

Arreglos y Punteros

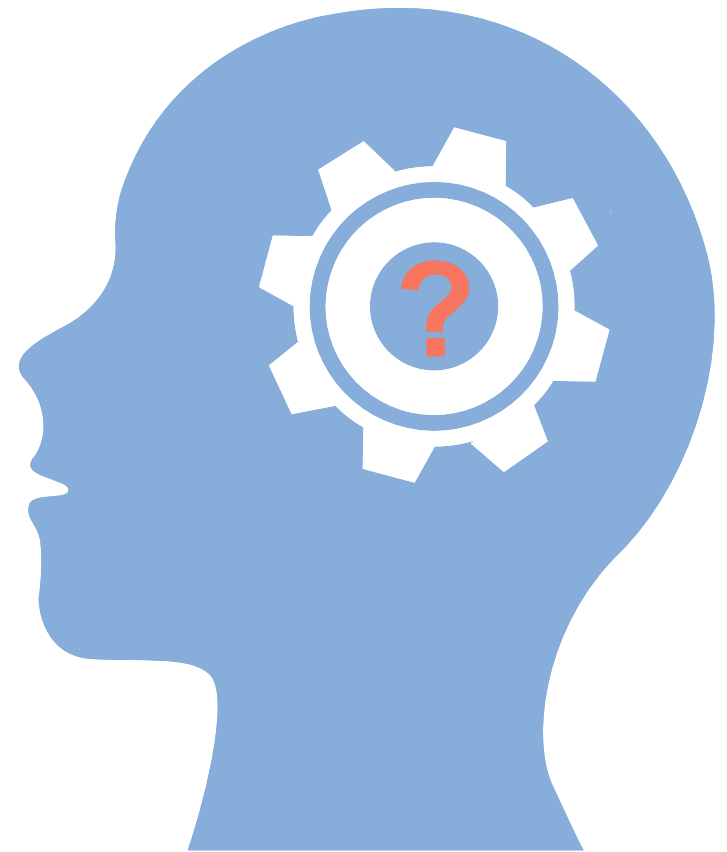
Taller de Lenguajes I
Facultad de Ingeniería – Facultad de Informática
Universidad Nacional de La Plata

Ejercicio a RESOLVER

Escriba un programa que calcule la suma de los elementos de un arreglo de 10 números enteros(int).

El arreglo debe ser inicializado con valores ingresados por teclado.

Modularice la inicialización y el cálculo de la suma.





Prototipo de funciones y programa principal

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #define N 10
4
5  void inicializar (int vector []);
6  int suma (int vector []);
7
8  int main () {
9      int numeros[N];
10     inicializar(numeros);
11     printf("La suma de los elementos del vector es: %d\n", suma(numeros));
12     return 0;
13 }
```

Prototipo de funciones y programa principal



```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #define N 10
4
5  void inicializar (int vector []);
6  int suma (int vector []);
7
8  int main () {
9      int numeros[N];
10     inicializar(numeros);
11     printf("La suma de los elementos del vector es: %d\n", suma(numeros));
12     return 0;
13 }
```

Implementación de funciones



```
15 void inicializar(int vector []) {
16     int i;
17     for (i=0; i<N; i++)
18         scanf("%d", &vector[i]);
19 }
20
21 int suma (int vector []) {
22     int i, suma = 0;
23     for (i=0; i< N; i++)
24         suma += vector[i];
25
26     return suma;
27 }
```

Observe los encabezados de las funciones *inicializar* y *suma...*



```
5 void inicializar (int vector []);  
6 int suma (int vector []);  
7
```

¿ Puede escribirse el parámetro vector de otra manera que resulte equivalente ?

Observe los encabezados de las funciones *inicializar* y *suma*...



```
5 void inicializar (int vector []);  
6 int suma (int vector []);
```

¿ Puede escribirse el parámetro vector de otra manera que resulte equivalente ?




```
5 void inicializar (int * vector);  
6 int suma (int * vector);
```

¿ Es necesario hacer algún cambio en los llamados a las funciones como agregar el operador & ? ¿ Por qué?



```
8 int main () {  
9     int numeros[N];  
10    inicializar(numeros);  
11    printf("La suma de los elementos del vector es: %d\n", suma(numeros));  
12    return 0;  
13 }
```



Observe los encabezados de las funciones *inicializar* y *suma*...



```
5 void inicializar (int vector []);  
6 int suma (int vector []);
```

¿ Puede escribirse el parámetro vector de otra manera que resulte equivalente ?



```
5 void inicializar (int * vector);  
6 int suma (int * vector);
```

¿ Es necesario hacer algún cambio en los llamados a las funciones como agregar el operador & ? ¿ Por qué?



```
8 int main () {  
9     int numeros[N];  
10    inicializar(numeros);  
11    printf("La suma de los elementos del vector es: %d\n", suma(numeros));  
12    return 0;  
13 }
```



No hace falta el &. recordar que el nombre de un arreglo es un PUNTERO CONSTANTE al primer elemento del mismo.

Observe la función *suma*...



```
21 int suma (int * vector) {  
22     int i, suma = 0;  
23     for (i=0; i< N; i++)  
24         suma += vector[i];  
25  
26     return suma;  
27 }
```

¿ Cómo se accedería a los elementos de *vector* empleando notación de punteros?

Observe la función *suma*...



```
21 int suma (int * vector) {  
22     int i, suma = 0;  
23     for (i=0; i< N; i++)  
24         suma += vector[i];  
25  
26     return suma;  
27 }
```

¿ Cómo se accedería a los elementos de *vector* empleando notación de punteros?



