



## Guía de estudio para el Examen a Título de Suficiencia de Fundamentos de Programación

**Parte I. Teoría. Instrucciones:** Responde lo que se te pide.

1. Dibuja la Arquitectura de Von Neumann

---

---

2. ¿Qué es un compilador?

---

---

3. Menciona 3 elementos del Lenguaje C.

---

---

---

4. Escribe 3 Bibliotecas utilizadas en Lenguaje C.

---

---

5. Escribe para que sirve una Constante Simbólica y como se declara en Lenguaje C.

---

---

6. ¿Para qué sirve la sentencia *break*?

---

---

7. Escribe el orden de las operaciones para encontrar el valor resultante de x de la siguiente expresión:

$$x = -21 + 5^3 - 7 * 5 + 18 / 3$$



8. Define que es un apuntador.

---

---

9. Escribe para qué sirve el operador \* y &.

---

---

10. Explica qué es la asignación dinámica de memoria.

---

---

11. ¿Cuál es la diferencia entre declarar una **función** y un **prototipo** en Lenguaje C? y escribe la sintaxis de cada uno.

---

---

12. Escribe la 1 biblioteca y 4 funciones utilizadas en Lenguaje C para el manejo de archivos.

---

---

13. Escribe la sintaxis de la función para abrir y cerrar un archivo.

---

---

14. Escribe 4 de los *modos* disponibles para abrir un archivo y explica cuál es el uso de cada uno.

---

---

15. Escribe la sintaxis de las funciones para escribir y leer datos de un archivo.

---

---

16. Escribe el concepto de recursividad y para cómo es aplicado en la programación estructurada.

---

---



**17. Escribe las diferencias entre memoria estática y memoria dinámica.**

---

---

**18. Describe cómo es la reservación dinámica de memoria.**

---

---

**19. Escribe tres razones por las cuales es importante escribir comentarios en el código fuente.**

---

---

## Parte II. Ejercicios

1. Realiza un programa que pida al usuario introduzca una frase, calcula el tamaño de la cadena e indica si la frase es un palíndromo.

2. Realiza un programa que calcule la desviación estándar de una lista de n números, la fórmula para calcular la desviación estándar es:

$$\sqrt{\frac{\sum_{i=1}^n (x_i - m)^2}{n - 1}}$$

\* Donde m es el valor de la media de la lista de números.

\* Pide al usuario que introduzca el valor de n.

\* También genera de manera aleatoria los n números, que sean valores entre 0 y 500 y guárdalos en un arreglo dinámico.

\* Imprime el valor de n, m y la desviación estándar.

## Ejemplo:

Si la cantidad de números fueran 3

20	498	100
----	-----	-----



Desviación  
estándar =

$$\sqrt{\frac{(20-m)^2 + (498-m)^2 + (100-m)^2}{2}}$$

3. En un país hubo elecciones para elegir a su presidente. El país consta de  $n$  regiones de las cuales se han levantado actas que contienen el total de votos obtenidos por  $m$  partidos políticos en dicha región. Realiza un programa en c que pida al usuario los valores de  $n$  y de  $m$ , crea una matriz dinámica de  $n*m$ , donde cada elemento represente los votos obtenidos de un partido en una región. Llena estos datos con aleatorios, donde el máximo valor devuelto sea 500 y muestra la información en pantalla.

- Realiza una función que reciba como parámetro la matriz de votos, en esta función calcula y muestra en pantalla la suma total de los votos obtenidos por cada partido.
- Calcula el porcentaje de votos que obtuvo cada partido en todo el país.
- Tomando en cuenta el punto anterior, calcula y muestra en pantalla qué partido ganó las elecciones en el país y también muestra con cuántos votos ganó.

4. Escribir una función para transformar un número entero en una cadena de caracteres formada por los dígitos del número entero.

