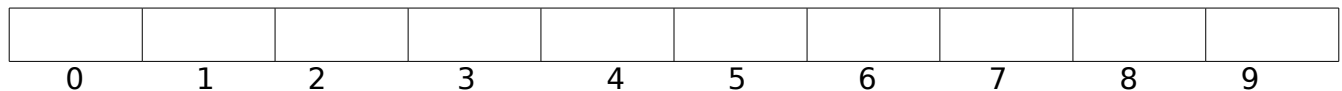
1. Leer el vector 1 (A)
2. Leer el vector 2 (B)
3. Sumar el vector 1 y 2 $\rightarrow C = A + B$
4. Calcular el máximo, mínimo y promedio del vector suma (C)

Solución por pasos

Paso 1: Leer el vector 1.

Inicialmente debemos decidir la dimensión de nuestro vector, asumamos 10 posiciones (pueden utilizar las que consideren convenientes). Luego designamos un nombre o identificador valido para nuestra variable del tipo arreglo unidimensional (A).

A



Recordando que las posiciones del vector se manejan desde 0 hasta N-1, tendremos un vector A cuyas posiciones varían entre 0 y 9, para un total de 10 elementos.

Si deseamos leer la posición 3 del vector A:

```
scanf("%d",&A[3]);
```

La misma instrucción se repite para la lectura de las distintas posiciones, por tanto si deseamos leer todas las posiciones (0-9) solo debemos utilizar una estructura de repetición cuyo contador varíe en este rango.

```
for (i=0;i<10;i++)
```

```
    scanf("%d",&A[i]);
```

lo cual es equivalente a decir:

```
for (i=0;i<=9;i++)
```

```
scanf("%d",&A[i]);
```

El programa completo hasta este punto quedaria:

```
#include<stdio.h> // definición de librería
#define N 10      // utilizamos una constante para definir la longitud del vector
int main(){
    int A[N],i;   // definimos el vector A de N elementos y el contador i
    for (i=0;i<N;i++)
        scanf("%d",&A[i]);
    return 0;
}
```

Si compilamos y ejecutamos este código, no observaremos mensaje alguno pero la computadora esperara que introduzcamos 10 números enteros. Si deseamos insertar algun mensaje el programa quedaria asi:

```
#include<stdio.h> // definición de librería
#define N 10      // utilizamos una constante para definir la longitud del vector
int main(){
    int A[N],i;   // definimos el vector A de N elementos y el contador i
    for (i=0;i<N;i++){
        printf("A[%d]: "); // este mensaje es variable y puede elegir uno de su gusto
        scanf("%d",&A[i]);
    }
    return 0;
}
```

Paso 2: Leer el vector 2 (B) . Es equivalente al algoritmo anterior solo que ahora leeremos el arreglo B.

```
#include<stdio.h> // definición de librería
#define N 10      // utilizamos una constante para definir la longitud del vector
```

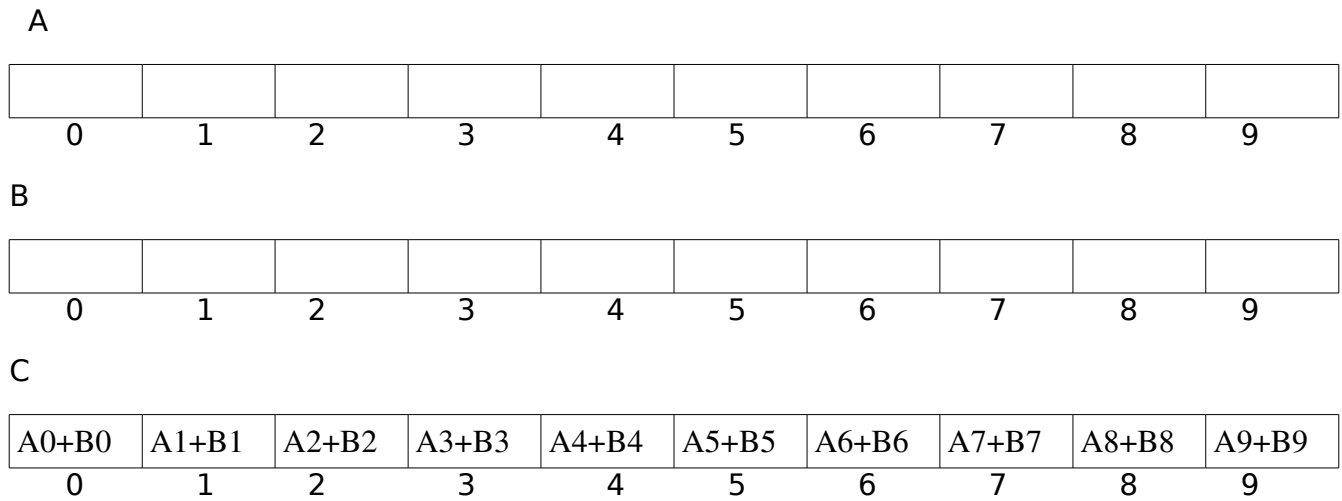
```

int main(){
    int A[N],i,B[N];    // definimos el vector A de N elementos y el contador i
    for (i=0;i<N;i++){
        printf("A[%d]: "); // este mensaje es variable y puede elegir uno de su gusto
        scanf("%d",&A[i]);
    }
    // lectura del vector B
    for (i=0;i<N;i++){
        printf("B[%d]: "); // este mensaje es variable y puede elegir uno de su gusto
        scanf("%d",&B[i]);
    }
    return 0;
}

