

Práctica 2 - Adicional

Arreglos - Punteros - Strings

1. Realice un programa que inicialice un vector con números al azar, imprima sus valores, los ordene y los vuelva a imprimir. Modularice con una función de inicialización, una función para ordenación y una función de impresión.
2. Compare las siguientes funciones. Enumere las diferencias y describa la tarea que realiza cada función. En caso de que el compilador indique una advertencia de “*operation on 'datos' may be undefined*” indique cual es el problema.

```
1 void opcion1(int datos[]){
2     int i;
3     for (i=0; i<CANT_DATOS; i++){
4         datos[i] = datos[i]*0.5;
5     }
6 }
```

```
1 void opcion2(int *datos){
2     int i;
3     for (i=0; i<CANT_DATOS; i++){
4         (*datos++) = (*datos)/2;
5     }
6 }
```

```
1 void opcion3(int *datos){
2     int i;
3     for (i=0; i<CANT_DATOS; i++, datos++){
4         *datos= *datos*0.5;
5     }
6 }
```

```
1 void opcion4(int datos[]){
2     int i;
3     for (i=0; i<CANT_DATOS; i++){
4         *datos+= *datos/2;
5     }
6 }
```

```
1 void opcion5(int datos[]){
2     int i;
3     for (i=0; i<CANT_DATOS; i++, datos--){
4         *datos+= *datos*0.5;
5     }
6 }
```

```
1 void opcion6(int *datos){
2     int i;
3     for (i=0; i<CANT_DATOS; i++){
4         *datos= *datos*0.5;
5     }
6 }
```

```
1 void opcion7(int datos[]){
2     int i;
3     for (i=0; i<CANT_DATOS; i++){
4         datos[i]= *(datos+i)*0.5;
5     }
6 }
```

```
1 void opcion8(int *datos){
2     int i;
3     for (i=0; i<CANT_DATOS; i++){
4         *(datos+i)= datos[i]/2;
5     }
6 }
```

3. Escriba las siguientes funciones:
 - a. **upperStr** que recibe como parámetro una cadena y la modifica convirtiendo todas sus letras a mayúsculas.
 - b. **lowerStr** que recibe como parámetro una cadena y la modifica convirtiendo todas sus letras a minúsculas.
 - c. **numToStr** que recibe como parámetros un entero y una cadena y retorna la cadena modificada con la representación del número en caracteres. Por ejemplo, para el número 124 la cadena de caracteres deberá quedar con “124”.
 - d. **numToText** que recibe como parámetros un entero (del 0 a 99) y una cadena y retorna la cadena modificada con la representación del número en texto. Ejemplo para el número 35 la cadena de caracteres deberá quedar con “treinta y cinco”.

Notas:

- Recuerde que la diferencia entre los códigos **ascii** de una letra minúscula y mayúscula es de un bit (el **bit 5** está en 1 para las minúsculas y en 0 para las mayúsculas).
- Recuerde agregar el **terminador** de cadenas en el ítem c).
- Uno o más arreglos de punteros a **char** podría ser una estructura útil para resolver el ítem d). Podría comenzar por resolver los números de 0 a 5, luego de 0 a 20 y finalmente los de 0 a 99.