```
21 int suma (int * vector) {  
22     int i, suma = 0;  
23     for (i=0; i< N; i++)  
24         suma += *(vector+i);  
25  
26     return suma;  
27 }
```

¿Se puede reescribir la línea 24 sin utilizar la variable *i* ?



```
21 int suma (int * vector) {  
22     int i, suma = 0;  
23     for (i=0; i< N; i++)  
24         suma += *(vector+i);  
25  
26     return suma;  
27 }
```

¿Se puede reescribir la línea 24 sin utilizar la variable *i* ?



```
21 int suma (int * vector) {  
22     int i, suma = 0;  
23     for (i=0; i< N; i++)  
24         suma += *(vector+i);  
25  
26     return suma;  
27 }
```

SI! → Utilizando aritmética de punteros



```
21 int suma (int * vector) {  
22     int i, suma = 0;  
23     for (i=0; i< N; i++)  
24         suma += *vector++;  
25  
26     return suma;  
27 }
```

Notar el orden de
Precedencia entre
los operadores
* y ++

Observe la implementación de la función *inicializar* ...



```
15 void inicializar(int *vector) {  
16     int i;  
17     for (i=0; i<N; i++)  
18         scanf("%d", &(vector[i]));  
19 }
```

¿Cómo accedería a los elementos de vector para asignarlos empleando notación de punteros?

Observe la implementación de la función *inicializar* ...



```
15 void inicializar(int *vector) {  
16     int i;  
17     for (i=0; i<N; i++)  
18         scanf("%d", &(vector[i]));  
19 }
```

¿Cómo accedería a los elementos de vector para asignarlos empleando notación de punteros?



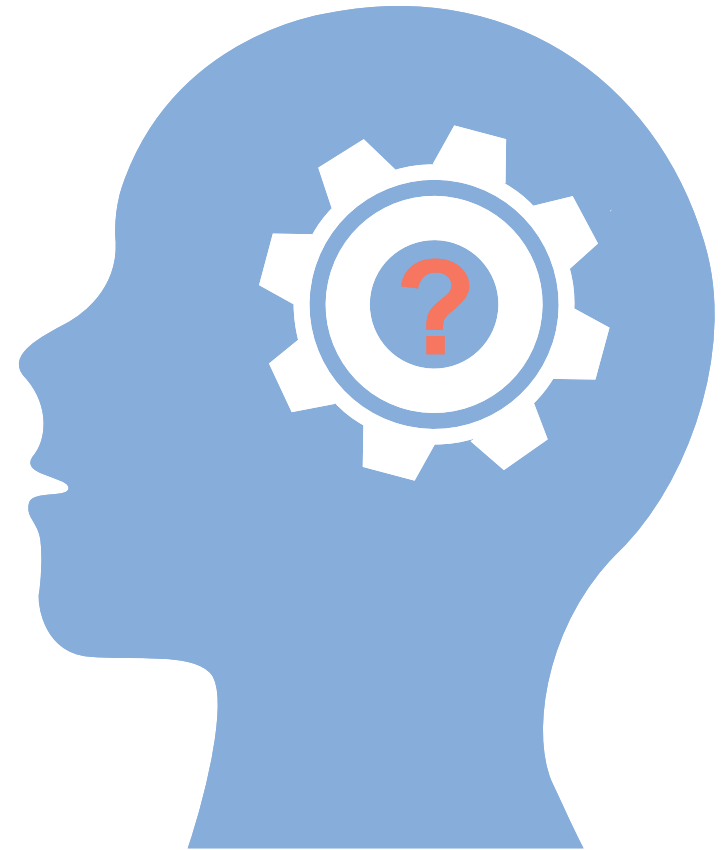
```
15 void inicializar(int *vector) {  
16     int i;  
17     for (i=0; i<N; i++)  
18         scanf("%d", (vector+i));  
19 }
```

Ejercicio

COMPLEMENTARIO

Extienda la función suma para que, además de la suma de los números del vector, calcule el máximo de ellos. Imprima ambos valores en el programa principal

¿Qué se debe cambiar en la función suma? Tener en cuenta que ahora debe retornar 2 valores (la suma y el máximo).



Prototipo de funciones y programa principal



```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <limits.h>
4  #define N 10
5
6  void inicializar (int * vector);
7  void suma_y_maximo (int * vector, int * sum, int * max);
8
9  int main () {
10     int numeros[N], suma, maximo;
11     inicializar(numeros);
12     suma_y_maximo(numeros,&suma,&maximo);
13     printf("La suma de los elementos del vector es: %d\n",suma);
14     printf("El máximo de los elementos del vector es: %d\n",maximo);
15     return 0;
16 }
```

En la línea 12:
¿Por qué las
variables suma y
Máximo llevan "&"
y la variable
numeros no?

Implementación de funciones



```
18 void inicializar(int * vector) {  
19     int i;  
20     for (i=0; i<N; i++)  
21         scanf("%d", &(vector[i]));  
22 }  
23  
24 void suma_y_maximo(int * vector, int * sum, int * max) {  
25     int i;  
26     *sum = 0;  
27     *max = INT_MIN;  
28     for (i=0; i< N; i++) {  
29         *sum += vector[i];  
30         if (vector[i] > *max) *max = vector[i];  
31     }  
32 }
```

¿Por qué es necesario utilizar las variables sum y max con un * delante?