Programación I: Ejercicios de clase

Facultad de Estudios Estadísticos
Universidad Complutense de Madrid
Curso 2021-2022

1. Identificadores, variables y expresiones

Cuadro 1: Precedencia y asociatividad de operadores

Operador	**Asociatividad**	**Prioridad**
( ) [ ]	Izquierda a derecha	1
− (cambio de signo) ++ −− !	Derecha a izquierda	2
∗ / %	Izquierda a derecha	3
+ −	Izquierda a derecha	4
< <= > >=	Izquierda a derecha	5
== !=	Izquierda a derecha	6
&&	Izquierda a derecha	7
||	Izquierda a derecha	8
= ∗= /= %= += −=	Derecha a izquierda	9

Ejercicio 1 Determina el valor de las expresiones propuestas, considerando las inicializaciones siguientes.
`int x=7; int y=11; float z=30.3; int w=30; bool esPar=true; bool esImpar=!esPar`

1. `x-x*y+y-z/3` = `(x-(x*y)+y-(z/3))`
2. `x&&y&&z%2` = `(x && y && (z%2))`
3. `x&&y&&w%2` = `(x && y && (w%2))`
4. `x>0||y+=1` = `((x>0)|| y)+=1`
5. `x>5&&y%3>1&&z>33` = `((x>5)&&(y%3>1)&&(z>33))`
6. `x>0&&w%2==0||y<0` = `(((x>0)&&(w%2==0))||(y<0))`
7. `x<=0||w%2!=0&&y>=0` = `((x<=0)||((w%2!=0)&&(y>=0)))`
8. `esPar&&!esImpar` = `(esPar&&(!esImpar))`
9. `!esImpar||x+y+z+w>100` = `((!esImpar)||((x+y+z+w)>100))`
10. `!(esImpar||x+y+z+w>100)` = `!(esImpar||((x+y+z+w)>100))`

Ejercicio 2 Diseña y escribe un programa que, dados el radio r y la altura h de un cono recto circular, calcule su volumen utilizando la fórmula $V = \frac{\pi r^2 h}{3}$.

Ejercicio 3 Diseña y escribe un programa que, dados el radio r y la altura h de un cilindro recto circular, calcule su área y volumen utilizando las fórmulas $A = 2\pi r(h + r)$ y $V = \pi r^2 h$.

Ejercicio 4 Diseña y escribe un programa que, dado un número positivo n de tres dígitos, decida si es o no capicúa. Por ejemplo, el número $n = 313$ es capicúa y el número $n = 314$ no es capicúa.

Ejercicio 5 Diseña y escribe un programa que, dado un número positivo n de tres dígitos, determine el número que se obtiene al invertir los dígitos de n. Por ejemplo, el resultado de invertir $n = 937$ es $m = 739$.

2. Instrucciones condicionales

Ejercicio 6 Escribe una asignación equivalente al fragmento de código siguiente.

```
bool trabajoRelevante;
bool aportacionNovedosa;
bool candidatoNobel;
if (trabajoRelevante)
  if (aportacionNovedosa)
    candidatoNobel=true;
  else
    candidatoNobel=false;
else
  candidatoNobel=false;
```

Ejercicio 7 [Suma de cubos] Diseña y escribe un programa que, dado un número natural de tres dígitos n, decida si n es igual o no a la suma del cubo de sus cifras.

Ejercicio 8 [Conversión de monedas] Diseña y escribe un programa que convierta una cantidad $c \geq 0$ expresada en euros en libras esterlinas y viceversa. Los datos de entrada del programa son la cantidad c a convertir y si esta está expresada en euros o libras. Las equivalencias entre monedas se deben declarar como constantes. Los datos de salida serán la cantidad obtenida tras la conversión y la equivalencia empleada.

Ejercicio 9 [Ecuación de primer grado] Diseña y escribe un programa que calcule la solución de una ecuación de primer grado $ax+b=0$. Los datos de entrada del programa son los coeficientes de la ecuación, a y b. El dato de salida será la raíz real de la ecuación.

Ejercicio 10 [Ecuación de segundo grado] Diseña y escribe un programa que calcule las soluciones de una ecuación de segundo grado $ax^2+bx+c=0$. Los datos de entrada del programa son los coeficientes de la ecuación, a, b y c. Los datos de salida serán las raíces reales de la ecuación.

Ejercicio 11 [Corrección de fechas] Diseña y escribe un programa que, dados un día y un mes, decida si la fecha introducida es o no válida. Por ejemplo, 31 de septiembre no es una fecha válida. Puedes asumir que el mes de febrero tiene siempre 28 días.

