

## Informática II – Ejercicios de práctica

### Repaso elementos básicos de C:

- 1) Mediante un programa en C verifique los tamaños que ocupan en tu PC variables de los siguientes tipos de datos:

Tipo	Tamaño (bytes)
char, unsigned char	
short int, unsigned short int	
int, unsigned int, long int, unsigned long int	
float	
double, long double	
Puntero de cualquier tipo	

```

#include <stdio.h>
#include <stdlib.h>
int main(){
    printf("Hola Mundo!\n");
    //verificar tamaño en memoria de variables:
    char a;
    unsigned char b;
    short int c;
    unsigned short int d;
    int e;
    float f;
    double g;
    float * p;
    printf("char: %d\n", sizeof(a));
    printf("unsigned char: %d\n", sizeof(b));
    printf("short int: %d\n", sizeof(c));
    printf("unsigned short int: %d\n", sizeof(d));
    printf("int: %d\n", sizeof(e));
    printf("float: %d\n", sizeof(f));
    printf("double: %d\n", sizeof(g));
    printf("float *: %d\n", sizeof(p));
    return 0;
}

```

- 2) Escribir las declaraciones apropiadas y asignar los valores iniciales dados para cada grupo de variables y arreglos:
  - a. Variables de coma flotante: a = -8.2, b = 0.005
  - b. Variables enteras: x = 129, y = 87, z = -22
  - c. Variables de carácter: c1 = w, c2 = &
  - d. Variables de doble precisión: d1 =  $2.88 \times 10^{-8}$ , d2 =  $-8.4 \times 10^5$
  - e. Variables enteras: u = 711 (octal), v = fa123 (hexadecimal)
  - f. Variable entera larga: grande = 123465789
  - g. Variable de doble precisión: c = 0.333333333
  - h. Variable de carácter: eol = carácter de nueva línea
  - i. Arreglo unidimensional de caracteres: mensaje = "Error!"
- 3) Leer dos números desde el teclado, calcule y muestre por pantalla el cociente entre el mayor y el menor

- 4) Leer tres números desde el teclado y mostrar en pantalla la suma del menor y el mayor
- 5) Desarrollar un programa que realice el cálculo del área y volumen de una esfera cuyo radio sea ingresado por teclado. Recordar que:

$$Volumen = \frac{4\pi R^3}{3} \text{ y } Area = 4\pi R^2$$

- 7) Calcular la media geométrica de una lista de números utilizando la fórmula:

### *Arreglos y punteros*

- 1) Desarrollar un programa que permita cargar un arreglo de 15 números enteros y muestre por pantalla solo los elementos pares positivos
- 2) Escribir un programa que cree un arreglo de 26 elementos y que genere y guarde en el mismo las letras minúsculas del alfabeto (sin la ñ).
- 3) Desarrollar un programa que permita cargar una tabla de 3x3 números de tipo double, luego debe recorrerla y mostrar por pantalla los elementos de su diagonal principal
- 4) Sea A una matriz de tamaño nxn, implemente un programa que dado un menú de opciones resuelva:
  - a. La transpuesta de A ( $A^t$ ).
  - b. Determine si A es simétrica o antisimétrica.
  - c. Determine si A es una matriz triangular superior o triangular inferior.
- 8) Desarrollar un programa que permita cargar un arreglo de 15 números enteros y muestre por pantalla solo los elementos ordenados

### Funciones en C

- 1) Escribir un programa que reciba dos números flotantes por valor y devuelva la suma.

- 2) Escribir un programa que reciba dos números flotantes por valor y guarde la suma en una variable global.
- 3) Modificar el programa anterior para que reciba los números y devuelva el resultado por referencia.
- 4) Realizar una función que intercambie el contenido de dos variables de tipo float mediante sus referencias.
- 5) Escribir una función que reciba las coordenadas rectangulares de dos puntos del plano y devuelva la distancia entre ellos.
- 6) Escribir una función que calcule la longitud total de un segmento de n puntos bidimensionales, respondiendo al siguiente prototipo:

```
double longitud_segmento(double puntos[][2], size_t n);
```

### *Cadenas de caracteres*

- 1) Crear una función que determine la longitud de una cadena de caracteres (mismo funcionamiento que la función strlen de string.h)
- 2) Escribir una función que reciba como argumento un arreglo de caracteres y lo devuelva invertido con la interfaz:

```
void string_reverse(char * c);
```

```

#include <stdio.h>
#include <stdlib.h>
void reverse(char * cadena)
{
    printf("\n");
    int n = sizeof(cadena);
    printf("\n%d",n);
    printf("\n");
    char temp;
    for(int i = 0; i<n/2; i++)
    {
        temp = *(cadena+i); //guardo el primer char
        printf("%c, ",temp);
        *(cadena+i) = *(cadena+n-1-i); //en la primer pos copio el
ultimo char
        printf("%c; ",*(cadena+i));
        *(cadena+n-1-i) = temp; //guardo al ultimo el primer valor
    }
}
int main(){
    char cadena[] = "12345678";
    printf("\n%s", cadena);
    reverse(cadena);
    printf("\n%s", cadena);
    return 0;
}

```

- 3) Escribir una función que reciba una cadena de caracteres s y un arreglo de caracteres con espacio suficiente t, y copie la cadena en el arreglo, terminando la cadena con el caracter '\0' (funcion strcpy() de la biblioteca ): `void strcpy(char *t, const char *s);`
- 4) Crear una función que permita obtener la parte izquierda de una cadena de caracteres definida por el número de caracteres que la forman.
- 5) Crear una función que determine cuál es el centro de una cadena y devuelva la mitad derecha de la misma.
- 6) Crear una función que concatena dos cadenas (cadena 1 y cadena 2) dejando el resultado de dicha operación sobre la primera (cadena 1).
- 7) Desarrollar un programa que permita ingresar mediante teclado una oración (texto continuo con palabras finalizada en punto, sin comas ni otros caracteres de puntuación) utilizar un buffer de tamaño suficientemente grande. A continuación contar la cantidad de palabras (separadas por espacios) y mostrar el número por pantalla.
- 8) Escribir una funcion que convierta un número de tipo *float* en una cadena de caracteres con dos cifras enteras y dos decimales (valores posibles 00,00 a 99,99).

- 9) Desarrollar un programa en C que permita cargar un arreglo de hasta 10 palabras de máximo 10 caracteres. Luego el programa deberá ordenar las palabras alfabéticamente (hacer diferencias entre mayúscula y minúsculas) mediante el método de la burbuja y utilizando la función `strcmp()`. Finalmente mostrar la lista de palabras ordenadas.