

Ejercicios de programación en C

Parcial 2 – Tipos de datos definidos por el usuario y funciones

1. Escribir un programa que calcule el producto escalar y vectorial de dos vectores de 3 elementos cuyos valores se introducen por pantalla con el programa principal.

2. Realizar un programa que lea 20 números (entre el 1 y el 10) y muestre aquel o aquellos que hayan aparecido más veces.

El programa preguntará si se quieren introducir los 20 números y en el caso en que la respuesta sea negativa rellenará el array con números aleatorios.

Nota: Para la generación de los números aleatorios, se deberán utilizar las funciones `rand`, `srand`, `time` y la constante definida `RAND_MAX`.

`int rand (void):` retorna un valor pseudoaleatorio entre 0 y el valor de la constante `RAND_MAX`.

3. Escribir un programa que pida un arreglo de caracteres por pantalla e invierta el orden de los caracteres mostrándolo por pantalla. La inversión se hará sin utilizar otro array auxiliar.

Nota: La impresión recorre de la posición 0 al tamaño del arreglo -1

4. Escribir un programa que calcule los números primos de 0 a 100 utilizando el llamado método de la criba de Eratóstenes. Este método consiste en definir e inicializar con todos sus elementos a `True` un array de 100 elementos binarios e ir “tachando” (pasando a `False`) en pasadas sucesivas todos los múltiplos de los números primos (2, 3, 5, 7...) hasta obtener sólo los números primos. Es decir:

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 ...

x x x x x x

y

En el ejemplo “x” señalan los múltiplos de 2, mientras “y” los múltiplos de 3 (que no son múltiplos de 2).

Nota: Aumentar el límite superior y comparar la velocidad de ejecución con el programa que calcula los números primos de Sentencias Repetitivas.

5. Realizar un programa que maneje un matriz de enteros a través de un menú con seis opciones:

- 1.- Añadir un elemento (comprobando que no esté lleno)

- 2.- Eliminar un elemento y todas sus repeticiones(comprobando que no esté vacío)

- 3.- Mostrar el contenido en el formato de matriz

- 4.- Contar las apariciones de un número en una columna determinada

- 5.- Terminar

6. Escribir un programa que permita guardar las cuentas de un banco con sus respectivos saldos. Para ello se guardará la información en un array paralelo (dos arrays unidimensionales, uno con los números de cuenta y otro los saldos)
El programa deberá mantener las cuentas ordenadas, de menor a mayor, por número de cuenta para facilitar la búsqueda de una cuenta.
El programa mostrará un menú con las siguientes opciones:
- 1.- Dar de alta una nueva cuenta (comprobando que no esté lleno y colocando la cuenta en la posición correspondiente)
 - 2.- Eliminar una cuenta (comprobando que no esté vacío y reposicionando las cuentas)
 - 3.- Mostrar una cuenta (mostrará el número de cuenta y el saldo correspondiente)
 - 4.- Mostrar información (Número de cuentas dadas de alta y dinero total de todas ellas)
 - 5.- Calcular el saldo medio, máximo y mínimo de las cuentas.
 - 6.- Mostrar todas las cuentas (1 línea por cuenta con su número y su saldo).
 - 7.- Terminar
7. Escribir un programa que rellene automáticamente una matriz 4x7 con la suma de sus índices (Es decir, el elemento $a_{11} = 1+1=2$, $a_{47} = 4+7=11$). El programa mostrará la matriz resultante por pantalla.
8. Escribir un programa que pida una matriz de orden $n \times m$ y muestre con un mensaje si es simétrica. Una matriz es simétrica si $n=m$ y se cumple $a_{ij}=a_{ji}$ para $1 \leq i \leq n$ y $1 \leq j \leq m$.
Nota. Utiliza una función para determinar la simetría esta devuelve un 1 en caso de si y un 0 en caso de no.
9. Crear un programa que permita reservar asientos de una sala de cine (8 filas x 20 columnas). La posición de cada asiento se definirá con una letra (A-H) para la fila y un número (1-20) para la columna. El programa deberá visualizar qué sitios están disponibles para que el cliente pueda decidir dónde sentarse. Antes de hacer la reserva, el programa deberá comprobar que el asiento está libre, en caso contrario devolverá un mensaje de error.

