

BOLETIN DE EJERCICIOS

TEMA 4



Extraídos y adaptados de “Ramón Mollá Vayá; Inmaculada García García; Laura Sebastián Tarín; Jon Ander Gómez Adrián; Jose Miguel Alonso Abalos; David Guerrero López; Miguel Ángel Martín Caro(2005). **Problemas resueltos en lenguaje C**. Editorial UPV. 8497058836”

Ejercicios de selección

1. Escribir un programa que, dados los lados de un triángulo a , b y c , indique si es equilátero (tres lados iguales), isósceles (sólo dos lados iguales) o escaleno (tres lados distintos) y que calcule su área mediante la fórmula de Herón:

$$A^2 = p(p-a)(p-b)(p-c), \text{ donde } p = \frac{a+b+c}{2}$$

Análisis

En este problema, dados los lados de un triángulo, se pide, por un lado, clasificar este triángulo según sus lados y, por otro, hallar el área del mismo.

En primer lugar, se determina cuáles son los datos de entrada y de salida y un esbozo del proceso a realizar para obtener estos datos de salida a partir de los datos de entrada:

- datos de entrada: lados del triángulo
- datos de salida: tipo de triángulo y su área
- proceso: comprobar el número de lados iguales del triángulo para determinar su tipo y aplicar la fórmula de Herón dada en el enunciado para hallar el área.

Para poder indicar de qué tipo es el triángulo, hay que comparar sus lados dos a dos (las expresiones relacionales son binarias) y determinar cuántos son iguales. Es decir:

- Si el lado a y el lado b son iguales, ya se tienen dos lados iguales. Si además el lado c también lo es (igual al lado a o al lado b , ya que son iguales entre sí), entonces los tres lados son iguales y por tanto, el triángulo es equilátero. Pero si el lado c no es igual, entonces sólo se tienen dos lados iguales y el triángulo es isósceles.
- Si el lado a y el lado b no son iguales, ya se sabe que el triángulo no puede ser equilátero. Entonces se debe comprobar si alguno de ellos es igual al lado c . Si se determina que el lado a o el lado b son iguales al lado c , entonces el triángulo es isósceles. Sino, todos los lados son distintos entre sí y por tanto el triángulo es escaleno.

En cuanto al cálculo del área, es necesario obtener en primer lugar el valor del semiperímetro p , para utilizarlo en la expresión del área. También hay que tener en cuenta que se debe calcular la raíz cuadrada para obtener el área real del triángulo.

A continuación, se muestra un esquema más detallado del funcionamiento del programa:

1. Leer los lados del triángulo y almacenarlos en las variables `lado_a`, `lado_b`, y `lado_c`.
2. Determinar el tipo de triángulo con el siguiente proceso:
 - a. Si la variable `lado_a` es igual a la variable `lado_b` entonces:
 - i. Si la variable `lado_a` es igual a la variable `lado_c` entonces escribir por pantalla que el triángulo es equilátero
 - ii. Sino, escribir por pantalla que el triángulo es isósceles
 - b. Sino:
 - i. Si la variable `lado_a` es igual a la variable `lado_c` o la variable `lado_b` es igual a la variable `lado_c` entonces escribir por pantalla que el triángulo es isósceles
 - ii. Sino, escribir por pantalla que el triángulo es escaleno
3. Calcular el semiperímetro según la expresión dada y almacenarlo en la variable `p`.
4. Calcular el área al cuadrado del triángulo según la fórmula de Herón, almacenando el resultado en la variable `area_cuadrado`.
5. Calcular la raíz cuadrada de la variable `area_cuadrado` y almacenar el resultado en la variable `area`.
6. Escribir en pantalla el contenido de la variable `area`.

2. Escribir un programa que lea las edades de tres personas escogidas al azar y muestre por pantalla la edad mínima. Modificar este programa para que procese las edades de cuatro personas en lugar de tres.

Análisis

Este problema consiste en leer las edades de tres personas e imprimir por pantalla la menor edad de las leídas. Por tanto, los datos de entrada serán las edades de las tres personas y como datos de salida se tendrá la edad mínima.

El proceso a realizar consiste en calcular el mínimo de tres valores. Hay distintas formas de hacerlo. En este momento, se utilizará una de las más sencillas que consiste en comparar todos los valores entre sí para obtener el mínimo. Es decir:

- si $\text{edad1} < \text{edad2}$, entonces edad1 es el mínimo entre edad1 y edad2 , por lo que se compara con edad3 :
 - o si $\text{edad1} < \text{edad3}$ entonces la edad mínima será edad1
 - o sino, se tiene $\text{edad3} < \text{edad1}$ y $\text{edad1} < \text{edad2}$, por lo que la edad mínima será edad3
- si edad1 no es menor que edad2 , entonces habrá que comparar edad2 con edad3 :
 - o si $\text{edad2} < \text{edad3}$ entonces la edad mínima será edad2
 - o sino, edad3 será la mínima

Se deja como ejercicio, pensar cómo modificar este esquema para calcular el mínimo de la edad de cuatro personas.

