

1. En España cada persona está identificada con un Documento Nacional de Identidad (DNI) en el que figura un número y una letra, por ejemplo 56999545W La letra que sigue al número se calcula siguiendo la metodología que vamos a indicar. Crea un programa que calcule la letra de un DNI a partir del número de DNI que introduzca el usuario. Es decir, se debe pedir el DNI sin la letra por teclado y el programa nos devolverá el DNI completo (con la letra). Para calcular la letra, se debe tomar el resto de dividir nuestro número de DNI entre 23. El resultado debe estar por tanto entre 0 y 22. Crea un método obtenerLetra(int numeroDNI) donde según el resultado de la anterior fórmula busque en un array de caracteres la posición que corresponda a la letra. Esta es la tabla de caracteres:

Posición: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22

Letra: T R W A G M Y F P D X B N J Z S Q V H L C K E

Por ejemplo, si introducimos el DNI 20267079, el resto de dividirlo por 23 sería 8, luego la letra sería la P, que es la que ocupa esa posición en la matriz de caracteres.

2. Un histograma para una colección de datos es una secuencia de parejas de la forma (d, f), donde d es un dato y f es su frecuencia en la colección. Por ejemplo, suponga que se le pide a 20 personas calificar con las letras a b c d y e el desempeño del gobierno actual, y que se obtienen las siguientes respuestas: c b c a b c d e e a b b d c a c c b d a.

El histograma para esta colección de datos se muestra enseguida en dos formas: con números y con asteriscos:

a: 4 a: ****

b: 5 b: *****

c: 6 c: *****

d: 3 d: ***

e: 2 e: **

Considere el problema de construir un algoritmo que haga un histograma para una lista de hasta 100 valores, donde cada valor es un número entero comprendido en el intervalo 1 al 5. Las entradas (datos conocidos) para el algoritmo son:

- El número de datos de la colección
- La colección misma

La salida esperada (dato desconocido)

- El histograma de la colección

3. Suponga que se quiere construir un algoritmo que califique un examen de selección múltiple presentado por un estudiante de Programación de Computadores. En cada una de las preguntas del examen, el estudiante debió elegir una de cinco opciones, identificadas con las letras A, B, C, D y E. Las entradas (datos conocidos) para el algoritmo son:

- El número de preguntas que tenía el examen
- Cada una de las respuestas dadas por el estudiante
- Las respuestas correctas.

La salida esperada (dato desconocido) es:

- La nota obtenida. Esta nota corresponde al número de aciertos que tuvo el estudiante.
4. Escriba un algoritmo que efectúe la normalización de una colección de números reales. Para llevar a cabo esta normalización, se debe en primer lugar encontrar el número mayor de la colección; luego se divide cada número por dicho valor máximo, de forma que los valores resultantes (normalizados) estén comprendidos en el intervalo del 0 al 1.
 5. Escriba un algoritmo que lea un arreglo de números enteros, y un número x, y escriba en la pantalla todos los índices de las posiciones del arreglo donde está x. Por ejemplo, si el arreglo es el que aparece enseguida y x es 2:

1	2	3	100	23	2	2	1
---	---	---	-----	----	---	---	---

El programa debe escribir: 1 5 6.

6. En álgebra lineal las matrices son tema central. Sobre ellas se definen varias operaciones, como por ejemplo:
 - La suma de dos matrices. Si A y B son matrices de igual dimensión, la matriz $C=A+B$ se calcula haciendo que $C[i][j] = A[i][j]+B[i][j]$, para todo i y j válidos.
 - La traspuesta de una matriz. Si A es una matriz de dimensión NxM, la matriz $B=A^t$ se calcula haciendo que $B[i][j] = A[j][i]$, para todo i y j válidos. Note que esto quiere decir que las filas se convierten en columnas y que la dimensión de B es MxN.
 - La traza de una matriz cuadrada. Si A es una matriz de dimensión NxN, la matriz traza es la suma de todos los elementos de la diagonal principal.
 - La multiplicación de dos matrices. Si A y B son matrices de dimensiones nxm y mxk, respectivamente, la matriz $C=A*B$, de dimensión nxk, se calcula haciendo que:

$$C[i][j] = \sum_{p=0}^{m-1} A[i][p] * B[p][j]$$

Especifique y escriba un algoritmo para cada una de estas operaciones.

