

RELACIÓN N° 1 DE PROBLEMAS

1. Indicar cuál sería la representación mediante diagramas de flujo de las estructuras de control que pueden existir en un programa (secuencial, alternativas y repetitivas).
2. Se desea leer (por teclado) el valor de tres variables A, B y C, y asignar el valor de A a B, el de B a C y el de C a A. Hay que imprimir los valores iniciales y finales de estas variables.
3. Escribir la siguiente secuencia de palabras: rojo rojo rojo verde verde verde rojo rojo rojo verde verde verde ...
4. Escribir 20 veces el número 20, 19 veces el 19, ... , 1 vez el 1 (en las 20 líneas respectivas).
5. Calcular las soluciones de una ecuación de segundo grado de la forma: $Ax^2+Bx+C = 0$.
6. Calcular el factorial de un número sabiendo que $0! = 1$.
7. Determinar de un conjunto de 100 valores leídos por teclado cuál es el valor máximo y cuál es el mínimo.
8. Dados tres números ordenarlos de mayor a menor, indicando cual es el mayor, cual el menor y cual está en el medio. Los tres números dados no tienen porqué ser distintos.
9. Obtener la sucesión de sumas parciales (S_n), $n \in \mathbb{N}$, siendo $S_i = 1 / 2^{i-1}$. Llegar hasta S_{80} . Suponer que disponemos de una máquina que no realiza potencias.
10. Dada la sucesión de término general $a_n = (2-3n^2) / (n^2+1)$. Obtener los términos a_{45} , a_{48} , a_{51} , a_{54} , ...; hasta superar el término 100.
11. Un número con n dígitos es un **número armstrong** si la suma de las potencias n -ésimas de los dígitos que lo forman es igual al propio número. Por ejemplo: $153 = 1^3 + 5^3 + 3^3$. Determinar todos los números armstrong de tres dígitos.

12. Calcular $\binom{P}{Q}$, usando un subprograma o función que calcule el factorial de un número.

$$\text{Se sabe que } \binom{P}{Q} = \frac{P!}{Q! (P-Q)!}.$$

13. Diseñar un programa que lea pares de coordenadas (x,y) de puntos del plano. Se debe contar los puntos leídos y de ellos los que están en el interior del círculo $x^2 + y^2 = 25$. La introducción acabará cuando x e y valgan 999.

14. Escribir un programa que calcule:

$$\sum_{i=1}^n i$$

Donde n debe leerse por teclado, debiendo ser superior a 1 e inferior a 100. Suponer que disponemos de una máquina que no realiza potencias.

15. Escribir un programa que lea un número entero N de dos cifras y escriba una salida de acuerdo con el siguiente formato:

RESULTADO

(N elevado 1) (N elevado a 2) (N elevado a 3) (N elevado a 4) (N elevado a 5)

A la hora de realizar los cálculos tendremos en cuenta que la máquina con la que trabajamos sólo realiza sumas. Tras terminar el proceso preguntaremos si el usuario desea repetirlo para otro número (S/N), no terminando el programa hasta que el usuario responda N.

16. Calcular el volumen del cubo, el de un octaedro, el de un cilindro y el de una esfera mediante funciones separadas. El programa presentará en pantalla un menú similar al siguiente:

- Menú -

Volumen de un cubo.....	1
Volumen de un octaedro.....	2
Volumen de un cilindro	3
Volumen de una esfera	4
Salir del programa	5

En cada caso el usuario pulsará un número y el ordenador hará el cálculo correspondiente, presentando el resultado con un mensaje. El proceso se repetirá hasta que no se indique lo contrario. Es necesario controlar los posibles errores de entrada, tanto en la elección de opción como en los datos necesarios para calcular los volúmenes.

Nota : Las fórmulas de los volúmenes en cuestión son:

$$V_{\text{cubo}} = L^3$$

$$V_{\text{octaedro}} = \sqrt{2} \cdot L^3 / 3$$

$$V_{\text{cilindro}} = \pi \cdot R^2 \cdot A$$

$$V_{\text{esfera}} = (4 \cdot \pi \cdot R^3) / 3$$

Donde L es el lado, R es el radio y A es la altura.

