Ejercicios sobre funciones
Programación
Depto. Sistemas Informáticos y Computación
Facultad de Estudios Estadísticos, U. Complutense de Madrid

---

1. Escribe una función que halle la suma de dos números enteros, y un programa que muestre su funcionamiento.

2. Escribe funciones para los siguientes cálculos, junto con programas que muestren su funcionamiento:

   (a) El máximo de dos números reales.

   (b) El área de un triángulo, conocidas su base y su altura.

   (c) El área de un triángulo equilátero a partir de su lado.

   (d) El número de días de un mes (con 28 para el mes 2, que es febrero).

   (e) La raíz de un número real positivo, de índice arbitrario.

   (f) El logaritmo de un número en una base arbitraria. Expresa los requisitos que debe cumplir los datos de dicha función.

   (g) Una función para hallar el logaritmo de un número, en base $\sqrt[12]{2}$. (Esta función calcula el intervalo (en semitonos) correpondiente al conciente entre dos frecuencias. Llamemos *prony* a esta función en honor a Garpard de Prony, que investigó la relación entre frecuencias e intervalos.)

   Entre las funciones anteriores, algunas exigen requisitos sobre sus datos. Documéntalas adecuadamente. Los programas que las usan, los requisitos precisos no se pueden ignorar, así que se presentan dos opciones: o bien se declaran los requisitos adecuadamente o bien se han de comprobar.

3. Define la función $armonica : \mathcal{N} \to \mathcal{R}$, que sume unos cuantos términos de la serie armónica, como se describe a continuación:

$$armonica(n) = 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n}$$

4. Define la función `diaSiguiente` que halle el día de la semana, siguiente a uno dado,

   (a) Representando los días de la semana como los enteros {`1, ..., 7`}.

   (b) Representando los días de la semana como los caracteres {`'L', 'M', 'X', 'J', 'V', 'S', 'D'`}.

5. Define sendas funciones que averigüen si un carácter...

   (a) ...es una letra minúscula o no.

   (b) ...es una letra mayúscula o no.

   (c) ...es una letra o no, haciendo uso de las funciones anteriores.

   Hazlo sin usar las funciones `isupper`, `islower` e `isalpha`, de la biblioteca `cctype`. Luego, aplícalas en un programa que las usa para examinar los caracteres de una línea dada en la entrada estándar.

6. Escribe funciones para hallar las siguientes cantidades:
   (a) Las cifras que tiene un entero.
   (b) La cifra k-ésima de un entero, siendo la de las unidades la 0-ésima.
   (c) La suma de las cifras de un entero. (Esta cantidad se conoce como *raíz digital*.)
7. Funciones factorial, producto (desde... hasta...) y número combinatorio, lo más eficientemente que sea posible.
8. Escribe un subprograma que halle el máximo común divisor de dos números enteros. Para ello, se pueden usar los métodos de Nicómaco (o de las diferencias) y el de Euclides (usando el resto de la división entera). También se puede descender desde uno de los enteros hasta llegar a un divisor común.
   (a) ¿Qué requisitos deben exigirse a los datos para poder garantizar que los subprogramas definidos pararán?
   (b) Pruébalos para distintos pares de enteros y compara la eficiencia de los mismos.
9. Función que da la letra del día de la semana, sabiendo su número: $1 \to' L'$, etc. Ídem a la inversa.
10. **Comparación de reales.** Escribe funciones para ver si dos números reales pueden considerarse *iguales*:
    (a) **absoluto** Si su diferencia es menor o igual que una cierta cantidad ϵ (constante de la función).
    (b) **relativo** $\frac{|x-y|}{|x|+|y|} \leq \epsilon$.
11. **Otras funciones booleanas.** Escribe funciones para averiguar las siguientes propiedades:
    (a) Si un número es par o no.
    (b) Si un número es primo o no.
    (c) Si dos números son primos entre sí.
    (d) Si un número es *perfecto* o no. Un número es *perfecto* si la suma de sus divisores (excluido él mismo) es igual al propio número. Si lo deseas, puedes hacer uso de una función auxiliar que sume las cifras de dicho número, definida en el ejercicio anterior.
12. Desarrolla un programa que busque el primer número perfecto a partir de un cierto entero dado por el usuario.
13. Desarrolla un programa que escriba todos los primos del 1 al 1000.
14. Suponiendo ya definida una función $f : \mathcal{R} \to \mathcal{R}$,
    (a) Define una función que halle su derivada en un punto.
    (b) Define una función que halle su integral definida.
    (c) Suponiendo que f es continua en $[a, b]$, negativa en $a \in \mathcal{R}$ y positiva en $b \in \mathcal{R}$, define una función que halle un cero de f en $[a, b]$.
    (d) El método anterior usa bipartición. Diseña ahora uno para trabajar con Newton-Raphson.

(e) Usando las funciones que te parezca conveniente, halla la cantidad $\sqrt[3]{2}$ como un cero de la función f, tal que $fx = x^2$.

(f) Idem. para hallar el valor de $arcsen(\sqrt{2}/3)$.

---

15. Define el subprograma `escribirFecha` que, aplicado a los datos `'D'`, `18`, `9` y `1960`, ponga en la pantalla lo siguiente:

```
Domingo, 18 de septiembre de 1.960
```

16. Define el subprograma `leerEnteroPositivo`, que lee un número entero e insiste hasta lograr que sea positivo estrictamente. Aplícalo en un programa que halla el m.c.d. y el m.c.m. de dos númeors naturales.

17. Define el subprograma `repetirCaracter`, que escribe un carácter dado tantas veces como se indique.

    (a) Con él, rehaz los programas ya planteados en la hoja de problemas anterior, para dibujar distintas figuras a base de asteriscos.

    (b) Escribe también un programa que dibuje la siguiente figura, consistente en n filas, donde la fila j es la secuencia de 2^j grupos formados cada uno por 2^{n-j-1} blancos y el mismo número de estrellas:

    (c) Ídem:

18. (a) Escribe un subprograma `pasaPasa` que manipule dos números enteros suprimiendo la última cifra del primero y añadiéndola al final del segundo. (b) Incluye ese procedimiento en un programa que invierta un número `num` (partiendo del propio `num` y de otro, con valor inicial cero),

$$(12345, 0) \to (1234, 5) \to (123, 54) \to (12, 543) \to (1, 5432) \to (0, 54321)$$

    a base de repetir la operación `pasaPasa` cuantas veces sea preciso.

19. Desarrolla un subprograma con el siguiente perfil,

```
void quitarDivisor(int & dividendo, int const divisor)
```

    que divida al dividendo (`ddo`) por el divisor (`dsor`) cuantas veces sea posible, dando la línea de la descomposición correspondiente de la manera usual:

$$\textit{dividendo} \mid \textit{divisor}$$

    Usa el procedimiento descrito en un programa que realice la descomposición de un número en sus factores primos.

```
void quitarDivisor(int & dividendo, int const divisor, int & numVeces)
```