7. Diseñe un algoritmo que permita guardar en un arreglo las sumas de las filas de una matriz. Esto es, la suma de los elementos de la primera fila deberá quedar guardada en la primera posición del arreglo, la suma de los elementos de la segunda fila en la segunda posición, y así sucesivamente para todas las filas de la matriz. La máxima dimensión de la matriz es 100x50 (100 filas y 50 columnas) y la del vector es 100. Por ejemplo, si el usuario ingresa la siguiente matriz de 3x5 (3 filas, 5 columnas)

3.5	6.5	30	8.2	0
4	0	-1	3.6	1.4
10	-1.5	3.4	6.6	2

El resultado sería un arreglo siguiente:

48.2	7	19
------	---	----

8. Dado una matriz, determinar la posición [i,j] del mayor y menor número.
9. Escribir un programa en el que se genere aleatoriamente un vector de 20 números enteros. El vector ha de quedar de tal forma que la suma de los 10 primeros elementos sea mayor que la suma de los 10 últimos elementos. Mostrar el vector original.
10. Crear un programa que cree un array con 1000 letras mayúsculas aleatorias y que cuenta cuántas veces aparece cada letra en el array.
11. Una secuencia de bits puede interpretarse como un número decimal. Cada bit está asociado a una potencia de dos, partiendo desde el último bit. Por ejemplo, la secuencia 01001 representa al número decimal 9, ya que:

$$0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 9$$

Escriba la función `numero_decimal(datos)` que entregue la representación decimal de un arreglo de datos.

12. Una fábrica de autos produce tres modelos: sedán, camioneta y económico. Cada auto necesita para su producción material, personal, impuestos y transporte. Los costos en unidades por cada concepto son los siguientes:

(Costos)	Sedán	Camioneta	Económico
Material	7	8	5
Personal	10	9	7
Impuestos	5	3	2
Transporte	2	3	1

Semanalmente, se producen 60 sedanes, 40 camionetas y 90 económicos.

Los costos de una unidad de material, personal, impuestos y transporte son respectivamente 5, 15, 7 y 2.

Escriba un programa que muestre:

- Las unidades semanales necesarias de material, personal, impuestos y transporte.
- El costo total de un auto de cada modelo.
- El costo total de la producción semanal.

13. La dieta Cambridge es una dieta que fue popular en la década de los 80, y fue el resultado de más de ocho años de trabajo clínico e investigación de un equipo de científicos liderados por el doctor Alan H. Howard en la Universidad de Cambridge. La dieta combina un balance preciso de carbohidratos, proteínas de alta calidad y grasa, junto con vitaminas, minerales, oligoelementos y electrolitos. Millones de personas han usado la dieta en años recientes para bajar rápidamente de peso.

Para alcanzar las proporciones de nutrientes deseadas, el doctor Howard debió incorporar una gran variedad de comidas en la dieta. Cada comida provee varios de los nutrientes, pero no en las proporciones correctas. Por ejemplo, la leche descremada es una buena fuente de proteínas, pero contiene mucho calcio. Por esto, se usó harina de soya (que tiene poco calcio) para proveer las proteínas; sin embargo, tiene proporcionalmente mucha grasa, por lo que se agregó suero de leche a la dieta, que desafortunadamente contiene muchos carbohidratos... como se hace evidente, el delicado problema de balancear los nutrientes es complejo.

