



Práctica 1 – Lenguaje C

- Dados los siguientes tipos de datos del lenguaje Pascal: *char*, *integer*, *longint*, *word*, *real* y *double*; obtener el tipo equivalente en el lenguaje C especificando la cantidad de bytes que ocupa y su cadena de formato.
- Responder:
 - ¿Cuál de los siguientes es ejemplo de un archivo de cabecera?: `printf`, `include`, `conio.h`, `#define`.
 - ¿Cuál es la librería que se requiere para utilizar cada una de las siguientes funciones?:
`printf`, `scanf`, `getch`, `getchar`.
- Describir lo que imprimen los siguientes fragmentos de código (justificar si no imprimen nada):
 - ```
char A;
A = 'a';
printf(" %c ", A); _____
printf(" %d ", A); _____
A = 'a' + 10;
printf(" %c ", A); _____
printf(" %d ", A); _____
```
  - ```
int B;
B = 5;
printf(" %d ", B); _____
printf(" %c ", B); _____
B = 5 + 'A';
printf(" %d ", B); _____
printf(" %c ", B); _____
```
 - ```
#define num 15
printf(" %d ", num); _____
printf(" %c ", num); _____
```
  - ```
#define num 15
num = 5 + 'A';
printf(" %d ", num); _____
printf(" %c ", num); _____
```
- Indicar los valores que almacenan las variables en las siguientes instrucciones de lectura:
 - ```
int a;
/* leer a = 3 */
scanf("%d", &a); _____
scanf("%f", &a); _____
scanf("%c", &a); _____
```
  - ```
char x;
/* leer x = 3 */
scanf("%d", &x); _____
scanf("%f", &x); _____
scanf("%c", &x); _____
```
 - ```
char w;
/* leer w = 2.5 */
scanf("%d", &w); _____
scanf("%f", &w); _____
scanf("%c", &w); _____
```
- Completar el resultado que muestran los siguientes programas:
  - ```
#include <stdio.h>
int main(){
    int a, b, c = 3, d = 2;
    a = 8 - 5 * 3 + 2;
    b = 7 % 3 + 4 * 2;
    printf(" valor de a %d\tvalor de b %d\n", a ,b); _____
    b %= a;
    printf(" valor de b %d\t\n", b); _____
    b = (-4) % 3;
    printf(" valor de b %d\t\n", b); _____
    if(d = a)
        printf(" d y a son iguales ");
    else printf(" d y a son distintos ");
    return 0;
}
```



b)

```
#include <stdio.h>
int main(){
    int c=3, d=2;
    c -= d++ * 3;
    printf(" valor de c %d\tvalor de d %d\n", c, d);
    return 0;
}
```

c)

```
#include <stdio.h>
int main() {
    int c = 3, l;
    c = (c * 2 - (l = 4, -- l));
    printf(" valor de c %d\t valor de l %d\n", c, l);
    getch();
    return 0;
}
```

6. Convertir las siguientes expresiones lógicas en Pascal al lenguaje C:

a) $(z < 5) \text{ AND } ((a \leq 5) \text{ OR } (c > 0))$

b) $(z = \text{CHR}(0)) \text{ OR NOT}(\text{total} = 0)$

7. Obtener el resultado de las siguientes expresiones, previa asignación de valores a las cuatro variables:

a) $a + b * c / d$

d) $a \leq b \text{ || } (3 > a)$

b) $(a + b) * c \% d$

e) $(a \geq d) \ \&\& \ (2 - 14 == c + 1)$

c) $d < 65 \ \&\& \ 3 == d$

f) $a > b \text{ || } a < c \ \&\& \ a = c \text{ || } a \geq b$

8. Completar la expresión del siguiente *if* para que se ejecute su parte verdadera y la variable a contenga el valor 1 luego de esa sentencia.

```
int a = 0;
if( )
    printf("a vale 0");
else
    printf("a vale 1");
```

9. Reescribir las siguientes sentencias utilizando la instrucción switch:

```
If (x == 4)
    y = 7;
else
    if (x == 5)
        y = 9;
    else
        if(x == 9)
            y = 14;
        else
            y = 22;
```

10. Reescribir cada una de las siguientes porciones de código en una sola sentencia mediante el uso del operador condicional:

a)

```
if(a < b)
    c = a;
else c = b;
```

b)

```
if(a < b && b < c)
    printf("ordenados");
else printf("desordenados");
```

11. Desarrollar una macro que devuelva cada uno de los siguientes resultados:

a) Máximo de 2 números

b) Caracter es alfabético

Implementar un programa que utilice las macros desarrolladas.

