



Facultad de Ingeniería Ingeniería y de Sistemas

Algoritmia y Estructura de Datos: Arreglos

Problema base

Problema

Escribir un algoritmo que lea n números enteros positivos, determine el promedio y cuantos valores son menores al promedio.

¿Qué se requiere para resolverlo?

La estrategia de solución requiere que los valores ingresados sean almacenados para poder recuperarlos luego del calcular el promedio y determinar cuántos cumplen la condición dada.

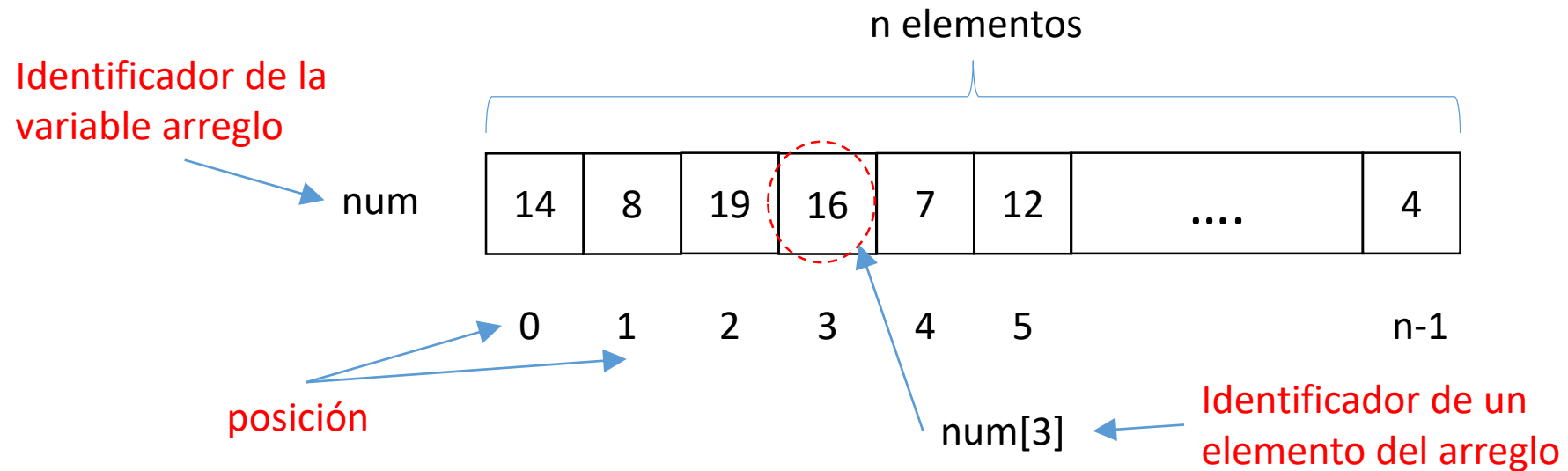
Observaciones

Construir un número entero para almacenar los valores, resulta impráctico por dos razones:

Primera, los numeros pueden ser de diferente cantidad de cifras, lo que implica una complicación adicional para recuperarlos

Segunda, la cantidad de números puede ser muy grande y desbordaría el tamaño de un número entero

¿Qué es un arreglo?



Es un conjunto finito y organizado de datos de un mismo tipo.

Finito, significa que hay una cantidad establecida de valores. El arreglo tiene un máximo número de elementos.

Organizado, significa que cada valor o elemento tiene un lugar establecido en el arreglo.