3. Instrucciones iterativas

Ejercicio 12 [Suma de cubos] Diseña y escribe un programa que busque todos los números de tres dígitos para los que la suma del cubo de sus dígitos iguale al propio número. Por ejemplo, $153 = 1^3+5^3+3^3$.

Ejercicio 13 [Suma de cifras] Diseña y escribe un programa que busque y cuente todos los números de cuatro cifras tales que la suma de la unidad y la centena sea mayor que la suma de las restantes cifras.

Ejercicio 14 [Máximo común divisor] Diseña y escribe un programa que, dados dos números naturales a y b, calcule su mcd siguiendo las reglas que te damos a continuación.

$$\mathtt{mcd}(\mathtt{a},\mathtt{b}) = \begin{cases} \mathtt{mcd}(\mathtt{a}-\mathtt{b},\mathtt{b}) & \text{si } \mathtt{a} > \mathtt{b} \\ \mathtt{mcd}(\mathtt{a},\mathtt{b}-\mathtt{a}) & \text{si } \mathtt{a} < \mathtt{b} \\ \mathtt{a} & \text{si } \mathtt{a} = \mathtt{b} \end{cases}$$

Ejercicio 15 [Número rectangular] Se dice que un número natural n es **rectangular** si es el producto de dos números distintos $n_1, n_2 \geq 2$. Diseña y escribe un programa que, dado un número natural n, decida si es o no rectangular.

Ejercicio 16 [Número friki] Se dice que un número natural n es **friki** si no es rectangular, su anterior es un cuadrado perfecto y su posterior es rectangular. Por ejemplo, 5 y 17 son frikis y 10 y 11 no lo son. Diseña y escribe un programa que muestre los 100 primeros números frikis. Cuida la eficiencia en tiempo del algoritmo, evitando en lo posible cálculos innecesarios (no es buena idea ir examinando los naturales uno a uno).

Ejercicio 17 [Números amigos] Se dice que dos números naturales n y m son **amigos** si la suma de los divisores de n comprendidos entre 1 y n–1 es igual a m y viceversa. Por ejemplo, 220 y 284 son amigos. Diseña y escribe un programa que, dados dos números naturales n y m, decida si son o no amigos.

Ejercicio 18 [Primalidad] Considera las propiedades siguientes sobre primalidad.

1. Un número n superior a 2 solo puede ser primo si es impar.
2. Un número n superior a 3 solo puede ser primo si verifica la propiedad $n^2 \,\%24 = 1$.
3. Un número entero positivo n es primo si y solo si no tiene divisores entre 2 y $\lfloor\sqrt{n}\rfloor$.

Diseña y escribe un programa que, dado un número entero positivo n, decida si es o no primo, comprobando si es 2 o impar; en el último caso, si cumple la propiedad $n^2 \,\%24 = 1$; y solamente cuando sí, busque algún divisor entre los indicados en la última propiedad.

Ejercicio 19 [Primalidad melliza] Se dice que dos números naturales n y m son **primos mellizos** si ambos son primos y difieren en dos unidades. Diseña y escribe un programa que, dada una cota $c \geq 0$, calcule la cantidad de pares de primos mellizos formados por números menores o iguales que c. Por ejemplo, para $c = 15$ hay tres pares de primos mellizos, que son $(3, 5)$, $(5, 7)$ y $(11, 13)$. Cuida la eficiencia en tiempo del algoritmo, evitando en lo posible cálculos innecesarios (no es buena idea ir examinando los naturales uno a uno).

Ejercicio 20 [Número primo-rectangular] Se dice que un número natural n es **primo-rectangular** si es el producto de dos números primos distintos $n_1, n_2 \geq 2$. Diseña y escribe un programa que, dada una cota $c \geq 0$, obtenga todos los números primo-rectangulares menores o iguales que dicha cota.
Indicación: Todo número natural tiene una representación única como producto de factores primos, salvo el orden (el 1 se representa como el producto vacío). El divisor más pequeño $d \neq 1$ de un entero positivo siempre es primo.