La siguiente tabla muestra el aporte en nutrientes por cada 100 gramos de cada uno de los tres ingredientes (leche descremada, harina de soya y suero de leche):

Nutrientes	LD	HS	SL
Proteínas	36	51	13
Carbohidratos	52	34	74
Grasas	0	7	1.1

La dieta de Cambridge debe proveer 33 gramos de proteínas, 45 gramos de carbohidratos y 3 gramos de grasa.

Escriba un programa que muestre qué cantidades de ingredientes se debe usar para satisfacer los requerimientos de la dieta de Cambridge.

14. Estudios demográficos muestran que, cada año, el 5% de la población de una ciudad se muda a los suburbios (y el 95% se queda), mientras que el 3% de la población de los suburbios se muda a la ciudad (y el 97% se queda).

Estos datos pueden ser representados en una matriz de migración:

$$M = \frac{1}{100} \begin{bmatrix} 95 & 3 \\ 5 & 97 \end{bmatrix}$$

Escriba un programa que pregunte al usuario cuáles son las poblaciones de la ciudad y los suburbios en el año 2011, y entregue una tabla con las poblaciones proyectadas para los siguientes 10 años:

Poblacion ciudad: **600**
Poblacion suburbios: **400**

Anno	Ciudad	Suburbios

2012	582.000	418.000
2013	565.440	434.560
2014	550.205	449.795
2015	536.188	463.812
2016	523.293	476.707
2017	511.430	488.570
2018	500.515	499.485
2019	490.474	509.526
2020	481.236	518.764
2021	472.737	527.263

15. Considere ahora la siguiente variación. Suponga que todos los años hay 14000 personas que se mudan a la ciudad desde fuera de la región (no desde los suburbios) y 9000 personas abandonan la región; además, hay 13000 personas que se mudan anualmente a los suburbios desde fuera de la ciudad.

Modifique el programa anterior para resolver este problema.

16. En un informe anual de SansanoGas S.A., el presidente informa a sus accionistas la cantidad anual de producción de barriles de 50 litros de lubricantes normal, extra y súper, en sus dos refinerías:

Refinería	Normal	Extra	Súper

A	3000	7000	2000

B	4000	500	600

Además, informa que en cada barril de 50 litros de lubricante existe la siguiente composición en litros de aceites finos, alquitrán y grasas residuales:

Componente	Normal	Extra	Súper
Aceites finos	10	5	35
Alquitrán	15	4	31
Grasas residuales	18	2	30

Escriba la función `totales_anuales(a, b)` que reciba como parámetros ambas matrices y retorne un arreglo con los totales de aceites finos, alquitrán y grasas residuales presentes en la producción anual.

17. Una matriz a es antisimétrica si para todo par de índices i y j se cumple que $a[i,j] = -a[j,i]$. Escriba la función `es_antisimetrica(a)` que indique si la matriz a es antisimétrica o no. Cree algunas matrices simétricas y otras que no para probar su función.
18. Una matriz a es diagonal si todos sus elementos que no están en la diagonal principal tienen el valor cero. Por ejemplo, la siguiente matriz diagonal:

$$\begin{bmatrix} 9 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

Escriba la función `es_diagonal(a)` que indique si la matriz a es diagonal o no.

19. Una matriz es idempotente si el resultado del producto matricial consigo misma es la misma matriz. Por ejemplo:

$$\begin{bmatrix} 2 & -2 & -4 \\ -1 & 3 & 4 \\ 1 & -2 & -3 \end{bmatrix} \begin{bmatrix} 2 & -2 & -4 \\ -1 & 3 & 4 \\ 1 & -2 & -3 \end{bmatrix} = \begin{bmatrix} 2 & -2 & -4 \\ -1 & 3 & 4 \\ 1 & -2 & -3 \end{bmatrix}$$

Escriba la función `es_idempotente(a)` que indique si la matriz a es idempotente o no.