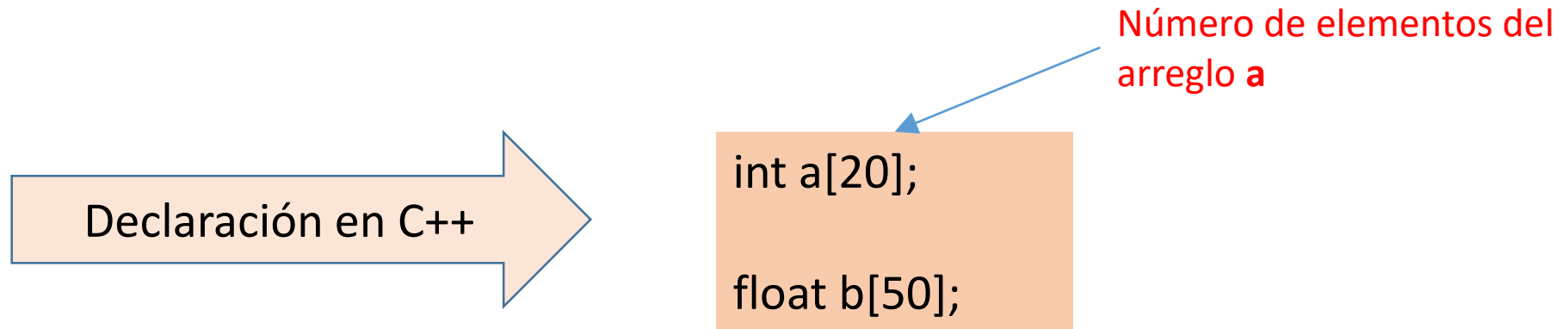
Observación

Todo conjunto organizado de datos es una estructura de datos.

Un arreglo es una estructura de datos

Declaración de variable arreglo en C++

Sea **a** una variable arreglo de 20 números enteros y **b** una variable arreglo de 50 números reales



Asignación, lectura y escritura de variables arreglo en C++

En las operaciones de asignación, lectura y escritura cada elemento del arreglo se maneja independientemente.

declaración

```
int a[20],i;  
float b[50],x;
```

lectura

```
cin>>a[i];  
cin>>b[i];
```

asignación

```
a[5] = 3 ;  
i = a[5] + 1;  
b[18] = b[17];  
b[i] = b[i]+2;  
x = b[i];
```

escritura

```
cout<<a[i] ;  
cout<<b[i];
```

Asignación, lectura y escritura de elementos sucesivos de un arreglo en C++

Cuando se requiere asignar, leer o escribir elementos sucesivos de un arreglo se utilizan bucles

declaración

```
int a[20],i;  
float b[50],x;
```

lectura

```
for(i = 0 ; i < 50; i++)  
cin>>b[i];
```

Lee valores para los 50 elementos del arreglo **b**, desde b[0] hasta b[49]

asignación

```
for(i = 0 ; i < 20; i++)  
a[i] = 0;
```

Pone en cero a los 20 elementos del arreglo **a**, desde a[0] hasta a[19]

escritura

```
for(i = 0 ; i < 20; i++)  
cout>>a[i]>>endl;
```

Escribe los 20 valores del arreglo **a**, desde a[0] hasta a[19]

Solución con arreglo del problema base

Problema

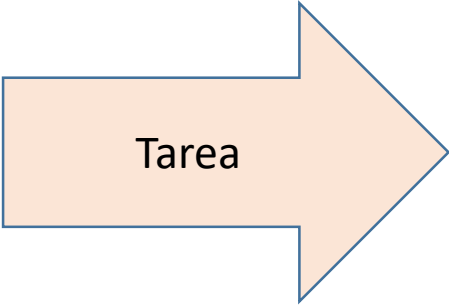
Escribir un algoritmo que lea n notas, determine el promedio y cuántas notas son menores que el promedio.