```

Paso 3: Sumar el vector 1 y 2 → C= A + B

La suma de vectores se realiza posición a posición,



Si observamos la figura anterior,

$$\left. \begin{array}{l}
 C_0 = A_0 + B_0 \\
 C_1 = A_1 + B_1 \\
 \dots \\
 C_9 = A_9 + B_9
 \end{array} \right\} C_i = A_i + B_i$$

Interesa realizar una estructura de repetición que varíe de 0 a 9, efectuando la operación anterior

```
for (i=0;i<N;i++)
```

```
    C[i]=A[i]+B[i];
```

Y si lo deseamos inmediatamente podemos mostrar el resultado de la operación

```
for (i=0;i<N;i++){
```

```
    C[i]=A[i]+B[i];
```

```
    printf("%d\t",C[i]); // utilizamos la tabulación para que los valores quede separados
```

```
}
```

El programa iría así

```
#include<stdio.h> // definición de librería
```

```
#define N 10 // utilizamos una constante para definir la longitud del vector
```

```
int main(){
```

```
    int A[N],i,B[N],C[N]; // definimos el vector C de N elementos
```

```
    for (i=0;i<N;i++){
```

```
        printf("A[%d]: "); // este mensaje es variable y puede elegir uno de su gusto
```

```
        scanf("%d",&A[i]);
```

```
    }
```

```
    // lectura del vector B
```

```
    for (i=0;i<N;i++){
```

```
        printf("B[%d]: "); // este mensaje es variable y puede elegir uno de su gusto
```

```
        scanf("%d",&B[i]);
```

```
    }
```

```
    for (i=0;i<N;i++){
```

```
        C[i]=A[i]+B[i];
```

```
        printf("%d\t",C[i]);
```

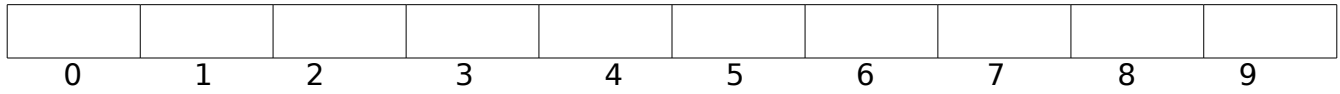
```
    }
```

```
    return 0;
```

```
}
```

Paso 4: Calcular el máximo, mínimo y promedio del vector suma (C)

C



Para calcular el máximo, podemos inicialmente asumir que el máximo elemento se encuentra en la posición 0 del vector C, y a partir de allí recorrer nuestro vector preguntandonos : “Es la posición X mayor que nuestro máximo inicial?” De ser cierto, estaríamos ante un nuevo máximo y deberíamos cambiar el máximo, de lo contrario no efectuamos acción alguna.

```
maximo=C[0]; // el máximo inicialmente es la posición 0 del vector C
for (i=1;i<N;i++) // Recorremos el vector, desde la posición 1
    if (C[i]>maximo) // Es la posición X mayor que nuestro máximo inicial?
        maximo=C[i]; // cambiamos el máximo, ahora es C[i]
```

Notese que no es necesario el uso de llaves puesto que solo se tiene una sentencia y una estructura.

El procedimiento para encontrar el mínimo es equivalente.

inicialmente asumimos que el minimo elemento se encuentra en la posición 0 del vector C, y a partir de allí recorrer nuestro vector preguntandonos : “Es la posición X menor que nuestro menor inicial?” De ser cierto, estaríamos ante un nuevo minimo y deberíamos cambiar el minimo, de lo contrario no efectuamos acción alguna.

```
minimo=C[0]; // el minimo inicialmente es la posición 0 del vector C
for (i=1;i<N;i++) // Recorremos el vector, desde la posición 1
    if (C[i]<minimo) // Es la posición X menor que nuestro minimo inicial?
        minimo=C[i]; // cambiamos el minimo, ahora es C[i]
```