17. Calcular el tiempo transcurrido entre dos horas dadas comprendidas entre las 0:0:0 y las 23:59:59. El tiempo transcurrido deberá darse en horas:minutos:segundos.
18. Dada una cantidad descomponerla en monedas de 2 euros, 1 euro, 50, 20, 10, 5, 2 y 1 céntimos de euro. Descomponerla en el mayor número de monedas posible y en el menor número de monedas posible. Por ejemplo, 7,88 euros: número máximo de monedas 788 de 1 céntimo de euro, y número mínimo, 3 de 2 euros, 1 de 1 euro, 1 de 50 céntimos, 1 de 20 céntimos, 1 de 10 céntimos, 1 de 5 céntimos, 1 de 2 céntimos y 1 de 1 céntimo. Escribir en pantalla la cantidad de monedas de cada valor necesarias para cada caso.
19. Calcular el MCM (Mínimo Común Múltiplo) y el MCD (Máximo Común Divisor) de dos números x e y. Controlar que ambos números son mayores que cero.
20. Completar los siguientes apartados:
 - a) Expresar las soluciones de los problemas del 2 al 10 mediante pseudocódigo.
 - b) Expresar las soluciones de los problemas del 11 al 19 mediante diagramas de flujo.
 - c) Expresar las soluciones de los problemas 5 y 8 mediante tablas de decisión y árboles de decisión.
 - d) Expresar las soluciones de los problemas del 2 al 10 mediante diagr. de Tabourier.
21. Realizar un programa en C que tras pulsar una tecla nos diga el código ASCII de la misma. Se deberán controlar los códigos extendidos, indicando los mismos con "0-" antes del código de la tecla. Se terminará al pulsar *Esc*.
22. Realizar un programa en C que calcule una diferencia de fechas, en la que las fechas de entrada serán dadas en formato: AAAA-MM-DD. Deberán controlarse todos los posibles errores de entrada, teniendo en cuenta que el rango de años permitido es de 0 a 2500. En los controles de entrada se tendrán en cuenta los años bisiestos, que son todos los múltiplos de 4, excepto los múltiplos de 100 y no de 400. La diferencia deberá darse en días, meses y años.
23. Dado un D.N.I. cualquiera, calcular la letra del N.I.F. del mismo. Para ello se tendrá en cuenta que dicha letra se averigua mediante la operación:

$$n = \text{DNI} - 23 * \text{PARTE_ENTERA}(\text{DNI}/23)$$

Es decir, n es igual al resto de la división entera de DNI entre 23. n servirá de índice del array que almacena la letra del N.I.F. correspondiente, y cuyas 23 posiciones son:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
T	R	W	A	G	M	Y	F	P	D	X	B	N	J	Z	S	Q	V	H	L	C	K	E

La salida tendrá el formato: **DNI - letra**. Se controlará que no existen DNIs de más de 8 dígitos.

24. Se introducen por teclado el Número de Expediente, Sexo y la Calificación de una serie de alumnos (no más de 100). Dichos datos deberán quedar almacenados en tablas. Se desea obtener el número de alumnos varones aprobados, el número de alumnos varones suspensos, y la nota media de los varones. Lo mismo para las mujeres. También se hallará el número total de alumnos aprobados y suspensos, así como la nota media global.

El Número de Expediente estará formado por 5 dígitos, acabándose la introducción al entrar el Número de Expediente 0. El sexo admitirá los valores V y M. La calificación irá de 0 a 10, permitiendo decimales. Deberán controlarse todos los posibles errores de entrada.

25. Implementar la **búsqueda dicotómica (o binaria)** para un vector de N (no superior a 100) elementos. En primer lugar deberá leerse N, posteriormente los elementos, y finalmente el valor buscado, controlando en cada caso cualquier posible error.

Si llamamos **I** a la posición inicial del intervalo de array a tratar, **F** a la posición final y **C** a la central, **T** a la tabla en cuestión y **n** al valor buscado; La búsqueda dicotómica posee los siguientes pasos (algoritmo):

- 1) Asignar I y F (posición inicial y final del array), y saber que buscamos la posición que contenga el valor *n*.
- 2) $C = \text{PARTE_ENTERA}((I + F) / 2)$.
- 3) Si $T[C] = n \Rightarrow$ Se encontró el elemento, fin del algoritmo.
Si $T[C] > n \Rightarrow F = C - 1$.
Si $T[C] < n \Rightarrow I = C + 1$.
- 4) Volver al paso 2 hasta que $I > F$, en cuyo caso si no se ha encontrado el elemento, es que éste no existe y se ha producido un error.

Debe tenerse en cuenta que este tipo de búsqueda a diferencia de la **secuencial** (recorrer el array elemento por elemento desde su posición inicial hasta la final) sólo puede aplicarse sobre arrays cuyos elementos estén ordenados (ascendentemente en el caso del algoritmo expresado) y preferentemente no repetidos, por tanto también deberá controlarse esto.

26. Realizar el cálculo de una diferencia de fechas, en la que las fechas de entrada serán dadas en formato: AAAA-MM-DD. Deberán controlarse todos los posibles errores de entrada, teniendo en cuenta que el rango de años permitido es de 0 a 2500. En los controles de entrada se tendrán en cuenta los años bisiestos, que son todos los múltiplos de 4, excepto los múltiplos de 100 y no de 400. La diferencia deberá darse en días y en su cálculo se tendrán en cuenta los años bisiestos.

