

INFORMÁTICA

Trabajo Práctico N°2 (parte 1) - Introducción a la programación

A – Responda las siguientes preguntas teóricas acerca del lenguaje de programación C.

- 1) ¿Qué son las palabras reservadas en ANSI C y cuántas existen?
- 2) ¿Cuál es la estructura general de un programa en C?
- 3) ¿Qué es un archivo de encabezado (.h) y cuál es su propósito en C?
- 4) ¿Qué es el preprocesador en C y cuáles son algunas de sus directivas más comunes?
- 5) ¿Cuáles son los tipos de datos primitivos en C y sus tamaños típicos?
- 6) ¿Cómo se manejan las variables globales y locales en C y cuál es su alcance y visibilidad?
- 7) ¿Qué son los operadores en C y cuáles son algunos ejemplos de operadores aritméticos, de asignación y lógicos?
- 8) ¿Qué son las estructuras de control del flujo de ejecución en C y cuáles son sus tipos principales?
- 9) ¿Qué es una subrutina en C y cómo se define una función?
- 10) ¿Qué es la recursión en C y cómo se estructura un procedimiento recursivo?

B – Implementación práctica: realice las siguientes actividades referidas al uso del lenguaje de programación C.

- 1) Identificadores válidos en C Indique cuáles de los siguientes identificadores no son válidos en C y justifique su respuesta:
 - a) _data123
 - b) 2start
 - c) fluencia0.2
 - d) mi_var
 - e) return
 - f) VarName
 - g) var-name
 - h) floatVar
 - i) const1
 - j) while
 - k) stop@end
- 2) Investigue sobre buenas practicas a la hora de nombrar variables (en C y en los lenguajes de programación en general. ¿Qué es el camelCase?¿Cómo deben ser los nombres de las variables?
- 3) Asigne el tipo de dato más adecuado en C para cada uno de los siguientes valores constantes:
 - a. 98.6
 - b. "hola"
 - c. X
 - d. 1234567890
 - e. 0
 - f. "44876523"

- g. "1.000.000"
- 4) Indique cuáles de las siguientes constantes no son válidas en C y justifique su respuesta:
- 3.14
 - 99
 - 8 + 4
 - "true"
 - 1,234
 - "pepe" + "honguito"
 - "dos" * 2
- 5) Indique si las siguientes expresiones son válidas y el tipo de dato que retornan en caso de ser válidas. Para las expresiones válidas, realice implementaciones en C que demuestren lo afirmado:
- 8 / 2 * 3 + 1
 - 10 == 10 || 5 < 3
 - 7 > 2 > 5
 - 4.5 * "2"
 - 15 % 4 = 3
 - 7 - 3 + 2.5
- 6) Escriba la siguiente expresión algebraica como expresión algorítmica usando el menor número de paréntesis. Considere la función `pow(base, exponente)` para el cálculo de una potencia y la función `sqrt(valor)` para la raíz cuadrada:

$$y = 1 - \frac{f^n \left[\frac{s \cdot l}{f} + \left(\frac{20}{f} \right)^w \right]}{20^n}$$

- 7) Revise los siguientes fragmentos de código, indique si son correctos o existe algún error. De ser correctos indique el resultado.

a)

```
1. int main() {
2.     int a = 5, b = 10, c;
3.     c = a + b;
4.     printf("%d\n", c);
5.     return 0;
6. }
```

b)

```
1. int main() {
2.     int x = 10;
3.     if (x > 5)
4.         printf("x is greater than 5\n");
5.     else
6.         printf("x is less than or equal to 5\n");
7.     return 0;
8. }
```

c)

```
1. int main() {  
2.     int i;  
3.     for (i = 0; i < 10; i++) {  
4.         printf("%d ", i);  
5.     }  
6.     return 0;  
7. }
```

d)

```
1. int main() {  
2.     int a = 5, b = 0;  
3.     printf("%d\n", a / b);  
4.     return 0;  
5. }
```

e)

```
1. int main() {  
2.     int arr[5];  
3.     arr[1] = 1;  
4.     arr[2] = 2;  
5.     arr[3] = 3;  
6.     arr[4] = 4;  
7.     arr[5] = 5;  
8.     return 0;  
9. }
```