12. Desarrollar un programa que muestre la suma de un conjunto de números naturales ingresados por teclado. La serie finaliza al ingresar el número 0.



13. Corregir los errores del siguiente fragmento de código para que muestre los números naturales impares hasta el 15 inclusive.

```
#define <stdio.h>
#define N 9
int i;
for(i = 1; I < N; i++);
printf("%d", &I);
```

14. Desarrollar un subprograma que muestre un cuadrado con bordes de N numerales (N es parámetro de entrada)

15. Desarrollar una función que reciba tres números enteros y devuelva si son consecutivos entre sí o si no lo son.

16. Desarrollar un subprograma que reciba una cantidad de segundos y devuelva su equivalente en el formato *horas:minutos:segundos*

17. Enumerar los valores de todos los componentes de los siguientes arreglos. Especificar los arreglos de caracteres que pueden ser utilizados correctamente como cadenas.

- | | |
|-------------------------------------|--|
| a) <code>int v1[4] = {0};</code> | d) <code>char s1[4] = {'h','o','y'};</code> |
| b) <code>int v2[5] = {6};</code> | e) <code>char s2[] = {'h','o','y'};</code> |
| c) <code>int v3[] = {2,4,6};</code> | f) <code>char s3[4] = {'h','o','y','\0'};</code> |

18. Desarrollar un subprograma que duplique los elementos de un vector ubicados en las posiciones pares. Implementar 3 veces utilizando *for*, *while* y *do while*

19. Responder, siendo x un arreglo de 10 enteros (`int x[10]`):

- ¿Se puede determinar por la siguiente invocación `funarr(x, n)` si x es parámetro de entrada o de salida?
- ¿Se puede acceder al contenido de `x[10]`? ¿Qué representa el valor obtenido?

20. Realizar un programa iterativo que calcule e imprima el equivalente binario de un número decimal dado.

21. Responder:

- ¿Por qué no es correcta la siguiente sentencia: `scanf("%s", &cad)`?
- ¿Qué instrucción resulta conveniente para leer por teclado apellido y nombre de una persona: *gets* o *scanf*? Justificar la elección.
- ¿Qué representan los valores devueltos por la función *strcmp*?

22. Corregir el siguiente código para que asigne el contenido de la cadena `cad1` a `cad3`, solamente si `cad1` y `cad2` son distintas.

```
char cad1[20], cad2[20], cad3[20];
gets(cad1); gets(cad2);
if(cad1 != cad2)
    cad3 = cad1;
```

23. Rehacer el ejercicio 12 suponiendo que los números se encuentran en un archivo de texto, todos en la misma línea. ¿Como modificaría la lectura, si hubiera un valor por línea?

24. Un sensor de temperatura realiza mediciones periódicas (una cada 5 seg.). Las mismas son registradas en un archivo binario, donde cada elemento es de tipo *float*. Implementar un programa que genere dicho archivo y, otro programa que lo lea y muestre por pantalla temperatura promedio, máxima y mínima.

25. Desarrollar un programa que lea y procese un archivo de texto generado por un radar vigía. Cada línea contiene patente de un vehículo, velocidad registrada, velocidad máxima permitida, fecha y hora de la medición. Cada dato está separado por un espacio. El programa debe



generar un archivo binario con los registros donde la velocidad registrada supere en más de un 20% a la velocidad máxima. Cada registro del archivo binario contiene los mismos datos que el archivo de texto. Mostrar, al final del proceso, la cantidad de mediciones procesadas y la cantidad de registros generados en el archivo binario.

26. El ranking de una asociación de tenistas es registrada en un archivo binario. Para cada jugador se registra apellido y nombre, nacionalidad y puntaje. Realizar un subprograma que muestre los datos del jugador ubicado en una posición recibida como parámetro.