Ej. de visualización:

```
1 . . . 5 . . . . 0 1 . . . 5 . . . . 0

A      *      *      A
B * * * * *      ** B
C  * * * * * * * * C
D *      * * * * * * * D
E  * * * * * * * * E
F * * * * * * * * F
G      * * * * * * * G
H  * * * * * * * * H
1 . . . 5 . . . . 0 1 . . . 5 . . . . 0
```

10. Escriba un programa que lea una matriz de enteros de 3*3 y determine si es un cuadrado mágico.

Nota: Se considera un cuadrado mágico aquel en el cual las filas, columnas y diagonales suman la misma cantidad.

11. Escriba un programa que lea las calificaciones de hasta un máximo de 300 alumnos. Las calificaciones se leen desde teclado y solo permiten un decimal. El programa deberá mostrar dos posibles histogramas según seleccione el usuario: del total de calificaciones o bajo los siguientes intervalos insuficiente(calif menores a 5.5), asesoría (5-6-7), suficiente(8,<9.5), excelente(>=9.5)

Un histograma es un gráfico que muestra la frecuencia con que aparecen valores en un conjunto dado dentro de sub-intervalos especificados. Por ejemplo, si un vector de enteros tiene elementos de tipo 0..9 y contiene los siguientes valores:

6 4 4 1 9 7 5 6 4 2 3 9 5 6 4

Su histograma sería:

	4				*									
Frecuencia	3				*	*								
	2				*	*	*			*				
	1	*	*	*	*	*	*	*	*	*		*		
Valor		0	1	2	3	4	5	6	7	8	9			

Esto indica que los valores 0 y 8 no aparecen en el array, los valores 1, 2, 3 y 7 aparecen una vez, el valor 5 aparece dos veces, el valor 6 tres veces y el valor 4 aparece 4 veces.

Nota: Para la realización del histograma se tendrán en cuenta las dimensiones de la pantalla no permitiéndose más de 22 filas, por tanto si la máxima frecuencia es mayor de 22 se deberá mostrar el histograma a escala.

12. Escribir un programa que calcule la suma de dos matrices A y B. El programa preguntará por las dimensiones de las matrices (dimensiones máximas 5 x 5) y a continuación, si se puede efectuar la suma, introducirá los elementos de ambas matrices y realizará la suma, mostrando finalmente el resultado.

13. Escribir una función que cambie las letras mayúsculas de una cadena a minúsculas y viceversa. El programa principal pedirá una cadena por pantalla y se la pasará a dicha función esperando el resultado correcto que se mostrará por pantalla.

Nota: No podrá hacer uso de las funciones `tolower` y `toupper`, para llevar a cabo el cambio debe tener en cuenta la representación de los caracteres en ASCII. ('A' – 65, 'a' – 97)

14. Escribir un programa que pida primero un carácter por teclado y que luego pida una cadena. El programa calculará cuántos caracteres tiene la cadena hasta que lea el carácter introducido primero.

15. Realizar un programa que lea una cadena de caracteres con espacios en blanco excesivos: elimine los espacios en blanco iniciales y finales y sólo deje uno entre cada dos palabras.