27. Se van a introducir por teclado los nombres y apellidos de 3 personas con el formato:

Apellido1 Apellido2, Nombre

Dichos datos deberán aparecer ordenados alfabéticamente por apellidos. En caso de igualdad de apellidos aparecerá primero aquel cuyo nombre tenga menor número de caracteres. Como salida deben escribirse cadenas con el formato:

Nombre Apellido1 Apellido2

En el caso de que alguno de los apellidos introducidos sea *Vega*, se sacará el nombre de dicha persona indicando si éste es su primer o segundo apellido.

28. Un **cuadrado mágico** de orden n es una tabla cuadrada de $n \times n$ que contiene en sus n^2 casillas, una vez y sólo una, cada uno de los números enteros de 1 a n^2 , de forma que la suma de los n términos de cada fila y cada columna y las dos diagonales principales sea siempre la misma. Para realizar un cuadrado mágico de orden impar n existe un algoritmo de cálculo basado en las siguientes reglas:

- 1) Colocar el 1 en la casilla situada debajo de la casilla central.
- 2) Cuando se ha colocado el entero $x-1$ en la casilla (i,j) se coloca x en la casilla $(i+1,j+1)$.
- 3) Si el índice calculado sobrepasa n se toma igual a 1.
- 4) Si se encuentra una casilla ya ocupada, por ejemplo (i,k) , se intentará colocar en la casilla $(i+1,k-1)$. Si $k = 0$ se toma $k = n$.

Ejemplos:

Cuadrado mágico de orden 3

4	9	2
3	5	7
8	1	6

Cuadrado mágico de orden 5

11	24	7	20	3
4	12	25	8	16
17	5	13	21	9
10	18	1	14	22
23	6	19	2	15

Realizar el programa que calcule el cuadrado mágico de orden n . Donde n debe

controlarse que sea un número impar inferior o igual a 15. Una vez calculado dicho cuadrado se escribirá en pantalla con un formato similar al mostrado en estos ejemplos.

29. Se introduce un grupo de palabras (no más de 100) en el ordenador. La entrada terminará al introducir la palabra FINAL. Queremos obtener al finalizar la entrada un listado de las distintas palabras introducidas y el número de veces que se ha repetido cada una de ellas.
30. Se lee un número y como resultado se debe indicar si es un término de la *sucesión de Fibonacci* (1, 1, 2, 3, 5, 8, 13, 21,...) o no, y en caso de que lo sea cuál es su orden. Dicha sucesión viene dada por: $a_1 = 1$, $a_2 = 1$, ..., $a_n = a_{n-1} + a_{n-2}$
31. Seleccionar de una lista de 100 números distintos, dados por teclado, todas las ternas de números tales que: $a^2 + b^2 + c^2 = 105$. Si no hay ninguna escribir el literal **“No hay ninguna”**.
32. Se llaman *puntos de silla* de una matriz a aquellos elementos que son mínimo de su fila y máximo de su columna. Encontrar los puntos de silla de una matriz que no contenga ningún elemento repetido (a controlar), sabiendo que los datos se leen por teclado en el orden: NF (número de filas), NC (número de columnas), M(1,1), M(1,2), ..., M(1,NC), M(2,1),... Se deben imprimir los puntos de silla y su posición en la matriz. Si no hay ninguno imprimir el mensaje: **“No hay puntos de silla en la matriz”**. La matriz no poseerá más de 100 filas o columnas.
33. Se deben leer por teclado los datos de una serie de personas (no más de 100). Para cada una de estas personas se introducirán los datos: número de D.N.I. y sueldo anual. El fin de la introducción se indicará con el D.N.I. número 100.000.000. Se desea imprimir el D.N.I. de cada persona junto con el impuesto que debe pagar, teniendo en cuenta que si el sueldo es menor que 2.000.000 ptas. el impuesto es nulo, si está entre 2.000.000 y 4.000.000 ptas. el impuesto es el 10% de lo que excede de 2.000.000 ptas., y si el sueldo es superior a 4.000.000 ptas. el impuesto es de 200.000 ptas. más el 20% de lo que el sueldo exceda de 4.000.000.
34. Tenemos una tabla ya cargada en memoria con las temperaturas habidas en una ciudad durante todo un año. Es decir, la tabla consta de 12 posiciones, cada una de las cuales corresponde a un mes. Cada posición tiene 31 campos que corresponden a los días del mes y a su vez cada campo se subdivide en 2 subcampos, en los cuales vienen las temperaturas máxima y mínima de ese día.
- Dar la posibilidad de que dado un mes y un día de dicho mes se devuelvan las temperaturas media, máxima y mínima de ese día. Además se desea saber la temperatura media de todos los meses del año y la temperatura media del año. En este caso a la hora de realizar los controles de entrada se supondrán meses de 31 días. Las temperaturas vendrán dadas mediante números reales.
35. Leer dos vectores de números enteros V1 y V2, de tamaños N y M ($N \leq M$) respectivamente, y determinar si V1 está incluido en V2. Si está incluido presentar el mensaje **“Está incluido”**, sino escribir **“No está incluido”**. Los vectores tendrán un

