



## Codificar en lenguaje C la solución de los siguientes problemas

### Problemas Resueltos

#### 1. Variables Simples.

- a. Ingresar un carácter, luego presentar por pantalla un mensaje indicando si es o no una letra del alfabeto.

b.


```
#include <stdio.h>
int main(int argc, char *argv[])
{
    char    x;
    printf("Ingrese un caracter: ");
    scanf("%c",&x);
    if (x>='A' && x<='Z' || x>='a' && x<='z')
        printf("Es una letra del alfabeto\n");
    else
        printf("No es una letra del alfabeto\n");
    return (0);
}
```

- c.  Mostrar por pantalla, las N primeras letras en mayúscula del alfabeto.

#### 2. Variables tipo puntero.

- a. Asignar a una variable `index` un valor constante, declarar dos punteros. Asignarle al primer puntero la dirección de memoria de la variable `index`, copiar el contenido del primer puntero al segundo puntero declarado. Mostrar por pantalla el contenido de cada una de las variables. Luego modificar el valor de `index` utilizando el primer puntero, presentar por pantalla el contenido de cada variable.

```
#include <stdio.h>
int main (int argc, char *argv[])
{
    int  index,*ptr1,*ptr2;
    index = 39;
    ptr1 = &index;        // ptr1 apunta a index
    ptr2 = ptr1;           // ptr2 toma el valor de ptr1
    printf("El valor es %d %d %d\n",index,*ptr1,*ptr2);
    *ptr1 = 13; /*a index se asigna 13 por indirección */
    printf("El valor es %d %d %d\n",index,*ptr1,*ptr2);
    return 0;
}
```

- b.  Ingresar un número entero, luego asignar la dirección de memoria de dicha variable a un puntero. Presentar por pantalla la información que contiene la variable puntero (dirección e indirección) y la variable entera almacenada.



**Los siguientes problemas, deben ser resueltos utilizando una o más funciones**  
**En caso de utilizar parámetros, No debe usar más de tres parámetros.**

**ATENCIÓN!!!**

En la función principal `int main (int argc, char *argv[])` **SOLAMENTE deberá:**


- Ingresar los datos de entrada solicitados en cada problema.
- Invocar a la/s función/es para el procesamiento de los datos de entrada.
- Presentar los resultados retornados por la/s función/es.

Recordar que las funciones definidas por el usuario deben ser:

- declaradas (declaración),
- implementadas (definición) y
- usadas (invocar o llamar).

- Resolver utilizando funciones que retornen valor y que empleen a los sumo tres parámetros por valor.
  - Ingresar un número real, presentar por pantalla el valor del número elevado al cuadrado (realizar el cálculo mediante una función).

```
#include <stdio.h>
float cuadrado (float y);
int main(int argc, char *argv[])
{
    float    x, resultado;
    printf("ingrese un numero real: ");
    scanf("%f",&x);
    resultado = cuadrado(x);
    printf("\nEl resultado de %f elevado al cuadrado", x);
    printf(" es %f",resultado);
    return (0);
}
float cuadrado (float y)
{
    return(y*y);
}
```

-  Ingresar dos números enteros y realizar la suma de los mismos.



4. Resolver utilizando funciones que no retornen valor, pero deben emplear parámetros por valor y/o parámetros por referencia (no usar más de tres parámetros en total).
- a. Ingresar dos números complejos (ingresando parte real y parte imaginaria), luego presentar por pantalla los números complejos ingresados y la suma de dichos números. Debe utilizar una única función para sumar.


```
#include <stdio.h>
void suma (int y, int z, int *r);
int main(int argc, char *argv[])
{
    int    ar, ai, br, bi, resultador, resultadoi;

    printf("Ingrese un número complejo: \n");
    printf("\tparte real del 1er numero: ");
    scanf("%d",&ar);
    printf("\tparte imaginaria del 1er numero: ");
    scanf("%d",&ai);
    printf("\nIngrese el segundo numero complejo: \n");
    printf("\tparte real del 2do numero: ");
    scanf("%d",&br);
    printf("\tparte imaginaria del 2do numero: ");
    scanf("%d",&bi);

    suma(ar, br, &resultador);
    suma(ai, bi, &resultadoi);

    printf("\nEl resultado de: ");
    printf("\n\t(%d+%di) ", ar, ai);
    printf("+ (%d+%di)", br, bi);
    printf(" es: %d+%di", resultador, resultadoi);
return (0);
}

void suma (int y, int z, int *r)
{
    *r=y+z;
}
```

- b.  Ingresar dos números enteros y realizar la suma de los mismos.



