

Guía

Funciones

Resuelva los siguientes ejercicios en lenguaje C.

Para la solución de los siguientes ejercicios, no debes imprimir resultados dentro de las funciones que escribas. Los resultados deben ser devueltos mediante el return de la función.

Importante:

Luego de escribir cada función, probala, invocándola desde el bloque principal del programa, pasándole distintos valores para que la prueba contemple varias alternativas y así estar seguro que funciona adecuadamente.

1) Escribir una función que reciba un valor n , entero, y devuelva la suma de los valores entre 0 y n .

Ejemplos:

`suma_n(5) = 15`

`suma_n(120) = 7260`

`suma_n(120120) = 7214467260`

2) Escribir una función que dado un número entero, devuelva un valor booleano que indique si dicho número es primo o no. Antes de plantear una solución ten presente las siguientes preguntas.

a) Con sólo encontrar un divisor del número a evaluar distinto a uno y a sí mismo, ya puedo afirmar que el número no es primo. ¿Tiene sentido seguir evaluando más divisores?

b) Teniendo en cuenta que todo número par a excepción del 2, no es primo, ¿Tiene sentido seguir en un ciclo, si al calcular el resto de la división del número a evaluar por 2, el resultado es cero?

c) Puedo encontrar un divisor del número a evaluar que sea mayor al número a evaluar dividido 2?

- 3) Escribir una función que reciba un valor entero y calcule el factorial del mismo. Si no se puede calcular el factorial del valor recibido, la función deberá devolver 0, de lo contrario deberá devolver el valor calculado.
- 4) Escribir una función que dados 2 números, calcule el porcentaje que el primero representa respecto del segundo.
- 5) Escribir una función que dados a y b ambos de tipo **entero** devuelva a^b de tipo **double** para los siguientes casos (**no se puede utilizar ninguna biblioteca de funciones matemáticas**). Implementar una solución iterativa.

En las pruebas debe invocar a la función para los siguientes 5 casos:

- i) $a > 0$ y $b > 0$
- ii) $a > 0$ y $b < 0$
- iii) $a < 0$ y $b > 0$
- iv) $a < 0$ y $b < 0$
- v) $a > 0$ y $b = 0$

Ejemplos:

- i) $a = 2$; $b = 3$; resultado = 8
- ii) $a = -2$; $b = 3$; resultado = 8
- iii) $a = -2$; $b = 0$; resultado = 1
- iv) $a = -2$; $b = 0$; resultado = 1

- 6) Escribir una función que dado un número entero positivo retorne un valor booleano que indique si el mismo es capicúa (palindromo).
- 7) Escribir una función que dados a y b devuelva el cociente de la división entera, sin utilizar el operador correspondiente del lenguaje. Para esta resolución deben considerar la utilización de restas sucesivas.
- 8) Escribir una función que dados a y b devuelva el resto de la división entera, sin utilizar el operador correspondiente de lenguaje. Para esta resolución deben considerar la utilización de restas sucesivas.

9) Escribir una función que dada una fecha en formato **aaaammdd** (número entero) la convierta en los parámetros numéricos **aaaa**, **mm** y **dd**.

Ejemplo:

fecha: 20140321 -> día=21 mes=3 año=2014

10) Escribir una función que dado un tiempo expresado en segundos, devuelva por parámetros el equivalente en días, horas, minutos y segundos.

Utilizar esta función dentro de un programa que solicite el valor al usuario.

Se debe validar que el valor ingresado sea entero positivo, de lo contrario, deberá mostrarse el mensaje: "Valor ingresado inválido".

Ejemplo:

Valor ingresado representando una cantidad de segundos: 1234567

Resultado: Días: 14, Horas: 6, Minutos: 56, Segundos 7

Para verificar el resultado pueden ir a la siguiente web:

<https://www.satsig.net/training/seconds-days-hours-minutes-calculator.htm>

11) Escribir una función que dado un código numérico de **m** dígitos, retorne un sub-código consistente en una cantidad $0 < n < m$ de dígitos a partir del **k-ésimo** lugar a contar del dígito menos significativo.