20. Se dice que dos matrices A y B conmutan si los productos matriciales entre A y B y entre B y A son iguales. Por ejemplo, estas dos matrices si conmutan:

$$\begin{bmatrix} 1 & 3 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} -1 & 3 \\ 3 & 0 \end{bmatrix} = \begin{bmatrix} -1 & 3 \\ 3 & 0 \end{bmatrix} \begin{bmatrix} 1 & 3 \\ 3 & 2 \end{bmatrix} = \begin{bmatrix} 8 & 3 \\ 3 & 9 \end{bmatrix}$$

Escriba la función conmutan que indique si dos matrices conmutan o no.

21. Escriba un algoritmo que transforme 2 números enteros positivos a su equivalente en binario, los cuales deberán estar almacenados e dos arreglos. Posteriormente realice las operaciones lógicas AND y OR entre ellos y almacene los resultados en arreglos.
22. Una fábrica de muebles tiene 16 representantes que viajan por toda España ofreciendo sus productos. Para tareas administrativas el país está dividido en cinco zonas: Norte, Sur, Este, Oeste y Centro. Mensualmente almacena sus datos y obtiene distintas estadísticas sobre el comportamiento de sus representaciones de sus representantes en cada zona. Se desea hacer un programa que lea los datos de todos los representantes con sus ventas en cada zona y calcule el total de ventas de una zona introducida por teclado, el total de ventas de un vendedor introducido por teclado en cada una de las zonas y el total de ventas de un día y para cada uno de los representantes.
23. Podemos definir un polígono definiendo las coordenadas de cada uno de sus lados. Diseñas la estructura de datos que permita definir un polígono de lado n con un máximo de 30 lados y crear un algoritmo que permita introducir las coordenadas cartesianas de cada uno de sus lados.
24. La Universidad del Valle requiere un programa que le permita conocer cómo califican los estudiantes la comida de la cafetería central. Para ello definió una escala de 1 a 10 (1 denota horrible y 10 denota excelente). El programa debe ser capaz capturar la calificación de cualquier número de estudiantes (no se sabe cuántos estudiantes se encuestarán, así que cuando el encuestador ingrese la calificación de 0, se sabrá que la encuesta habrá concluido). El programa deberá mostrar en su salida cuántos estudiantes fueron encuestados así como el resumen de la encuesta con histograma así:

Estudiantes encuestados: 111

Frecuencia de las calificaciones:

Calificación núm	Estudiantes	Histograma
1	3	***
2	2	**
3	10	*****
...
8	9	*****
9	15	*****
10	7	*****

Utilice un arreglo de enteros de 10 elementos para almacenar la frecuencia con la que ocurre cada calificación

25. En teoría de la computación, la función de Ackermann es una función recursiva que toma dos números naturales como argumentos y devuelve un único número natural. Escriba un programa que determine la serie de Ackerman cuando se proporcionan los parámetros m y n . Como norma general se define como sigue:

$$A(m, n) = \begin{cases} n + 1 & \text{Si } m = 0 \\ A(m - 1, 1) & \text{Si } m > 0 \text{ y } n = 0 \\ A(m - 1, A(m, n - 1)) & \text{Si } m > 0 \text{ y } n > 0 \end{cases}$$

26. Se dice que una matriz tiene un punto de silla si alguna posición de la matriz es el menor valor de su fila, y a la vez el mayor de su columna. Escribir un programa que tenga como entrada una matriz de números reales, y calcular la posición de un punto de silla (si es que existe).
27. El departamento de policía de la ciudad de Tuxtepec ha acumulado información referente a las infracciones de los límites de velocidad durante un determinado periodo de tiempo. El departamento ha dividido la ciudad en cuatro cuadrantes y desea realizar una estadística de las infracciones a los límites de velocidad en cada uno de ellos. Para cada infracción se ha preparado una tarjeta que contiene la siguiente información:
- Número de registro del vehículo.
 - Cuadrante en el que se produjo la infracción.
 - Límite de velocidad en milla por hora

Diseñe un diagrama para producir 2 informes; el 1o. que contiene una lista de la multa de velocidad recolectadas, donde la multa se calcula como la suma del costo de la corte (\$20,000) mas \$ 1,250 por cada mph que exceda la velocidad limite. Prepare una tabla con los siguientes resultados:

Este informe debe ser seguido de un segundo en el cual se proporcione un análisis de las infracciones por cuadrante. Para cada uno de los 4 cuadrantes mencionados, debe darse el número de infracciones y la multa promedio ordenado por éste último.