### Problemas Propuestos

1. Ingresar dos letras del alfabeto en minúscula (realizar el control), luego presentar por pantalla dichas letras, pero ordenadas en forma alfabética, o un mensaje en caso de que sean iguales.
2. Ingresar un número entero no negativo expresado en binario (realizar el control de datos de entrada), presentar por pantalla el número ingresado pero expresado en decimal
3. Ingresar dos números complejos (ingresando parte real y parte imaginaria en variables separadas), luego presentar por pantalla los números ingresados, la suma y la diferencia de dichos números.

### Resolver los siguientes problemas utilizando variables tipo puntero

4. Ingresar un número, luego presentar por pantalla el valor duplicado de dicho número.
5. Ingresar un número no negativo (realizar el control de datos de entrada), presentar por pantalla el número ingresado y el valor del factorial del mismo.
6. Presentar por pantalla la sumatoria de los números primos menores que un valor entero N ingresado.
7. Ingresar un número entero, luego utilizando una variable tipo puntero, presentar por pantalla el valor ingresado, el domicilio de memoria en donde se aloja dicho valor y el domicilio de memoria del puntero utilizado.

### Resolver utilizando funciones que retornen valor y que empleen a lo sumo tres parámetros por valor.

8. Ingresar dos números B y E, siendo B un valor real y E un valor entero no negativo (realizar el control mediante una función), presentar por pantalla el valor de B elevado a E; para ello deberá elaborar una función que realice dicha tarea (tener en cuenta, que el resultado es un número real).
9. Ingresar dos números N y D enteros positivos (realizar el control con una función), luego realizar (con otra función) la división entera entre N y D utilizando el método de restas sucesivas. Presentar por pantalla el resto de la división
10. Ingresar un número entero no negativo en formato binario (realizar el control con una función), presentar por pantalla el número ingresado pero expresado en decimal (realizar la conversión con otra función)
11. Leer un número entero positivo de dos o más dígitos, deberá hacer el control de datos de entrada empleando una función. Presentar por pantalla al número ingresado y su invertido, para hacer esta tarea, deberá utilizar otra función.
12. Ingresar dos números enteros N y B, controlar mediante una función que N sea no negativo y mediante otra función controlar que B sea mayor que 1 y menor que 10, presentar por pantalla el valor ingresado pero convertido a base B, utilizar una función para realizar la conversión a la base B.
13. Ingresar un número entero positivo (realizar el control con una función). Luego mostrar por pantalla un mensaje indicando si el número ingresado es o no primo (utilizar una función para determinar si el número es o no primo).



14. Presentar por pantalla la sumatoria de los primeros números impares positivos menores que un valor entero N ingresado. (realizar una función que realice la sumatoria, a su vez dicha función debe invocar a otra función para determinar si un número es o no impar).

**Resolver utilizando funciones que pueden o no retornar valor, pero deben emplear parámetros por valor y/o parámetros por referencia (no usar más de tres parámetros en total).**

15. Ingresar dos letras del alfabeto en minúscula (controlar con una función), luego presentar por pantalla dichas letras, pero ordenadas en forma alfabética, o un mensaje en caso de que sean iguales (debe realizar una función para ordenar las letras y determinar si son iguales).
16. Ingresar dos números N y D enteros positivos (realizar el control con una función), luego calcular el cociente y el resto entre N y D (con una función) utilizando el método de restas sucesivas. Presentar por pantalla el cociente y el resto de la división
17. Ingresar tres números enteros, ordenarlos en forma descendente mediante una función. Luego, presentar por pantalla los números ordenados, o un mensaje en caso de que los números ingresados sean iguales.
18. Realizar un programa que permita ingresar N números, luego presentar por pantalla el menor valor impar ingresado (debe utilizar una función para encontrar el menor valor impar y determinar si se ingresaron o no números impares). En caso de que no se ingrese ningún valor impar, presentar el mensaje correspondiente. Ejemplo: Se ingresa: 7, -2, 10, -5, -11, 3, 10, -2 Muestra: El menor valor impar ingresado es -11
19. Ingresar dos números enteros positivos N1 y N2, debe controlar mediante una función que N1 sea menor que N2, luego sumar los números primos comprendidos entre N1 y N2 (no considerar en la sumatoria a N1 ni a N2 en caso de que alguno o ambos sean primos); la suma la debe realizar una función que a su vez invocará a otra función para determinar si un número es o no primo. Presentar por pantalla el resultado.