Solución

[*] 10-notasmenorpromarr.cpp

```
1 // Algoritmo para determinar el promedio de n notas
2 // y determinar cuantas son menores que el promedio
3 #include<iostream>
4 using namespace std;
5 main(){
6     int n,num[100],i, cont=0;
7     float suma=0, prom;
8     do{
9         cout<<"Ingrese la cantidad de notas ";
10        cin>>n;
11    }while(n<=0);
12    for(i=0;i<n;i++){
13        do{
14            cout<<"Ingrese nota ";
15            cin>>num[i];
16        }while(num[i]<=0 || num[i]>20);
17        suma = suma + num[i];
18    }
19    prom = suma/n;
20    cout<<"El promedio de las notas ingresados es "<<prom<<endl;
21    for(i=0;i<n;i++){
22        if (num[i]<prom)
23            cont=cont + 1;
24    }
25    cout<<"hay "<<cont<<" nota(s) menor(es) que el promedio";
26 }
```

Arreglos



Tarea

Desarrollar un algoritmo que lea las n notas de una práctica calificada de algoritmia, y determine cuántas veces ocurre cada nota. Debe mostrar solo las que ocurren.

Ejemplo: 15 notas

17 13 8 15 5 12 7 13 10 5 15 12 13 15 8

Respuesta:

5 ocurre 2 veces

7 ocurre 1 vez

8 ocurre 2 veces

10 ocurre 1 vez

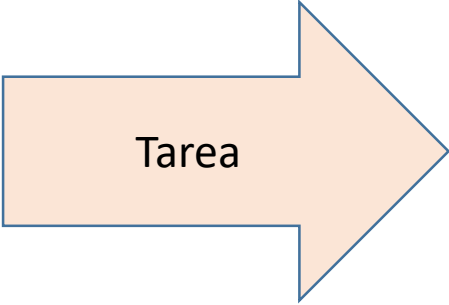
12 ocurre 2 veces

13 ocurre 3 veces

15 ocurre 3 veces

17 ocurre 1 vez

Arreglos



Tarea

Desarrollar un algoritmo que lea un número entero positivo y determine cuántos dígitos diferentes tiene y cuántas veces ocurre cada uno