El programa quedaria así hasta el momento

```
#include<stdio.h> // definición de librería
#define N 10 // utilizamos una constante para definir la longitud del vector
int main(){
    int A[N],i,B[N],C[N],maximo,minimo;
    for (i=0;i<N;i++){
```

```

        printf("A[%d]: "); // este mensaje es variable y puede elegir uno de su gusto
        scanf("%d",&A[i]);
    }
    // lectura del vector B
    for (i=0;i<N;i++){
        printf("B[%d]: "); // este mensaje es variable y puede elegir uno de su gusto
        scanf("%d",&B[i]);
    }
    for (i=0;i<N;i++){
        C[i]=A[i]+B[i];
        printf("%d\t",C[i]);
    }
    maximo=C[0]; // el máximo inicialmente es la posición 0 del vector C
    for (i=1;i<N;i++) // Recorremos el vector, desde la posición 1
        if (C[i]>maximo) // Es la posición X mayor que nuestro máximo inicial?
            maximo=C[i]; // cambiamos el máximo, ahora es C[i]
    minimo=C[0]; // el minimo inicialmente es la posición 0 del vector C
    for (i=1;i<N;i++) // Recorremos el vector, desde la posición 1
        if (C[i]<minimo) // Es la posición X menor que nuestro minimo inicial?
            minimo=C[i]; // cambiamos el minimo, ahora es C[i]
    return 0;
}

```

Tanto para encontrar el máximo como el minimo, se recorre el vector C de la misma manera, por lo tanto podemos agrupar las estructuras en una sola y realizar ambas comparaciones al mismo momento.

```

#include<stdio.h> // definición de librería
#define N 10 // utilizamos una constante para definir la longitud del vector
int main(){
    int A[N],i,B[N],C[N],maximo,minimo;
    for (i=0;i<N;i++){
        printf("A[%d]: "); // este mensaje es variable y puede elegir uno de su gusto
    }
}

```

```

        scanf("%d",&A[i]);
    }
    // lectura del vector B
    for (i=0;i<N;i++){
        printf("B[%d]: "); // este mensaje es variable y puede elegir uno de su gusto
        scanf("%d",&B[i]);
    }
    for (i=0;i<N;i++){
        C[i]=A[i]+B[i];
        printf("%d\t",C[i]);
    }
    minimo=C[0]; // el minimo inicialmente es la posición 0 del vector C
    maximo=C[0]; // el máximo inicialmente es la posición 0 del vector C
    for (i=1;i<N;i++){ // Recorremos el vector, desde la posición 1
        if (C[i]>maximo) // Es la posición X mayor que nuestro máximo inicial?
            maximo=C[i]; // cambiamos el máximo, ahora es C[i]
        if (C[i]<minimo) // Es la posición X menor que nuestro minimo inicial?
            minimo=C[i]; // cambiamos el minimo, ahora es C[i]
    }
    return 0;
}

```

Solo falta el promedio, es decir realizar la sumatoria de los elementos del vector y luego dividir entre el número de elementos, para ello necesitamos un acumulador (ac)

Inicializamos el acumulador en 0 y recorremos el vector acumulando cada posición en el acumulador (ac)

```

ac=0;
for (i=0;i<N;i++)
    ac=ac+C[i];
printf("Promedio: %f",ac/10.0);

```

O si lo deseamos podemos aprovechar la estructura de repetición utilizada para el máximo y mínimo. En esta solo recorreremos el vector desde 1-9, por lo tanto el acumulador debe inicializarse con el valor del vector en la posición 0, que es la posición que no recorreríamos.

```
ac=C[0];
for (i=1;i<N;i++)
    ac=ac+C[i];
printf("Promedio: %f",ac/10.0);
```

El programa completo quedaria finalmente como:

```
#include<stdio.h> // definición de librería
#define N 10      // utilizamos una constante para definir la longitud del vector
int main(){
    int A[N],i,B[N],C[N],maximo,minimo,ac;
    for (i=0;i<N;i++){
        printf("A[%d]: "); // este mensaje es variable y puede elegir uno de su gusto
        scanf("%d",&A[i]);
    }
    // lectura del vector B
    for (i=0;i<N;i++){
        printf("B[%d]: "); // este mensaje es variable y puede elegir uno de su gusto
        scanf("%d",&B[i]);
    }
    for (i=0;i<N;i++){
        C[i]=A[i]+B[i];
        printf("%d\t",C[i]);
    }
    ac=C[0];
    minimo=C[0]; // el minimo inicialmente es la posición 0 del vector C
```



```
maximo=C[0]; // el máximo inicialmente es la posición 0 del vector C
for (i=1;i<N;i++){ // Recorremos el vector, desde la posición 1
    if (C[i]>maximo) // Es la posición X mayor que nuestro máximo inicial?
        maximo=C[i]; // cambiamos el máximo, ahora es C[i]
    if (C[i]<minimo) // Es la posición X menor que nuestro minimo inicial?
        minimo=C[i]; // cambiamos el minimo, ahora es C[i]
    ac=ac+C[i];
}
//Damos salida a los datos
printf("\nMaximo: %d",maximo);
printf("\nMinimo: %d",minimo);
printf("\nPromedio: %f",ac/10.0);
return 0;
}
```