3. En un chiringuito de alquiler de vehículos acuáticos disponen de la siguiente lista de precios:

- Patín a pedales: 2€/hora
- Patín a motor: 6€/hora
- Tabla de windsurf: 4€/hora
- Moto acuática: 7€/hora

Los vehículos pueden alquilarse por horas o por fracciones de media hora. Se pide escribir un programa que, dado el tipo de vehículo y el tiempo que se desea alquilar, devuelva el precio total.

Análisis

Este problema consiste en calcular el precio total a pagar por el alquiler de un vehículo acuático según el tipo de vehículo y el tiempo de alquiler.

Si solamente hubiera un tipo de vehículo, el precio total se obtendría multiplicando el precio por hora de alquiler y el tiempo en horas que se va a alquilar el vehículo. Como en este problema hay distintos tipos de vehículos con distintos precios, se debe efectuar el mismo cálculo pero distinguiendo por tipo de vehículo. Es decir:

- si el vehículo es un patín a pedales, el precio total será 2 euros por el tiempo de alquiler en horas
- si el vehículo es un patín a motor, el precio total será 6 euros por el tiempo de alquiler en horas
- si el vehículo es una tabla de windsurf, el precio total será 4 euros por el tiempo de alquiler en horas
- si el vehículo es una moto acuática, el precio total será 7 euros por el tiempo de alquiler en horas

Por tanto, los datos de entrada serán el tipo de vehículo a alquilar y el tiempo de alquiler. El tipo de vehículo puede codificarse de distintas formas: con números, con letras, etc. En este caso, se van a utilizar letras: P para el patín a pedales, T para el patín a motor, W para la tabla de windsurf y M para la moto acuática.

El dato de salida será el precio total a pagar, calculado según el tipo de vehículo a alquilar.

4. Escribir un programa que, dados los valores correspondientes a tres lados de un triángulo, nos dice si este triángulo es rectángulo o no, teniendo en cuenta que la siguiente relación se cumple en los triángulos rectángulos: $h^2 = c_1^2 + c_2^2$

Este problema se resuelve aplicando la expresión anterior:

1. suponiendo que el primer lado leído es la hipotenusa
2. suponiendo que el segundo lado leído es la hipotenusa
3. suponiendo que el tercer lado leído es la hipotenusa

Si en algún caso se cumple la relación, entonces el triángulo es rectángulo.

Por ejemplo, si se introducen los valores 3, 4 y 5 para cada lado, la relación se cumple si el tercer lado es la hipotenusa.

5. Escribir un programa que calcule el valor de la siguiente función matemática mostrando el resultado para un valor de x leído de teclado:

$$f(x) = \begin{cases} 2x^2 - x - 5 & \text{si } x < 0 \\ 5x^3 - 4x^2 & \text{si } 0 \leq x < 5 \\ 3x^5 - 3x^3 + x^2 & \text{si } 5 \leq x \end{cases}$$

6. Un supermercado tiene las siguientes ofertas sobre un tipo de producto: 3x2 o 30%. Para mejorar la atención al cliente, se desea disponer de un programa que dado el número de unidades que compra un cliente y el precio del producto, indique la oferta más conveniente y el total a pagar. Por ejemplo, si un producto cuesta 1€, la siguiente tabla muestra los totales resultantes aplicando ambas ofertas para diferentes cantidades de producto:

| Unidades | Total con la oferta 3x2 | Total con la oferta del 30% | Oferta más Conveniente |
|----------|-------------------------|-----------------------------|------------------------|
| 3 | 2€ | 2'10€ | 3x2 |
| 4 | 3€ | 2'80€ | 30% |
| 5 | 4€ | 3'50€ | 30% |
| 6 | 4€ | 4'20€ | 3x2 |

7. En una oficina de cambio sólo se intercambian dólares y euros. Escribir un programa que pregunte qué tipo de cambio se desea realizar, cuál es la cotización en ese momento y la cantidad a cambiar y escriba en pantalla la cantidad a entregar en la otra moneda.

8. Escribir un programa que calcule el valor de la siguiente función matemática mostrando el resultado para un valor de x leído de teclado:

$$f(x) = \begin{cases} x^2 - 2x - 5 & \text{si } x < 0 \\ x^3 - 3x^2 & \text{si } 0 < x < 3 \\ x^4 - 4x^3 + x^2 & \text{si } 3 \leq x < 5 \\ x^5 - 5x^4 + x^3 & \text{si } 5 \leq x \end{cases}$$

9. La fecha del domingo de Pascua corresponde al primer domingo después de la primera luna llena que sigue al equinoccio de primavera. Dado un año, los cálculos que permiten conocer esta fecha son:

A = año % 19

B = año % 4

C = año % 7

D = (19 * A + 24) % 7

E = (2 * B + 4 * C + 6 * D + 5) % 7

N = (22 + D + E)

donde N indica el número del día del mes de marzo (o abril si N es superior a 31) correspondiente al domingo de Pascua. Escribir un programa que lea un año y muestre el día y el mes en el que se celebró o se celebrará el domingo de Pascua ese año.

Por ejemplo: si introducimos el año 2002, los cálculos son los siguientes:

A = 2002 % 19 = 7

B = 2002 % 4 = 2

C = 2002 % 7 = 0

D = (19 * 7 + 24) % 7 = 157 % 7 = 3

E = (2 * 2 + 4 * 0 + 6 * 3 + 5) % 7 = 27 % 7 = 6

N = (22 + 3 + 6) = 31

Es decir, el año 2002 el domingo de Pascua fue el 31 de marzo.