Utilizar para este ejercicio un menú de entrada que me permita seleccionar la acción a realizar.

28. ApocalipSYS

Un prestigioso investigador de la “Universidad del Fin del Mundo” está seguro de haber logrado descubrir un patrón que le permite determinar el año exacto del apocalipsis, en su investigación encontró que los eventos catastróficos se

presentan en años bisiestos y para los cuales la suma de sus dígitos produce un número primo. Su trabajo consiste en determinar a partir de un año dado cual será el próximo año con un evento catastrófico.

Entrada:

Un número entero que corresponde al año a partir del que se debe determinar el próximo “apocalipsis”. (Un numero entero positivo entre 0 y 10000).

Salida:

El año del próximo evento catastrófico

Ejemplo

Entrada	Salida
2011	2012
2016	2032
2041	2056

Explicación:

Para el segundo caso los años bisiestos que siguen al 2016 (inclusive) son 2016, 2020, 2024, 2028 y 2032. El año 2032 es bisiesto y la suma de sus dígitos es 7 (el cual es primo).

29. El administrador IT de una empresa multinacional está cansado de que sus usuarios utilicen contraseñas simples para los sistemas de información, así que ha decidido que la contraseña elegida por cada usuario sea convertida de forma que sea difícil de adivinar utilizando las siguientes reglas:

- Las letras mayúsculas son reemplazadas por minúsculas y del mismo modo en el sentido contrario (señorita Antioquia, año 2008).
- Las vocales (sin importar si están en mayúsculas o minúsculas) son reemplazadas por números (a=4, e=3, i=1, o=0, u=8).
- Cada dos caracteres se incluirá un símbolo de esta secuencia: +-* / repitiendo siempre la misma secuencia.

Entrada:

Un texto con la contraseña elegida por el usuarios (solo contendrá caracteres del alfabeto)

Salida:

El texto modificado según las reglas del administrador IT

Ejemplo

Entrada	Salida
HABIANDOSGAtos	h4+b1-4n*d0/sg+4T-0S
BuenaEsa	b8+3N-43*S4
cLaVe	Cl+4v-3

30. En etapa preescolar a los niños se les enseña que para sumar dos números de más de un dígito los números se suman dígito a dígito de derecha a izquierda y si el resultado es mayor a 9 se “llevan” un 1 desde una posición

a la posición siguiente. Su trabajo es contar el número de “lleva 1” para cada problema de adición dado.

Entrada:

En cada entrada se tienen dos enteros positivos de menos de 10 dígitos

Salida

Para cada entrada, calcular el número de operaciones “lleva 1” que resultan de adicionar los dos números.

Ejemplo

Entrada	Salida
123 456	0
555 555	3
123 594	1

Explicación: En el tercer caso se suma $3+4=7$, $2+9=11$ (se debe llevar 1), $5+1+1$ (el que se llevó). Se hace un “lleva 1”.

31. La NASA ha decidido enviar un mensaje numérico a un planeta llamado PRIMUS recientemente descubierto en una galaxia lejana, en dicho planeta existe una civilización desconocida que se sabe, puede recibir solo mensajes numéricos. Se presume por comunicaciones previas que la civilización de PRIMUS odia los números primos, sin embargo se desconoce la base que utilizan los Aliens para interpretar los mensajes numéricos. Con el fin de evitar una guerra inter-galáctica su trabajo consiste en analizar los números proporcionados por la NASA para determinar si cada número es primo o no en cualquiera de las bases (desde la base 2 hasta 10).

