



## Practica N°5(07-10-2015)

### Algoritmos y Lenguajes de Programación



**Importante:** Para desarrollar los ejercicios en esta guía es necesario usar las funciones **malloc** y **calloc** de la librería **stdlib.h**

#### Problema 1:

Haga un programa al que se le ingrese un número definido de notas (guardadas en un arreglo), para luego calcular la nota máxima, nota mínima y promedio, con tres funciones del mismo nombre.

Debe también calcular si es que el usuario ha aprobado o no.

El resultado debe ser el siguiente:

```
Ingrese numero de notas:
6
Ingrese sus 6 notas:
5.6
3.9
6.1
4.0
4
5.8
Su nota maxima es 6.100000
Su nota minima es 3.900000
Su promedio es 4.900000, ha aprobado.
Process returned 0 (0x0)   execution time : 129.201 s
Press any key to continue.
```

**Pistas y tips:** Puesto que las notas son ingresadas y guardadas en el `main()`, para acceder a ellas desde las funciones, debe usar punteros. Recuerde que la dirección del primer elemento de un arreglo es simplemente el nombre de ese arreglo (considerando `"int notas[5];"`, la dirección del primer elemento es `"notas"`, la del el segundo es `"(notas+1)"`, y así en adelante ).

#### Problema 2:

Dado las variables:

`char s[8]="puntero";`

`int v[5]={1,2,3,4,5};`

**a)** Escriba una función que modifique mediante punteros:

- La palabra "puntero" y la deje como "p\*ntero".

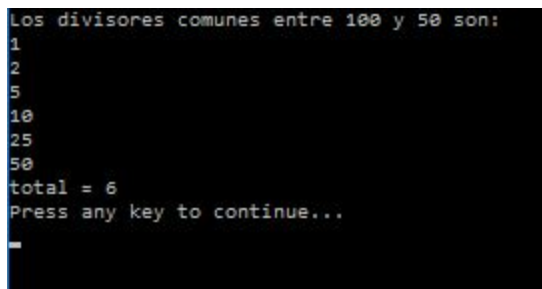
- El arreglo `v[5]={1,2,3,4,5}` y lo deje como `v[5]={1,2,0,4,5}`. Imprima los valores de ambas variables antes y después de la modificación.

**b)** Escriba otra función en el programa anterior para que haga lo siguiente:

- Pase la palabra “puntero” de minúsculas a mayúsculas: “PUNTERO”.
- Sume 2 a cada elemento del arreglo `v[5]: {3,4,5,6,7}`.

### Problema 3:

Genere un programa que encuentre los divisores comunes entre 2 números ingresados por el usuario entre 1 y 100, los guarde en un arreglo y luego los despliegue en pantalla. Haga esto mediante una función **`void div_com(int a, int b, *int lista, *int total)`** que guarde los divisores comunes entre a y b en un arreglo “lista” y el número de divisores encontrados en “total”.



```
Los divisores comunes entre 100 y 50 son:
1
2
5
10
25
50
total = 6
Press any key to continue...
_
```

**Pistas y tips:** Declaren “lista” como un arreglo de 20 elementos y solo vayan trabajándola en su código hasta el elemento `lista[total]`.

### Problema 4:

Haga un código en donde el usuario ingrese un número n y el programa genere 2 arreglos **a** y **b** de enteros de largo n con todos sus elementos nulos. Luego genere funciones que le permitan:

- Ingresar uno a uno los valores de un arreglo.
- Hacer todos los elementos de un arreglo igual a un valor k ingresado por el usuario.
- Desplegar los valores de todos los arreglos.
- Copiar los valores de **a** en **b**.
- Comprobar si **a** es igual a **b**.
- Liberar la memoria para luego terminar el programa. (**importante**)

Haga un menú por el cual el usuario pueda acceder a todas las funciones.

### **Problema 5:**

Cree un programa al que se le pueda ingresar y guardar  $n$  puntos  $(x,y)$  en un arreglo “tabla” y luego los despliegue en pantalla. Para esto guarde cada punto como un arreglo de dos elementos y guarde estos puntos en “tabla” usando punteros.

```
Ingrese el numero de puntos.  
4  
Ingrese los valores x e y del punto 1 separados por un espacio: 3 4  
Ingrese los valores x e y del punto 2 separados por un espacio: 1 2  
Ingrese los valores x e y del punto 3 separados por un espacio: 5 6  
Ingrese los valores x e y del punto 4 separados por un espacio: 7 3  
  
Punto 1 = (3,4)  
Punto 2 = (1,2)  
Punto 3 = (5,6)  
Punto 4 = (7,3)  
Press any key to continue...
```

**Desafio:** Haga una función que le diga al usuario cual es el par de puntos más cercano entre ellos.