10. Escribir un programa que aconseje qué prenda de abrigo debe llevar una persona basándose en datos de temperatura y humedad proporcionados por el usuario:

- Si la temperatura es menor de 20 grados o la humedad está dentro del rango 50-80 debe aconsejar plumífero.
- Si la temperatura es menor de 20 grados pero la humedad no supera el 50% debe aconsejar abrigo.
- Si la temperatura es mayor de 20 grados y la humedad es menor del 80% debe aconsejar chaqueta fina.
- Si la temperatura es mayor de 20 grados y la humedad supera el 80% debe aconsejar no coger ninguna prenda de abrigo.

11. El dueño de una papelería desea un programa que le indique el precio de venta de un artículo dado. Este precio se calcula de acuerdo con la siguiente fórmula: $PVP = \text{precio_coste} + \text{ganancia}$, donde la ganancia será:

- el 15% si el precio de coste no supera los 3€
- 50 céntimos si el precio de coste está entre 3 y 6€
- el 25% si el precio de coste supera los 6€

12. En una empresa cada empleado debe teclear un código identificador de 3 cifras en la entrada. Escribir un programa que muestre por pantalla la categoría del empleado teniendo en cuenta que:

- si el código es divisible por 2, por 3 y por 5, la categoría del empleado es "Director general"
- si el código es divisible por 3 y por 5 pero no por 2, la categoría del empleado es "Directivo"
- si el código es divisible por 2, pero no por 3 ni por 5, la categoría del empleado es "Staff"
- si el código no es divisible por 2, ni por 3 ni por 5, la categoría del empleado es "Seguridad"

13. La Dirección General de Tráfico necesita un programa que, dado el tipo de vehículo que conduce un conductor, su índice de alcohol y si ha sido medido en sangre o en aire, determine si da positivo en el control de alcoholemia o no. La tasa máxima de alcohol tanto en sangre como en aire según el tipo de vehículo se resume en la siguiente tabla:

| Vehículo | Indicador | Tasa máxima en sangre | Tasa máxima en aire |
|-------------|-----------|-----------------------|---------------------|
| Camión | C | 0.3 | 0.15 |
| Autobús | A | 0.3 | 0.15 |
| Turismo | T | 0.5 | 0.25 |
| Motocicleta | M | 0.3 | 0.15 |

14. Una empresa de transporte por autobús ofrece tres tarifas de viaje posibles, pero sólo una de ellas es aplicable:

- Grupal, si son 4 o más personas. En este caso, dos personas pagan el billete entero, y el resto tienen un descuento del 50%
- Ida y vuelta. El billete tiene un 10% de descuento
- Joven. El billete tiene un 25% de descuento

Se necesita un programa para gestionar estas tarifas. El usuario introduce la cantidad de gente que viene a comprar billete, cuántos de ellos son jóvenes, si el viaje es de ida y vuelta y el precio base del billete. Con estos datos, el programa debe realizar lo siguiente:

- calcular el precio total de todo el grupo según cada tipo de tarifa (si alguna no es aplicable, mostrar un mensaje indicándolo)
- recomendar la tarifa mínima

Por ejemplo, si el billete cuesta 10€ y hay 6 personas que compran el billete de ida y vuelta, siendo 2 de ellas jóvenes, el resultado será:

| Tarifa | Precio con descuento | Precio total |
|--------------|------------------------------------|-----------------------------------------|
| Grupal | $10 - 10 \cdot 0.5 = 5\text{€}$ | $2 \cdot 10 + 4 \cdot 5 = 40\text{€}$ |
| Ida y vuelta | $10 - 10 \cdot 0.1 = 9\text{€}$ | $6 \cdot 9 = 54\text{€}$ |
| Joven | $10 - 10 \cdot 0.25 = 7.5\text{€}$ | $4 \cdot 10 + 2 \cdot 7.5 = 55\text{€}$ |

En este caso, el programa deberá recomendar la tarifa grupal.

15. Se dice que tres números enteros positivos forman una terna pitagórica si se cumple que la suma del cuadrado de dos de ellos es igual al cuadrado del tercero. Escribir un programa que lea tres números enteros positivos a, b y c y determine si forman una terna pitagórica, es decir, si se cumple que:

- $a^2 + b^2 = c^2$ ó
- $a^2 + c^2 = b^2$ ó
- $b^2 + c^2 = a^2$

Ejercicios de repetición

1. Escribir un programa que lea el precio de una serie de coches y muestre el mayor y el menor de todos ellos, preguntando previamente al usuario cuántos coches hay.

Análisis

Este problema consiste en calcular el precio máximo y mínimo de un conjunto de coches. Para ello, es necesario leer el precio de cada uno de los coches de este conjunto y procesarlos de manera que se obtenga el mayor y el menor precio.

Dado que no es conocido el número de coches del conjunto (se debe preguntar al usuario), no es posible escribir un programa con tantas instrucciones `input` como sean necesarias para leer el precio de todos los coches. Por esta razón, se debe plantear una instrucción repetitiva que permita la lectura del precio de tantos coches como indique el usuario. Es decir, en primer lugar, habrá que preguntar al usuario cuántos coches hay, para después escribir un bucle que lea el precio de cada coche.