Ejercicio 21 [Método de Newton] Se desea calcular aproximaciones de raíces positivas $\sqrt[k]{r}$. Los datos de entrada son el radical k (entero positivo), el radicando r (real positivo) y una cota de error ε. Según el método de Newton, la sucesión $x_0, x_1, \ldots, x_n$ definida por la recurrencia siguiente tiene como límite $\sqrt[k]{r}$. La aproximación finaliza cuando $|x_{i+1} - x_i| < \varepsilon$.

$$x_0 = r \quad x_{i+1} = x_i - \frac{(x_i)^k - r}{k(x_i)^{k-1}} = \frac{(k-1)x_i + \frac{r}{(x_i)^{k-1}}}{k}$$

Diseña y escribe un programa que, dados $k > 0$, $r > 0$ y $\varepsilon > 0$, calcule la raíz $\sqrt[k]{r}$ por el método de Newton.

4. Subprogramas

Ejercicio 22 [Jaque mate] Diseña y escribe una función que, dadas dos posiciones del tablero de ajedrez, decida si la figura del Caballo puede saltar desde la primera posición hasta la segunda. Los datos de entrada de la función son la posición inicial del Caballo (f_1, c_1) y la posición final (f_2, c_2), donde $1 \leq f_1, c_1, f_2, c_2 \leq 8$. El dato de salida será un valor booleano que indica si el movimiento es o no posible.

Ejercicio 23 [Reverso] Diseña y escribe una función que, dado un número natural n, calcule su reverso. Por ejemplo, el reverso de 3952 es 2593 y el reverso de 7580 es 857.

Ejercicio 24 [Número palíndromo] Se dice que un número natural n es **palíndromo** si se lee igual de izquierda a derecha y de derecha a izquierda. Diseña y escribe una función que, dado un número natural n, decida si es o no palíndromo. Diseña y escribe una función que, dada una cota $c \geq 0$, obtenga todos los números palíndromos menores o iguales que dicha cota.

Ejercicio 25 [Número perfecto] Se dice que un número natural n es **perfecto** si es igual a la suma de sus divisores propios, es decir, un número n es perfecto si es **amigo de sí mismo**. Por ejemplo, $6 = 1+2+3$ es perfecto. Diseña y escribe una función que, dado un número natural n, decida si es o no perfecto.

Ejercicio 26 [Aproximaciones de π] Diseña y escribe funciones que calculen aproximaciones al número π, utilizando las fórmulas matemáticas siguientes. El dato de entrada de cada función es la cantidad k de términos calculados.

1. Fórmula de Viète:

$$\frac{2}{\pi} = \sqrt{\frac{1}{2}} \cdot \sqrt{\frac{1}{2} + \frac{1}{2}\sqrt{\frac{1}{2}}} \cdot \sqrt{\frac{1}{2} + \frac{1}{2}\sqrt{\frac{1}{2} + \frac{1}{2}\sqrt{\frac{1}{2}}}} \cdots$$

2. Producto de Wallis:

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

3. Serie de Leibniz:

$$\frac{\pi}{4} = \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots$$

4. Algoritmo de Borwein:

$$\begin{array}{lll} x_0 = \sqrt{2} & y_1 = \sqrt[4]{2} & \pi_0 = 2 + \sqrt{2} \\ x_{n+1} = \frac{1}{2}\left(\sqrt{x_n} + \frac{1}{\sqrt{x_n}}\right) & y_{n+1} = \frac{y_n\sqrt{x_n} + \frac{1}{\sqrt{x_n}}}{y_n + 1} & \pi_n = \pi_{n-1}\frac{x_n+1}{y_n+1} \end{array}$$

Ejercicio 27 [Intercambio] Diseña y escribe una función que intercambie el valor de dos variables.

Ejercicio 28 [Goldbach] La conjetura de Goldbach establece que todo número par $p>2$ puede escribirse como suma de dos números primos. Diseña y escribe una función que, dado un número par $p>2$, busque una descomposición del mismo en dos sumandos primos $n1$ y $n2$, indicando además si lo ha conseguido o no. Diseña y escribe una función que, dada una cota $c \geq 0$, obtenga todos los números pares $p>2$ menores o iguales que dicha cota que satisfacen la conjetura de Goldbach.

Ejercicio 29 [Descomposición en factores primos] Diseña y escribe una función que, dado un número natural n, calcule su descomposición en factores primos.

Ejercicio 30 [Medidas] Diseña y escribe una función que pida y lea un conjunto de números, los cuente y calcule su media, varianza y desviación típica. La desviación típica es la raíz cuadrada de la varianza. La media y la varianza de un conjunto de números $x_1, x_2, \dots, x_n$ se calculan empleando las fórmulas siguientes.

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad\qquad \sigma^2 = \frac{1}{n}\sum_{i=1}^{n} x_i^2 - \frac{1}{n^2}\left(\sum_{i=1}^{n} x_i\right)^2$$