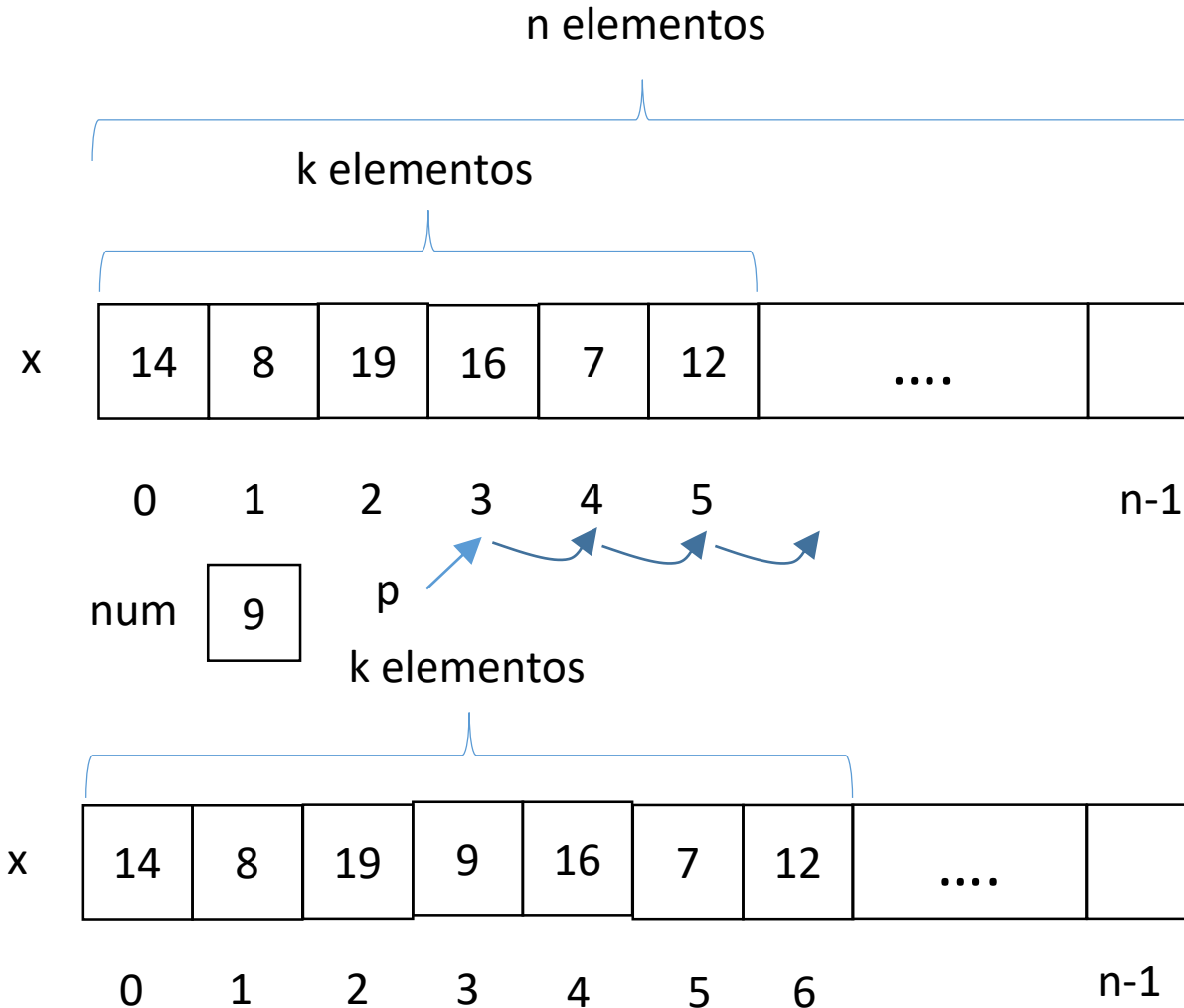
Ejemplo: 123456346

6 digitos diferentes

1 ocurre 1 vez
2 ocurre 1 vez
3 ocurre 2 veces
4 ocurre 2 veces
5 ocurre 1 vez
6 ocurre 2 veces

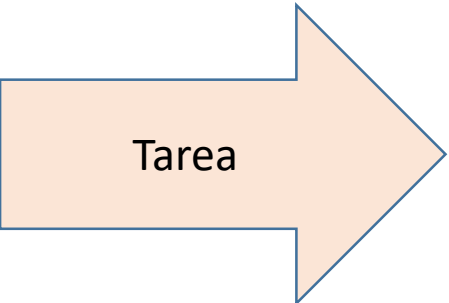
Inserción de un elemento en un arreglo

Se tiene un arreglo x de n elementos con k elementos utilizados, donde $k < n$, y se requiere insertar un valor num en la posición p , donde $p < k$



```
for (i=k; i>p ;i--)  
    x[i] = x[i-1];  
x[p] = num;  
k = k + 1;
```