Además, hay que tener en cuenta que la lectura de los precios no puede hacerse en una variable distinta para cada coche, ya que a priori (es decir, mientras se está escribiendo el programa) no se sabe cuántos coches hay. Por ello, se debe utilizar la misma variable para la lectura del precio de todos los coches. Esto implica que, cuando se lea el precio del segundo coche, se borrará el precio leído del primero. Por tanto, es necesario procesar el precio de un coche antes de la lectura del siguiente. Es decir, se debe ir calculando el precio máximo y mínimo en el mismo bucle que se plantee para la lectura.

Así, el esquema del proceso a seguir es el siguiente:

1. Leer el número de coches a procesar en la variable `num_coches`
2. Leer el precio del primer coche en la variable `precio`
3. Asignar a la variable `minimo` el contenido de la variable `precio`
4. Repetir `num_coches - 1` veces:
 - a. Leer el precio del siguiente coche en la variable `precio`
 - b. Si `precio` es menor que `minimo`, el precio mínimo pasa a ser el nuevo precio leído, es decir, se debe asignar el contenido de la variable `precio` a la variable `minimo`.
5. Escribir en pantalla el contenido de la variable `minimo`, que será el precio mínimo de los leídos.

Con este proceso se calcula el precio mínimo. Sólo queda añadir la parte para calcular el precio máximo, que es análoga.

2. Escribir un programa que lea la edad de cada uno de los asistentes a un curso y calcule la media.

Análisis

Este problema consiste en leer la edad de los asistentes a un curso y calcular la media de estas edades, es decir, se deben sumar todas las edades leídas y dividir la suma resultante entre el número total de asistentes al curso.

Como el enunciado no indica cuántos alumnos hay, no es posible leer la edad de cada alumno en variables distintas y plantear una expresión que sume todas las edades. Sin embargo, es posible utilizar una única variable para la lectura, es decir, se utiliza la variable `edad` para leer

la edad de cada alumno y se intenta plantear una expresión que, al final del proceso de todos los datos, haga que la variable `suma_edades` contenga la suma de las edades de todos los alumnos. La suma de todos los alumnos será:

`Suma_edades = edad del primero + edad del segundo + edad del tercero + etc.`

Que, suponiendo que las siguientes expresiones se evalúan secuencialmente, es lo mismo que decir:

`Suma_edades = edad del primero`

`Suma_edades = Suma_edades + edad del segundo = edad del primero + edad del segundo`

`Suma_edades = Suma_edades + edad del tercero = edad del primero + edad del segundo + edad del tercero`

Etc.

Este punto es importante ya que hay que tener en cuenta que, si se utiliza la misma variable para leer la edad de cada asistente al curso, cuando se lea la edad del segundo asistente, se borrará la edad del primero. Esto implica que la edad de cada asistente debe acumularse en la variable `suma_edades` antes de la lectura de la edad del siguiente asistente. El proceso, teniendo en cuenta que la lectura de datos finaliza cuando se introduce una edad menor o igual que 0, quedaría de la siguiente forma:

- leer la edad del primer asistente en la variable `edad`
- si `edad > 0`
 - o `suma_edades = suma_edades + edad`
 - o leer la edad del segundo asistente en la variable `edad`
 - o si `edad > 0`
 - `suma_edades = suma_edades + edad`
 - leer la edad del tercer asistente en la variable `edad`
 - si `edad > 0`
 - `suma_edades = suma_edades + edad`
 - etc.

De esta forma, se podría leer la edad de cada asistente y al final, tener la suma de todas ellas en la variable `suma_edades`. Sin embargo, tampoco sería posible escribir un programa así, porque no se sabe cuántos asistentes hay y por tanto, no se sabe cuántas lecturas serían necesarias. Es decir, en el esquema anterior (sin tener en cuenta el etc.), si hay dos alumnos, funciona bien porque cuando se lee la edad del tercer asistente, ésta es negativa o cero y por tanto ya no se sigue; pero si hubiera cinco alumnos, ya no funcionaría porque leería el tercero y finalizaría.

Por tanto, se debe plantear una instrucción repetitiva que procese cada alumno. Cuando se utilizan instrucciones repetitivas, se debe determinar el proceso a repetir y la condición de continuación. En este caso, la condición de continuación viene dada por el enunciado, ya que dice que se deben leer edades hasta que la edad introducida sea menor o igual que 0, es decir, la condición de continuación es que la edad introducida sea mayor que 0.