Los parámetros de la función deben ser **código**, **n** y **k**.

En el caso que los parámetros no sean consistentes y la extracción no puede realizarse la función debe retornar **-1**.

Si el sub-código contiene ceros a izquierda serán ignorados en el valor entregado como resultado.

Ejemplo a: código = 48573628 (m = 8) n = 4 k = 3 resultado = 8573	Ejemplo b: código = 1234 (m = 4) n = 3 k = 0 resultado = 234	Ejemplo c: código = 1234 (m = 4) n = 3 k = 1 resultado = 123
Ejemplo d:	Ejemplo e:	Ejemplo f:

código = 2002 (m = 4) n = 3 k = 0 resultado = 2	código = 2002 (m = 4) n = 0 k = 0 resultado = -1	código = 2002 (m = 4) n = 5 k = 3 resultado = -1
--	---	---

12) Escribir una función que reciba dos valores enteros, y devuelva el máximo común divisor entre ambos números. Recordemos que se define el máximo común divisor (MCD) de dos o más números enteros al mayor número entero que los divide sin dejar resto alguno. **Utilizar el algoritmo de Euclides.**

13) Escribir una función que reciba un mes y un año como números enteros; y devuelva la cantidad de días del mes, considerando los años bisiestos. Tenga en cuenta que un año bisiesto es aquel divisible por 4, salvo que sea divisible por 100, en cuyo caso también debe ser divisible por 400.

14) Un número entero positivo se dice perfecto si es igual a la suma de todos sus divisores excepto él mismo. Ejemplo: los números 6, 28 y 496 son perfectos.
(6 = 1+2+3)
(28 = 1+2+4+7+14)
(496 = 1+2+4+8+16+31+62+124 +248)

Se pide:

- a)** Escribir una función booleana que permita discernir si un número (único parámetro) es perfecto.
- b)** Dar un ejemplo de cómo se hace referencia a dicha función desde un programa o desde otro subprograma.

15) Diseñar un algoritmo que calcule la aproximación del seno:

sen(x)= x - (x³)/3! + (x⁵)/5! - (x⁷)/7! +

$$\text{sen } x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + \frac{(-1)^n x^{2n+1}}{(2n+1)!} + \dots = \sum_{k=0}^{+\infty} \frac{(-1)^k x^{2k+1}}{(2k+1)!} \quad (\forall x \in \mathbb{R}).$$

Nota: Una buena aproximación está dada por el desarrollo de 10 términos como mínimo.

Utilizar la función de potenciación del ejercicio 3 y la del factorial del ejercicio 5.

16) Dos números se dicen amigos cuando uno de ellos es igual a la suma de todos los divisores del otro excepto el mismo. Ejemplo: los números 220 y 284 son amigos.

(284 = 1+2+4+5+10+11+20+22+44+55+110)

(220 = 1+2+4+71+142)

Se pide:

- a)** escribir una función booleana que llamada desde un programa, permita discernir si dos números (parámetros) son amigos.
- b)** Usar otra función para calcular la suma de los divisores de un número determinado.
- c)** Hacer referencia a dichas funciones desde un programa o desde otro subprograma.

17) El producto de Wallis es una expresión matemática, utilizada para representar el valor del número Pi, que fue descubierta por John Wallis en 1655 y que establece que:

$$\prod_{n=1}^{\infty} \left(\frac{2n}{2n-1} \cdot \frac{2n}{2n+1} \right) = \frac{2}{1} \cdot \frac{2}{3} \cdot \frac{4}{3} \cdot \frac{4}{5} \cdot \frac{6}{5} \cdot \frac{6}{7} \cdot \frac{8}{7} \cdot \frac{8}{9} \cdots = \frac{\pi}{2}$$

Escribir una función, que reciba por parámetro, el valor más alto a utilizar en el cálculo (n). La función debe calcular el valor de Pi utilizando la fórmula de Wallis y devolver el valor de Pi obtenido.

Proba la función, utilizando al menos, como valor de n, 100, 1000 y 10000.

Fuente de consulta: https://es.wikipedia.org/wiki/Producto_de_Wallis