5. Crea una matriz dinámica de tamaño  $n*m$  (el valor de  $n$  y  $m$  pídelo al usuario) y calcula la suma de cada una de sus filas y columnas, colocando dichos resultados en dos arreglos, en uno se almacene la suma de las filas y en otro la suma de las columnas.

6. Una empresa tiene  $n$  almacenes y necesita crear un programa que lea las ventas de cada mes del año de los  $n$  almacenes, calcular la media de ventas por mes y obtener un listado por cada mes de los almacenes cuyas ventas mensuales son superiores a la media por mes. Pide el valor de  $n$  al usuario y crea una matriz dinámica en la que almacenes las ventas de cada almacén en cada mes del año.

7. Guarda en una matriz las ventas realizadas por 5 supermercados (Wal-Mart, soriana, Superama, Aurrera, Chedraui) en 7 regiones del país (Lindavista, centro, Azcapotzalco, aeropuerto, Coyoacán, Iztapalapa y Buenavista). Guarda en otra



matriz los nombres de los supermercados y en una tercer matriz los nombres de las regiones del país.

- Muestra el total de ventas realizadas por cada supermercado, por ejemplo:

SUPERMERCADO	VENTAS
WALMART	12000.00
SORIANA	9000.00
SUPERAMA	10500.00
...(ETC)	

- Muestra el total de ventas realizadas por cada región, por ejemplo:

REGION	VENTAS
LINDAVISTA	340000.00
CENTRO	78900.00
AZCAPOTZALCO	230500.00
...(ETC)	

También muestra el nombre del supermercado cuyas ventas fueron más altas

**8.** Se dice que una matriz tiene un punto de silla si alguna posición de la matriz es el menor valor de su fila, y a la vez el mayor de su columna. Escribir un programa que tenga como entrada una matriz de números reales y calcule la posición de un punto de silla (si es que existe).

**9.** Se quiere sumar números grandes, tan grandes que no pueden almacenarse en variables de tipo long. Por lo que se ha pensado en introducir cada número como una cadena de caracteres y realizar la suma extrayendo los dígitos de ambas cadenas. Hay que tener en cuenta que la cadena suma puede tener un carácter más que la máxima longitud de los sumandos.

**10.** Realiza un programa donde con una estructura representes una hora (horas, minutos y segundos), pide al usuario que introduzca una hora (horas, minutos y segundos) y que introduzca una cantidad de segundos, suma la cantidad de segundos a la hora y muestra el resultado en pantalla.

**11.** Escribir un programa que permita hacer las operaciones de suma, resta, multiplicación y división de números complejos. El tipo complejo ha de definirse como una estructura.

**12.** En una competición de natación se presentan 16 nadadores. Cada nadador se caracteriza por su nombre, edad, prueba en la que participa y tiempo (minutos, segundos) de la prueba. Escribir un programa que lea los datos de los 16



nadadores y calcule la desviación estándar respecto al tiempo. Realiza la lectura de los datos desde un archivo y representa la información de los nadadores con un tipo de dato compuesto.

**13.** Realiza la prueba de escritorio del siguiente Código Fuente en Lenguaje C, muestra cuantas iteraciones del ciclo *while* realiza y calcula el resultado de la variable *sum* al término de su ejecución.

```
#include <stdio.h>
void main (void){
    float sum =0.0, j=1.0, i=2.0;
    while( i/j > 0.05){
        j = j + j;
        sum = sum + i/j;
        printf("%f \n", sum);
    }
}
```

**14.** Escribir un programa que reciba un número entero mayor a 0 y menor a 10. El número recibido, será el número de minutos que se deberá contar en forma regresiva hasta llegar a 0 segundos, se debe mostrar en todo momento los minutos y segundos faltantes para llegar a cero. Finalmente, cuando el contador regresivo llegue a cero segundos, se debe mostrar el mensaje “Se terminó el tiempo”.