En cuanto al proceso, se debe analizar en el esquema anterior cuál es la parte que se repite, que es:

- `suma_edades = suma_edades + edad`
- leer la edad del siguiente asistente en la variable `edad`

Por tanto, el esquema general del proceso de lectura y suma de edades será:

- `suma_edades=0`
- leer la edad del primer asistente en la variable `edad`
- mientras `edad > 0`
 - o `suma_edades = suma_edades + edad`

- leer la edad del siguiente asistente en la variable `edad`

Por último, una vez termina la instrucción repetitiva, se debe calcular la media de las edades, para lo cual se necesita la suma de las edades (ya calculada) y el número total de asistentes, que no se sabe cuál es. Esto implica que, al mismo tiempo que se leen y suman las edades, se debe contar cuántas edades correctas se han introducido y almacenarlo en la variable `num_asistentes`. El esquema general del programa quedará como:

- `num_asistentes=0`
- `suma_edades=0`
- leer la edad del primer asistente en la variable `edad`
- mientras `edad > 0`
 - `suma_edades = suma_edades + edad`
 - `num_asistentes = num_asistentes + 1`
 - leer la edad del siguiente asistente en la variable `edad`
 - `media = suma_edades / num_asistentes`
- escribir en pantalla el contenido de la variable `media`

3. Escribir un programa que determine si un número entero dado es primo o no.

Análisis

Un número es primo cuando sólo es divisible por 1 y por sí mismo. Es decir, un número a será primo si no existe otro número b entre 1 y a (no incluidos) tal que a sea divisible por b .

El dato de entrada a este programa será el número del cual se desea saber si es primo y la salida será un mensaje que indique si el número leído es primo o no. Así, el proceso que resolverá este problema consiste dividir el número leído entre todos los números entre 2 y el número $- 1$; si se encuentra algún número que es divisor del número leído, entonces éste ya no es primo. Se puede utilizar una variable `es_primo` para indicar si el número es primo o no. Inicialmente se asigna el valor 1 (verdadero) a la variable `es_primo` y si se encuentra algún número que es divisor del número leído, se asigna el valor 0 (falso) a la variable `es_primo`. Se puede esquematizar este proceso de la siguiente manera:

1. Leer un número y almacenarlo en la variable `numero`
2. Asignar a la variable `divisor` el valor 2
3. Asignar el valor 1 a la variable `es_primo`
4. Mientras el contenido de la variable `es_primo` es 1 y el contenido de la variable `divisor` sea distinto del contenido de la variable `numero`:
 - a. Si `divisor` es divisor de `numero`, asignar el valor 0 a la variable `es_primo`
 - b. Incrementar el contenido de la variable `divisor`
5. Si la variable `es_primo` vale 1, escribir por pantalla: El número es primo
6. Si la variable `es_primo` vale 0, escribir por pantalla: El número no es primo

4. Escribir un programa que calcule la aproximación de e^x mediante el desarrollo de la siguiente serie de Taylor:

$$e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!}$$

- Realizar el cálculo para un número N de sumandos indicado por el usuario.
- Reescribir el programa considerando que la aproximación es válida cuando el último sumando calculado es menor que un error E proporcionado por el usuario.

Apartado (a)

Análisis

Este problema consiste en calcular e^x mediante la aproximación de la serie de Taylor, desarrollando los N primeros sumandos de la misma. Estos N primeros sumandos se pueden desarrollar como:

$$e^x = \sum_{i=0}^{N-1} \frac{x^i}{i!} = \frac{x^0}{0!} + \frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots + \frac{x^{N-1}}{(N-1)!} =$$

$$= \frac{1}{1} + \frac{x}{1} + \frac{x*x}{1*2} + \frac{x*x*x}{1*2*3} + \frac{x*x*x*x}{1*2*3*4} + \dots$$

En este problema, al igual que en problemas anteriores, es necesario plantear una estructura repetitiva para calcular esta serie, ya que se deben utilizar los N primeros sumandos y N es un valor proporcionado por el usuario. Por tanto, no es posible escribir una expresión que efectúe este cálculo.

Al tratarse de un sumatorio, se debe plantear una expresión que acumule el valor de cada uno de los sumandos. La expresión anterior puede reescribirse como sigue:

$$e_x = 1$$

$$e_x = e_x + \frac{x}{1} = 1 + \frac{x}{1}$$

$$e_x = e_x + \frac{x*x}{1*2} = 1 + \frac{x}{1} + \frac{x*x}{1*2}$$

$$e_x = e_x + \frac{x*x*x}{1*2*3} = 1 + \frac{x}{1} + \frac{x*x}{1*2} + \frac{x*x*x}{1*2*3}$$

$$e_x = e_x + \frac{x*x*x*x}{1*2*3*4} = 1 + \frac{x}{1} + \frac{x*x}{1*2} + \frac{x*x*x}{1*2*3} + \frac{x*x*x*x}{1*2*3*4}$$

...

Por tanto, la expresión que se debe repetir en la estructura repetitiva será:

$$e_x = e_x + \frac{x^i}{i!}$$

Cada sumando es un cálculo complejo, ya que supone el cálculo de una potencia y de un factorial, operaciones que requieren el uso de otras estructuras repetitivas. Sin embargo, si se

observa el desarrollo en serie, es posible calcular cada sumando a partir del anterior. Es decir, dejando aparte el primer sumando cuyo valor es 1, el resto de sumandos se forman de la siguiente forma:

- el numerador es el numerador del sumando anterior multiplicado por x
- el denominador es el denominador del sumando anterior multiplicado por el lugar que ocupa el nuevo sumando en la serie, es decir, por i .