tamaño máximo de 100 posiciones, lo cual debe controlarse a la hora de aceptar los números N y M.

36. Realizar la intersección de los elementos de dos vectores V1 y V2 (de tamaños N y M respectivamente) y guardar el resultado en un vector V3. Escribir el resultado en pantalla. Los vectores V1 y V2 tendrán un tamaño máximo de 100 posiciones, lo cual debe controlarse a la hora de aceptar los números N y M. Los elementos de ambos vectores serán leídos por teclado.
37. Multiplicar dos matrices bidimensionales. Los datos se leen por teclado en el orden: NF (número de filas), NC (número de columnas), M(1,1), M(1,2), ..., M(1,NC), M(2,1),... Deberá controlarse si realmente es posible el producto, no permitiendo la lectura de la segunda matriz hasta que ésta posibilite el producto. El resultado de la multiplicación se escribirá en pantalla. Las matrices no poseerán más de 100 filas o columnas.
38. Escribir un programa que lea una sucesión de 15 caracteres alfabéticos o blancos e imprima la sucesión obtenida por ella eliminando los blancos y sustituyendo cada letra por la que le sigue en orden alfabético, considerando circular el alfabeto, y, por tanto, el carácter "A" como siguiente del "Z". Un ejemplo de datos de entrada y los correspondientes datos de salida correctos es el siguiente (el símbolo "/" indica blanco):

Entrada: CODIGO/SECRETO///

Salida: DPEJHPTFDSFUP

39. Estamos en la Edad Media y el rey os ha encargado utilizar vuestra máquina infernal para la resolución del problema de decidir a quién dar a su hija como esposa. El ha numerado de 1 a N a los pretendientes, y os va a dar para cada uno de ellos:
1. Su nombre.
 2. Su edad.
 3. Su peso.
 4. Su estatura.
 5. Sus bienes, que pueden ser: Casa, Caballos y Monedas (de Oro).

El rey quiere que vos escribáis un "programa" que, tras leer los datos de los nobles caballeros, y sabiendo que no existen más de 100 pretendientes:

- Lea un número d, siendo $1 \leq d \leq N$.
- Renumere los pretendientes considerando como primero el de lugar d, y haciendo seguir al de lugar N los d-1 primeros.
- De entre todos los pretendientes se quede con aquellos, que posean una edad entre 25 y 30 años, con una estatura entre 1,75 y 1,80 m. y un peso entre 70 y 80 kilos, pues busca esposar a su hija con un joven bien agraciado.
- Además, deberá elegir a aquéllos que posean el mayor número de bienes posibles,

pues para el buen rey es importante que su hija tenga de todo una vez casada.

- Finalmente deberá comunicar si existe algún pretendiente que cumpla todos los requisitos necesarios para casarse con la princesa. Si no existiera, dará la mala noticia con el mensaje:

“Su majestad, no existe galán alguno a la altura de su hija”

Si existiera más de uno, se escogerá el de menor orden, de aquí la renumeración inicial. Indicando la elección con el mensaje:

**“El caballero (nombre) de (edad) años de edad, (peso) Kgs. de peso y (estatura) m. de estatura, con los siguientes bienes: (bienes en cuestión).
Ha sido elegido para esposar a la bella princesa.”**

A la hora de indicar los bienes, es posible que todos los caballeros no tengan las mismas pertenencias, por lo que se indicarán las que poseen con esas palabras (Casa, Caballos y Monedas), terminando de introducir los bienes cuando se hayan introducido todos los posibles o cuando se indique la palabra: **“Ya”**. Lógicamente, sólo deberán reconocerse estas palabras, controlando todos los posibles errores de entrada.

Por último, indicar que debe quedar entendido que no podéis esperar ninguna ayuda sobre los problemas referentes al lenguaje de programación, estructuras de datos, algoritmos, etc., ni del rey ni de los magos de la corte (que, al contrario, os miran con envidia y recelo).

Ultima advertencia: El rey es muy irascible y os ha dado tiempo hasta el ocaso.