Ejemplo: N1 = 5, N2 = 12; Suma = 7 + 11 = 18

20. Ingresar N números, presentar por pantalla el mayor valor ingresado, indicando en qué posición ingresó o un mensaje, en caso de que todos los números ingresados sean iguales, debe utilizar una función para encontrar el mayor valor ingresado y su posición correspondiente. Ejemplo: Se ingresa -8, 0, 4, -9, 7, 11, 74, 15 Muestra: El mayor valor ingresado es 74 e ingresó en la posición número 7

**Resolver utilizando funciones que no retornen valor, utilizando parámetros por valor y/o parámetros por referencia (no debe usar más de tres parámetros en total).**

21. Ingresar dos números complejos (ingresando parte real y parte imaginaria), luego presentar por pantalla los números complejos ingresados, la suma y diferencia de dichos números. Debe utilizar funciones para sumar los números complejos y para efectuar la diferencia
22. Ingresar un número real positivo B y un número entero no negativo E (realizar los controles de datos mediante funciones), encontrar el valor de B elevado a E (utilizando una función), utilizando únicamente operaciones de producto. Presentar por pantalla el resultado.



23. Realizar un programa que permita ingresar dos números enteros N1 y N2, luego mostrar por pantalla el resultado de la suma de los números primos comprendidos entre ellos (utilizar una función para calcular la suma, además dicha función debe invocar a otra función para determinar si un número es o no primo).
24. Ingresar un número no negativo (realizar el control de datos de entrada mediante una función), presentar por pantalla el número ingresado y el valor del factorial del mismo (utilizar una función para calcular el valor del factorial).
25. Realizar un programa que permita ingresar dos números enteros positivos N y B, donde B debe ser menor que 11 y mayor que 1 (realizar el control de datos de entrada mediante funciones), luego presentar por pantalla el número N expresado en la base B (utilizar una función para realizar la conversión).
26. Realizar un programa que permita ingresar N números enteros positivos (controlar mediante una función que se ingresen solamente números positivos), luego determinar el valor promedio (o media aritmética) de los números ingresados que sean primos (debe utilizar una función para realizar el cálculo del promedio, dicha función invocará a otra función para determinar si un número es o no primo). Mostrar por pantalla el resultado o un mensaje en caso de que no se hayan ingresado valores primos. Recordar que el promedio no es un valor entero.

Resolver utilizando funciones que utilicen solamente un parámetro por valor, que retorne un valor y que haga uso de variables static (y auto si fuese necesario).

27. Realizar un programa que permita ingresar N números enteros, luego presentar por pantalla la cantidad de series de números impares ingresados.

Ejemplo: 4, 9, 2, 5, 3, 2, 15, 3, 7, 9

|----| |-----|  
1 2

2 series

28. Ingresar N números enteros positivos, luego presentar por pantalla cuantas series de números estrictamente crecientes aparecen.

Ejemplo: 5, 1, 9, 7, 6, 0, 1, 9, 3, 2, 4, 7, 9

|--| |-----| |-----|  
1 2 3

3 series



**Resolver utilizando una o más funciones que retornen valor, no utilicen parámetros y que haga uso de variables locales static (y auto si fuese necesario).**

29. Presentar por pantalla, las N primeras letras del alfabeto en mayúsculas.
30. Un número triangular es un número que puede recomponerse en la forma de un triángulo equilátero (por convención, el primer número triangular es el 1). Es decir, los primeros números triangulares son 1, 3, 6, 10, etc. Elabore un programa que muestre en pantalla los N primeros números triangulares.
31. Presentar por pantalla los N primeros números de la serie de Fibonacci.