Así, el esquema del programa quedará como sigue:

1. Leer el valor de x y de N y almacenarlos en las variables x y N , respectivamente
2. Asignar el valor 1 a las variables i , suma , $x_elevado_i$ y factorial_i , que almacenan el número del siguiente sumando a calcular, el sumatorio, el numerador y el denominador de la expresión, respectivamente
3. Repetir $N-1$ veces:
 - a. Multiplicar $x_elevado_i$ por x
 - b. Multiplicar factorial_i por i
 - c. Acumular a la variable suma el resultado de $x_elevado_i$ dividido por factorial_i
 - d. Incrementar el valor de i
4. Mostrar el contenido de la variable suma por pantalla.

Apartado (b)

Este problema se resuelve de forma muy similar al anterior. Básicamente, se diferencian en la condición de continuación del bucle: mientras en el problema anterior el bucle se repetía un número de veces determinado por el usuario, en este problema el usuario indica el error máximo a cometer y el bucle continúa mientras el último sumando calculado es superior a este error máximo.

Así, el siguiente programa se diferencia del anterior en:

- la lectura del error máximo en lugar de la variable N
- la introducción de una nueva variable sumando , que almacenará el resultado del sumando actual
- el uso de la variable sumando en la condición del bucle `while`

5. Escribir un programa que lea un número entero y escriba por pantalla su factorización. Por ejemplo, $300 = 2 * 2 * 3 * 5 * 5$
6. Escribir un programa que calcule la raíz cuadrada de un número usando el método de Newton de aproximación por intervalos. Este método funciona restringiendo la amplitud el intervalo de trabajo $[i, s]$ hasta que se encuentra la solución. Inicialmente se toma como intervalo de trabajo $[0, x]$ siendo x el número del cual se desea calcular la raíz cuadrada. Se calcula el punto medio m del intervalo de trabajo $[i, s]$. Si m^2 es mayor que x , el nuevo intervalo será $[i, m]$. En caso contrario, el nuevo intervalo será $[m, s]$. Se repite este proceso hasta que la diferencia en valor absoluto entre x y m^2 es menor que un error máximo permitido especificado por el usuario.

Por ejemplo, si x es 9 y el error es 0.01, el proceso funcionaría de la siguiente forma:

| Iteración | Límite inferior | Límite superior | Punto medio | Diferencia |
|-----------|-----------------|-----------------|-------------|------------|
| inicio | 0 | 9 | 4.5 | 11.25 |
| 1 | 0 | 4.5 | 2.25 | 3.9375 |
| 2 | 2.25 | 4.5 | 3.375 | 2.3906 |
| 3 | 2.25 | 3.375 | 2.8125 | 1.0898 |
| 4 | 2.8125 | 3.375 | 3.0937 | 0.5713 |
| 5 | 2.8125 | 3.0937 | 2.9531 | 0.279 |
| 6 | 2.9531 | 3.0937 | 3.0234 | 0.1412 |
| 7 | 2.9531 | 3.0234 | 2.9883 | 0.0702 |
| 8 | 2.9883 | 3.0234 | 3.0058 | 0.0352 |
| 9 | 2.9883 | 3.0058 | 2.9971 | 0.0176 |
| 10 | 2.9971 | 3.0058 | 3.0015 | 0.0088 |

7. Escribir un programa que, dado un número entero, imprime todos sus divisores o el mensaje "El número es primo" si no tiene divisores.
8. Escribir un programa que lea las edades de un grupo de personas hasta que se introduce un número negativo y calcule la media, la edad máxima y la mínima y el número de personas que están jubiladas (su edad es mayor de 65 años) y cuántos son menores de edad.
9. Leonardo Fibonacci, matemático italiano del siglo XII, planteó en su obra Liber Albaci un problema que consistía en averiguar, a partir de una pareja de conejos, cuántas parejas nacerían en un lugar cerrado en el transcurso de un año, obteniendo como resultado la serie de números:
1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

Escribir un programa que muestre la serie de Fibonacci hasta un límite dado por el usuario, sabiendo que esta serie se define como:

$$F(0)=1 \quad F(1)=1 \quad F(i)=F(i-1)+F(i-2)$$

10. La compañía eléctrica de una ciudad desea mejorar su servicio informando al cliente sobre la media y la varianza del consumo de electricidad que realiza.

Escribir un programa que lea el consumo de un cliente en los últimos N meses y realizar estos cálculos utilizando las siguientes formulas:

$$\text{varianza} = \frac{1}{n} \sum_{i=1}^n x_i^2 - \frac{1}{n^2} \left(\sum_{i=1}^n x_i \right)^2$$

$$\text{media} = \frac{1}{n} \sum_{i=1}^n x_i$$

11. Escribir un programa que calcule el valor final de la siguiente serie armónica, donde n es un valor entero distinto de cero especificado por el usuario:

$$1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$$

12. Se desea realizar una estadística de las precipitaciones recogidas en una estación hídrica en un año. Para ello, escribir un programa que lea la cantidad mensual de litros por metro cuadrado recogidos en dicha estación y mostrar:

- a) mes que más ha llovido
- b) mes que menos ha llovido
- c) media de las lluvias recogidas
- d) porcentaje de los meses que ha llovido menos de 10 l/m²
- e) porcentaje de los meses que ha llovido más de 100 l/m²
- f) porcentaje de los meses que ha llovido entre 20 y 50 l/m²

13. Según Pitágoras, un número perfecto es un entero positivo que es igual a la suma de todos los enteros positivos (excluido él mismo) que son divisores del número. El primer número perfecto es 6, ya que los divisores de 6 son 1, 2, 3 y $1 + 2 + 3 = 6$. Escribir un programa que encuentre los tres primeros números perfectos.