Inserción de un elemento en un arreglo

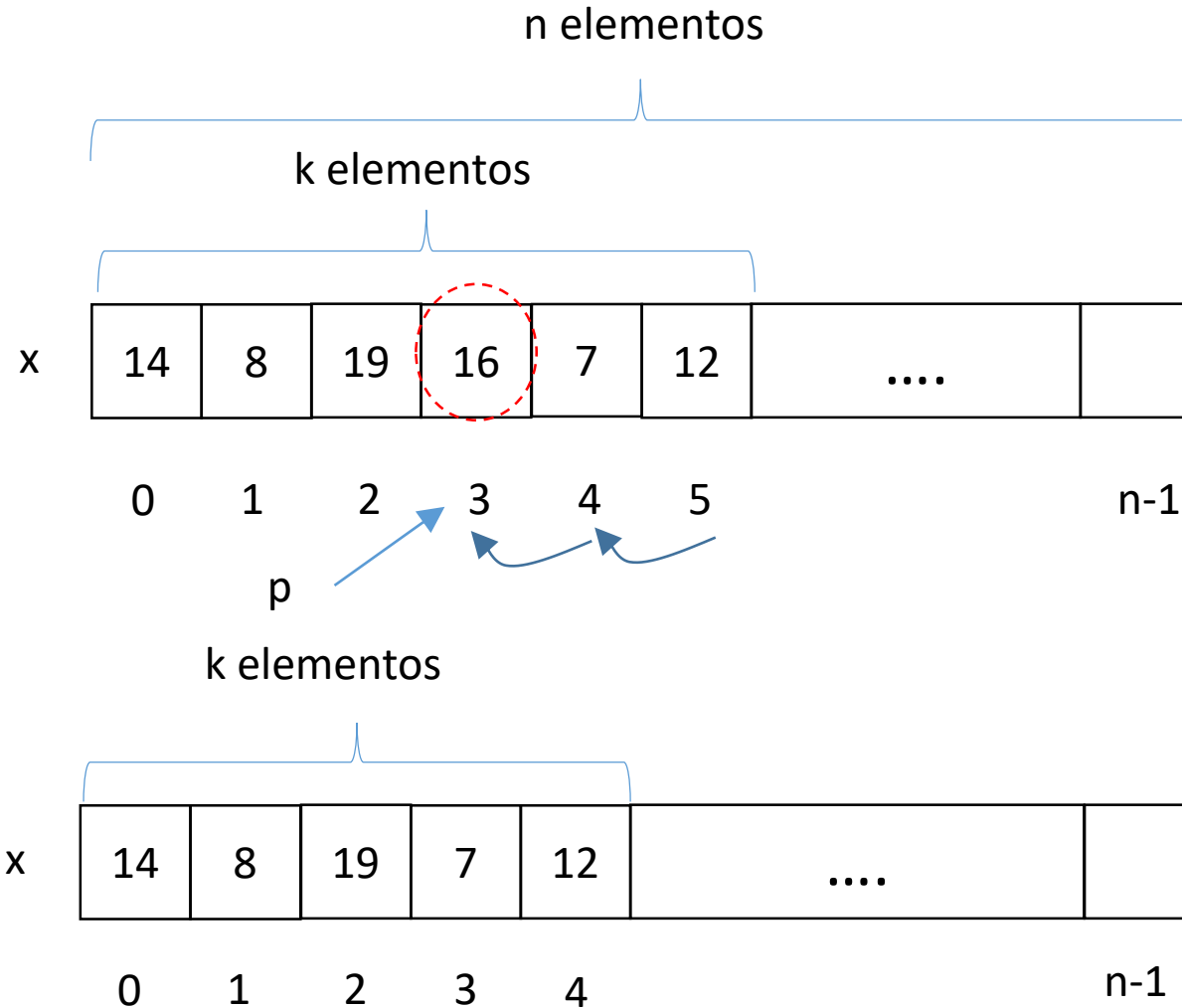


Tarea

Desarrollar un algoritmo que lea k notas, ordenadas de menor a mayor en un arreglo. Luego lea una nueva nota y la coloque en el arreglo, de forma tal, que siga ordenado.

Eliminación de un elemento en un arreglo

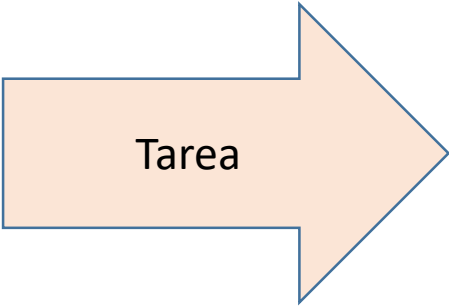
Se tiene un arreglo x de n elementos con k elementos utilizados, donde $k < n$, y se requiere eliminar el elemento de posición p , donde $p < k$



```
for (i=p; i<k-1; i++)  
    x[i] = x[i+1];  
k = k - 1;
```



Eliminación de elementos de un arreglo



Tarea

Desarrollar un algoritmo que lea k notas en un arreglo y luego elimine las notas menores que el promedio