**Nota:** Para el desarrollo del programa debes utilizar el ciclo do-while.

**15.** Escribe la estructura de datos para la siguiente información.

- Id (entero)
- Nombre de estudiante
  - Nombre(s) (cadena)
  - Apellido paterno (cadena)
  - Apellido materno (cadena)
- CURP (cadena)
- Edad (entero)
- Teléfono (cadena)
- Correo electrónico (cadena)
- Dirección (cadena)

**16.** De la estructura declarada en la pregunta anterior, declara un apuntador de dicha estructura y escribe la sintaxis correspondiente para guardar los datos que proporcione el usuario, recuerda utilizar el operador adecuado.



**17.** Escribe un programa que guarde la estatura de un número de personas *indicado por el usuario* en un arreglo declarado mediante el manejo dinámico de memoria e imprima el promedio de dichas estaturas.

(Escribir a mano el código Fuente en C con su respectiva prueba de escritorio).

**18.** Dado el siguiente código fuente en C, escribe que valor tienen las variables solicitadas al término de su ejecución y desarrolla su prueba de escritorio a mano.

Considera que la variable *x* comienza en la dirección de memoria: 61FF10, mientras que la variable *y* comienza en la dirección de memoria 61FF0C.

```
#include <stdio.h>
int funcion( int *, int * );

int main( void ){
    int x, y, *apx, *apy;
    int valorRetorno;
    x = 20;
    y = 34;
    apx = &x;
    apy = &y;
    printf( "Valor de x: %d y de y: %d",x,y );
    printf("\n El valor del apuntador de x: %p y del apuntador de y: %p",apx, apy);
    valorRetorno = funcion( apx, apy );
    printf( "\n Nuevo valor de x: %d y de y: %d",x,y );
    printf( "\n El valor de valorRetorno es: %d", valorRetorno );
    return 0;
}

int funcion( int *apx, int *apy ){
    int variable;
    *apx = *apx * 4;
    *apy = *apx - *apy;
    variable = *apx + *apy;
    return variable;
}
```

**Valores de:**

x: \_\_\_\_\_  
y: \_\_\_\_\_  
&x: \_\_\_\_\_  
&y: \_\_\_\_\_  
apx: \_\_\_\_\_



apy: \_\_\_\_\_  
valorRetorno: \_\_\_\_\_

**19. Construye un programa en Lenguaje C que resuelva el siguiente problema matemático utilizando recursividad:**

$$\left\{ \begin{array}{l} b(1) = -5 \\ b(n) = b(n-1) + 9 \end{array} \right.$$

**20. Tomando en cuenta el siguiente fragmento de código:**

- Coloca lo que contienen las celdas de memoria asignadas a a, pa y pb cuando se ejecuta la última instrucción.
- Coloca que imprime en pantalla.

```
char a='c';  
char *pa,*pb=&a;  
pa=pb;  
printf    *pa;  
*pb='z';  
printf    a;  
printf    pa;  
printf    pb;  
printf    &a;  
*pa='i';  
printf    *pb;  
printf    a;
```

	Memoria	Dirección (decimal)
a		221
		222
		223
pa		224
pb		225
		226

**21. Tomando en cuenta el siguiente fragmento de código:**

- Coloca lo que contienen las celdas de memoria asignadas a v[0], v[1], v[2],...,v[5] y p cuando se ejecuta la última instrucción.
- Coloca las direcciones de las celdas asignadas a v[1], v[2], v[3], v[4] y v[5]
- Coloca que imprime en pantalla.





```
int v={9,-82,39,-99,0,100};  
int *p,i,b;  
*a=-3;  
p=a;  
printf *(v+3);  
printf *p+5;  
*(p+2)=270;  
*(p+5)=4;  
b=sizeof(v)/sizeof(int);  
for(i=0;i<b;i++)  
    printf \n v[i]  
printf *(v+2)+2
```

	Memoria	Dirección (decimal)
p		90
v	111	
v[0]		111
v[1]		
v[2]		
v[3]		
v[4]		
v[5]		