Entrada:

Un número compuesto por N dígitos (no tendrá más de 10) que la NASA desea enviar a PRIMUS en una base desconocida (2 a 10).

Salida:

SI o NO

SI: El número es primo en alguna de las bases de 2 a 10.

NO: El número no es primo en ninguna base.

Ejemplo

Entrada	Salida
1110	NO
101	SI
22021	SI

Explicación:

El primer número no es primo en ninguna base, si estuviese en base 2 su valor sería 14, el cual no es primo, en base 3 el número es 39, que tampoco

es primo, en base 4 el número sería 84 que tampoco es primo, sucesivamente el número no es primo en ninguna de las bases de 2-10. El segundo número es primo en base 2 (5 es primo), por ejemplo en base 10 también lo es (101 es primo) El tercer número en base 3 equivale a 223, y 223 es primo.

32. Diseñar una función que toma una tabla bidimensional de enteros, representando un tablero de ajedrez. Disponemos de las constantes PB (peón blanco), TN (torre negra), etc. (P, T, C, A, R, D). Dicho módulo debe devolver un valor booleano, que indique si el rey negro está amenazado.
33. Se pretende realizar un programa para gestionar la lista de participaciones en una competición de salto de longitud. El número de plazas disponible es de 10. Sus datos se irán introduciendo en el mismo orden que vayan inscribiéndose los atletas. Diseñar el programa que muestre las siguientes opciones:
- Inscribir un participante.
 - Mostrar listado de datos.
 - Mostrar listado por marcas.
 - Finalizar el programa.

Si se selecciona 1, se introducirán los datos de uno de los participantes: Nombre, mejor marca del 2002, mejor marca del 2001 y mejor marca del 2000. Si se elige la opción 2, se debe mostrar un listado por número de dorsal. La opción 3 mostrará un listado ordenado por la marca del 2002, de mayor a menor. Tras procesar cada opción, se debe mostrar de nuevo el menú inicial, hasta que se seleccione la opción 4, que terminará el programa.

34. Se desea escribir un programa que permita manejar la información de habitantes de un complejo habitacional. El mismo posee 7 torres; a su vez cada torre posee 20 pisos y cada piso 6 departamentos. Se desea saber:
- Cantidad total de habitantes del complejo.
 - Cantidad promedio de habitantes por piso de cada torre.
 - Cantidad promedio de habitantes por torre
35. La empresa de transportes "The Big Old" cuenta con N choferes, de los cuales se conoce su nombre y los kilómetros que conducen durante cada día de la semana, esa información se guarda en un arreglo de $N \times 6$. Se requiere un algoritmo que capture esa información y genere un vector con el total de kilómetros que recorrió cada chofer durante la semana.
36. Crea un programa que pida su nombre y su (primer) apellido al usuario, A partir de estos datos, deberá crear una cadena que contenga el apellido, una coma, un espacio y el nombre, y deberá mostrarla en pantalla. Por ejemplo, si el nombre es "Nacho" y el apellido es "Cabanés", la cadena resultante será "Cabanés, Nacho".

37. Crea un programa que pida al usuario 10 palabras, las guarde en un array y luego le pregunte de forma repetitiva qué texto quiere buscar. Le responderá si dicho texto era alguno de lo que se habían introducido inicialmente. Dejará de repetirse cuando se introduzca "fin".
38. Escribir una función que simule una calculadora científica que permita calcular el seno, coseno, tangente, exponencial y logaritmo neperiano. La función preguntará al usuario el valor y la función a aplicar, y mostrará por pantalla una tabla con los enteros de 1 al valor introducido y el resultado de aplicar la función a esos enteros.
39. Escribir que reciba una frase y devuelva un diccionario con las palabras que contiene y su longitud.
40. Escribir una función reciba una lista de notas y devuelva la lista de calificaciones correspondientes a esas notas.
41. Escribir una función reciba un diccionario con las asignaturas y las notas de un alumno y devuelva otro diccionario con las asignaturas en mayúsculas y las calificaciones correspondientes a las notas.
42. Escribir una función reciba dos tuplas y devuelva su producto escalar.
43. Escribir una función reciba un diccionario con las asignaturas y las notas de un alumno y devuelva otro diccionario con las asignaturas en mayúsculas y las calificaciones correspondientes a las notas aprobadas.
44. Escribir un programa que guarde en un diccionario los precios de las frutas de la tabla, pregunte al usuario por una fruta, un número de kilos y muestre por pantalla el precio de ese número de kilos de fruta. Si la fruta no está en el diccionario debe mostrar un mensaje informando de ello.