14. Escribir un programa que lea un número entero y diga si es capicúa.

15. Escribir un programa que calcule la suma de los 100 primeros términos de la siguiente sucesión:

$$a_n = (-1)^n \frac{n^2 - 1}{2n}$$

16. En unos grandes almacenes han detectado un fraude que consiste en el uso de tarjetas de crédito falsas. Para evitarlo, necesitan un programa que verifique si el número de una tarjeta de crédito es correcto. Los pasos que hay que seguir son los siguientes:

- a. Multiplicar cada dígito por su peso (los pesos se alternan 1, 2, 1, 2, ...). Cuando el número de dígitos es par la serie de pesos comienza con un 2, en caso contrario empieza con 1.
- b. Si el resultado de multiplicar cada dígito por su peso es mayor que 9, se le resta 9.
- c. Sumar todos los valores obtenidos en el paso b) y calcular el módulo entre 10.
- d. Si el resultado es 0 el número de tarjeta es correcto, en caso contrario no lo es.

Ejemplo:

| | | | | | | | | | | | | | | | | |
|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|
| tarjeta | 3 | 7 | 0 | 5 | 0 | 9 | 2 | 3 | 7 | 3 | 3 | 7 | 2 | 0 | 3 | 5 |
| pesos | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 |
| producto | 6 | 7 | 0 | 5 | 0 | 9 | 4 | 3 | 14 | 3 | 6 | 7 | 4 | 0 | 6 | 5 |
| | | | | | | | | | -9 | | | | | | | |
| | 6+ | 7+ | 0+ | 5+ | 0+ | 9+ | 4+ | 3+ | 5+ | 3+ | 6+ | 7+ | 4+ | 0+ | 6+ | 5 =70 |

El resto de dividir 70 por 10 es cero, luego el número de tarjeta es correcto.

17. Juan va a pedir un préstamo y desea conocer cuál será la anualidad que deberá pagar, teniendo en cuenta el capital (C) que va a solicitar y las condiciones del banco: interés anual (I) y los años de amortización (N). Escribir un programa que permita realizar el cálculo de la anualidad para cada uno de los años de amortización mediante la siguiente fórmula:

$$anualidad = \frac{C * \left(1 + \frac{I}{100}\right)^N * \frac{I}{100}}{\left(1 + \frac{I}{100}\right)^N - 1}$$

El programa también deberá calcular para cada año, la parte de la anualidad dedicada al pago de intereses (PI) y la parte dedicada a la amortización de la deuda (A), según las siguientes expresiones:

$$A_1 = anualidad - C * \frac{I}{100}$$

$$A_k = A_1 * \left(1 + \frac{I}{100}\right)^{k-1}$$

18. Escribir un programa que calcule el valor de la siguiente serie, donde n es un valor entero distinto de cero introducido por el usuario:

$$\frac{1}{2} + \frac{2}{2^2} + \frac{3}{2^3} + \dots + \frac{n}{2^n}$$

19. Escribir un programa que calcule el centro de masas de un sistema de n masas puntuales. Para ello el programa pedirá conjuntos de tres números reales: la coordenada x del punto, la coordenada y y su masa m . El número n de masas no estará limitado de antemano, sino que después de la introducción de los datos de una masa se preguntará al usuario si desea introducir más. El programa imprimirá por pantalla como resultado las coordenadas xg e yg del centro de masas del conjunto de puntos que se han introducido. Las fórmulas que dan la posición del centro de gravedad son las siguientes:

$$xg = \frac{\sum_{i=1}^n m_i x_i}{\sum_{i=1}^n m_i} \quad yg = \frac{\sum_{i=1}^n m_i y_i}{\sum_{i=1}^n m_i}$$

20. Según el teorema de Bolzano, si una función continua tiene valores de distinto signo en los extremos a y b de un intervalo, la función tiene al menos un cero (una raíz) en ese intervalo. Este teorema se utilizará como base para un método de cálculo de raíces de funciones, llamado Método de la Bisección. Se parte de un par de puntos a y b en los que se cumple la condición $f(a)*f(b)<0$ (la función tiene signos opuestos, luego existe al menos una raíz en $[a, b]$). A continuación se estudia la función en el punto c , que es el punto medio del intervalo ($c=(a+b)/2$), y se estudia el signo de $f(c)$. Si $f(c)$ tiene el mismo signo que $f(a)$ se hace $a=c$ y se vuelve a comenzar (se tiene un nuevo intervalo $[a, b]$ en el que está la raíz y que es la mitad que el anterior); si $f(c)$ tiene el mismo signo que $f(b)$ se hace $b=c$ y se vuelve a comenzar. El proceso prosigue con intervalos $[a, b]$ cada vez más pequeños hasta que se cumple una de las siguientes condiciones:

- $f(c)=0$, por tanto, c es la solución buscada
- $f(c) < \text{eps1}$, por tanto, c es la solución aproximada buscada
- $b-a < \text{eps2}$, por tanto, $c=(b-a)/2$ es la solución aproximada buscada

Escribir un programa que aplique este método para encontrar la raíz de la función $f(x)=3x^2 - 0.5$ en el intervalo $[0, 3]$, con valores de $\text{eps1}=10^{-3}$ y $\text{eps2}=10^{-5}$. El resultado es 0.408.

21. El Instituto Americano de Construcción en Acero especifica las siguientes fórmulas para evaluar el esfuerzo permisible en columnas secundarias.

$$s = 17000 - 0.485 (l / r)^2, \text{ donde } l / r \leq 120 \text{ y}$$

$$s = 18000 / (1 + l^2 / (18000 r^2)), \text{ donde } 120 \leq l / r < 200$$

Variables:

$$s = \text{Esfuerzo unitario (lb/ pulg}^2\text{)}$$

$$l = \text{Longitud de la columna (pulg)}$$

$$r = \text{Radio de giro de la sección (pulg)}$$

$$l / r = \text{Razón de esbeltez}$$

Escribir un programa que lea los valores l y r , seleccione la fórmula adecuada y calcule el esfuerzo permisible de tensión. Si la razón de esbeltez es 200 o más debe imprimirse "Esbeltez demasiado alta. Por favor aumenta la sección" y repita la operación con un nuevo valor de r .

22. Una empresa de alimentación codifica el número de lote de fabricación de sus productos con un número de 6 dígitos que indican lo siguiente:

- primer dígito: año de fabricación, por ejemplo: 0 para el 2000, 2 para el 2002
- segundo dígito: mes de fabricación, de enero (1) a octubre (0)
- tercer dígito: semana dentro del mes (1, 2, 3 ó 4)
- cuarto dígito: día de la semana (de 1 a 7)
- quinto dígito: mañana o tarde (1 ó 2)
- sexto dígito: número de lote de esa mañana o tarde (de 1 a 9)

Con estos datos, escribir un programa que lea un número de seis dígitos (de una sola vez, no dígito a dígito) y un número de 1 a 6 indicando la posición a leer y muestre por pantalla el dígito que ocupa esta posición y su significado.

Por ejemplo: si se introduce el número 243428 y la posición 2, el programa deberá escribir: 4 – abril.

23. Juan compra todos los días de la semana el cupón de la ONCE y quiere hacer un programa que, dado el número de su cupón y el número que salió para cada uno de los días de la semana, le diga cuánto dinero ha ganado. Los premios son los siguientes:

- 100.000 euros: cinco cifras y serie
- 12.000 euros: cinco cifras
- 150 euros: cuatro últimas cifras
- 15 euros: tres últimas cifras
- 2 euros: dos últimas cifras
- 1 euro: última cifra
- 1 euro: primera cifra
- 1 euro: tercera cifra

Además, teniendo en cuenta que el cupón le cuesta 1 euro cada día, indicar el dinero total que ha ganado, es decir, restar del total de premios el coste de cada cupón.

24. Escribir un programa que muestre un menú con estas opciones:

- a) calcular la aproximación del seno de x
- b) calcular la aproximación del coseno de x
- c) terminar

El programa leerá la opción elegida por el usuario, comprobando que es correcta y realizará el cálculo solicitado. Las aproximaciones pueden calcularse con los siguientes desarrollos de Taylor:

$$\text{sen}(x) = \sum_{i=0}^{\infty} \frac{(-1)^i x^{2i+1}}{(2i+1)!}$$

$$\text{cos}(x) = \sum_{i=0}^{\infty} \frac{(-1)^i x^{2i}}{(2i)!}$$

En ambos casos consideraremos que la aproximación es válida cuando el último sumando calculado sea menor que un error E proporcionado por el usuario.

25. La Dirección General de Tráfico está realizando unas estadísticas acerca de los controles de alcoholemia. Para ello, se necesita un programa que lea los valores correspondientes al tipo de vehículo (según la siguiente tabla) y al índice de alcohol en sangre de los controles realizados el último sábado por la noche.

| Vehículo | Indicador | Tasa máxima |
|-------------|-----------|-------------|
| Camión | C | 0.3 |
| Autobús | A | 0.3 |
| Turismo | T | 0.5 |
| Motocicleta | M | 0.3 |

El programa mostrará un informe con:

- a) el total de conductores que fueron sometidos al control, clasificados por el tipo de vehículo que conducían
- b) el número de conductores que dieron positivo según el tipo de vehículo que conducían. Los valores máximos para cada vehículo se muestran en la tabla anterior.
- c) cuál fue la media de la tasa de alcohol en sangre de los conductores que dieron positivo

26. Dada una secuencia de números enteros acabada en 0, obtener la suma de aquellos números tales que su número de cifras sea igual a la suma de las mismas.

Ejemplo: 1 111 201 303 0

Suma = 1+111+201

27. Dado un conjunto de N resistencias eléctricas, la resistencia total que producen si se conectan en paralelo o en serie viene dada por las siguientes expresiones:

$$R_p = \sum_{i=1}^N \frac{1}{R_i}$$

$$R_s = \sum_{i=1}^N R_i$$

Escribir un programa que lea los valores de cada una de las N resistencias y a continuación pregunte si se conectan en serie o en paralelo. Por último, debe mostrar el resultado correspondiente y preguntar si se desea realizar otro cálculo.