Fruta	Precio
Plátano	1.35
Manzana	0.80
Pera	0.85
Naranja	0.70

45. Escribir un programa que pregunte una fecha en formato dd/mm/aaaa y muestre por pantalla la misma fecha en formato dd de <mes> de aaaa donde <mes> es el nombre del mes.
46. Escribir un programa que almacene el diccionario con los créditos de las asignaturas de un curso {'Matemáticas': 6, 'Física': 4, 'Química': 5} y después muestre por pantalla los créditos de cada asignatura en el formato <asignatura> tiene <créditos> créditos, donde <asignatura> es cada una de

las asignaturas del curso, y <créditos> son sus créditos. Al final debe mostrar también el número total de créditos del curso.

47. Escribir un programa que gestione las facturas pendientes de cobro de una empresa. Las facturas se almacenarán en un diccionario donde la clave de cada factura será el número de factura y el valor el coste de la factura. El programa debe preguntar al usuario si quiere añadir una nueva factura, pagar una existente o terminar. Si desea añadir una nueva factura se preguntará por el número de factura y su coste y se añadirá al diccionario. Si se desea pagar una factura se preguntará por el número de factura y se eliminará del diccionario. Después de cada operación el programa debe mostrar por pantalla la cantidad cobrada hasta el momento y la cantidad pendiente de cobro.
48. Escribir un programa que permita gestionar la base de datos de clientes de una empresa. Los clientes se guardarán en un diccionario en el que la clave de cada cliente será su NIF, y el valor será otro diccionario con los datos del cliente (nombre, dirección, teléfono, correo, preferente), donde preferente tendrá el valor True si se trata de un cliente preferente. El programa debe preguntar al usuario por una opción del siguiente menú: (1) Añadir cliente, (2) Eliminar cliente, (3) Mostrar cliente, (4) Listar todos los clientes, (5) Listar clientes preferentes, (6) Terminar. En función de la opción elegida el programa tendrá que hacer lo siguiente:
- Preguntar los datos del cliente, crear un diccionario con los datos y añadirlo a la base de datos.
 - Preguntar por el NIF del cliente y eliminar sus datos de la base de datos.
 - Preguntar por el NIF del cliente y mostrar sus datos.
 - Mostrar lista de todos los clientes de la base datos con su NIF y nombre.
 - Mostrar la lista de clientes preferentes de la base de datos con su NIF y nombre.
 - Terminar el programa.
49. Escribir un programa que cree un diccionario de traducción español-inglés. El usuario introducirá las palabras en español e inglés separadas por dos puntos, y cada par <palabra>:<traducción> separados por comas. El programa debe crear un diccionario con las palabras y sus traducciones. Después pedirá una frase en español y utilizará el diccionario para traducirla palabra a palabra. Si una palabra no está en el diccionario debe dejarla sin traducir.
50. Escribir un programa que cree un diccionario simulando una cesta de la compra. El programa debe preguntar el artículo y su precio y añadir el par al

diccionario, hasta que el usuario decida terminar. Después se debe mostrar por pantalla la lista de la compra y el coste total, con el siguiente formato:

Lista de la compra	
Artículo 1	Precio
Artículo 2	Precio
Artículo 3	Precio
...	...
Total	Coste