16. Crear un programa que pida una cadena de caracteres y devuelva otra sin signos de puntuación ni números.
17. Crear un programa que pida por pantalla una cadena de 80 caracteres de longitud máxima y que calcule el número de veces que aparece otra cadena determinada, también pedida por teclado.
18. Escribir un programa que compruebe si una cadena pedida por teclado es un palíndromo o no. El programa no tendrá en cuenta si la palabra está escrita con mayúsculas o minúsculas.
Nota 1: Un palíndromo es una palabra que se lee igual al derecho que al revés.
Nota 2: los espacios no forman parte de la revisión
Ej: radar, 11011011, Ana, Otto
19. Escribir una función que compare 2 cadenas de caracteres devolviendo -1 si son iguales y 0 si son distintas.
20. Escribir un programa que pida dos cadenas (de longitud máxima 10 caracteres) y muestre por pantalla el resultado de las siguientes operaciones:
Comparar alfabéticamente ambas cadenas indicando si son iguales o bien cuál es la mayor y cuál la menor.
Concatenar la segunda cadena al final de la primera, dejando un espacio blanco entre ambas.
Copiar el contenido de la segunda cadena en la primera.
Nota: No utilizar las funciones de comparación, concatenación copia de cadenas del ANSI C
21. Escribir un programa (utilizando funciones) con un menú que permita las siguientes opciones:
 1. INTRODUCIR una cadena de 40 caracteres de longitud máxima
 2. CONVIERTE la cadena a formato de oración, es decir, la primer letra en Mayúscula, el resto en minúscula y finalizar con punto.
 3. DDESPLAZA, desplaza n caracteres de cad a la derecha, es decir, dada la cadena de caracteres introducida en el punto 1, obtener otra, de forma que la cadena resultante tenga sus caracteres a una distancia **d** de los caracteres de la cadena original. La distancia **d** se introducirá desde teclado.
 4. IDESPLAZA, desplaza n caracteres de cad a la izq.
 5. Salir**Nota :** En todos los casos si se intenta efectuar alguna operación antes de introducir la cadena se deberá mostrar un mensaje de error comunicando al usuario que la cadena está vacía.
22. Julio Cesar enviaba mensajes a sus legiones encriptando los mensajes mediante el siguiente algoritmo:
Se escogía un número n como clave y se sumaba a cada letra en el alfabeto n posiciones. Así, si la clave escogida fuese 5, la 'a' pasaría a ser la 'f', mientras que la

'f' pasaría a ser la 'k'. Para las últimas letras del abecedario se seguiría desde el principio. Así, con la clave de 5 la 'y' pasaría a ser la 'd'.
Se pide crear un programa que encripte una frase mediante este algoritmo.

23. Construir un programa que implemente una calculadora para números enteros:
El programa pedirá primero la operación y luego los operandos. + 2 4
Las operaciones válidas serán: sumar, restar, multiplicar, dividir, salir.
Si la operación es distinta de salir se pedirán los operandos y luego se mostrará el resultado.

Verifique que los datos ingresados sean validos

Nota: Los operandos se recogerán como cadenas de caracteres y se convertirán en números enteros con la función atoi()

24. Realizar un programa que permita calcular el NIF., conocido el DNI. de una persona.
El programa deberá leer, sobre una cadena de caracteres, el número del DNI. Del interesado. Seguidamente deberá averiguar si es un valor válido (todos los caracteres deben ser numéricos y representar un valor entre 100.000 y 99.999.999). Si la entrada es válida se deberá calcular el NIF. y representar el número completo con los puntos de millares y millones en las posiciones correspondientes, así como la letra del NIF al final de la cadena de caracteres separada por un espacio en blanco.

Nota: Para el cálculo de la letra del NIF:

Se obtiene el resto de la división del número del DNI, entre 23, y en función del resultado se asigna un carácter según la siguiente tabla:

0 = 'T'	7 = 'F'	14 = 'Z'	21 = 'K'
1 = 'R'	8 = 'P'	15 = 'S'	22 = 'E'
2 = 'W'	9 = 'D'	16 = 'Q'	
3 = 'A'	10 = 'X'	17 = 'V'	
4 = 'G'	11 = 'B'	18 = 'H'	
5 = 'M'	12 = 'N'	19 = 'L'	
6 = 'Y'	13 = 'J'	20 = 'C'	

25. Escribir una función en el que dados los numeradores y denominadores de dos fracciones devuelva la suma de estos.

Considere que

$$\frac{n1}{d1} + \frac{n2}{d2} = \frac{n1 * d2 + n2 * d1}{d1 * d2} = \frac{nR}{dR}$$

Nota 1. Resuelva primero enviando la información como parámetros (2 numeradores y dos denominadores) y muestre el resultado directamente en el modulo

Nota 2. Resuelva nuevamente el ejercicio pero ahora la información estará dentro de una estructura fracción (numerador, denominador) y devuelva el resultado en una estructura similar.

26. Tal vez el esquema de codificación más famoso de todos es el código Morse, desarrollado por Samuel Morse en 1832 para el sistema telegráfico. El código Morse asigna una serie de puntos y rayas a cada letra del abecedario, a cada dígito y a algunos caracteres especiales (Ver tabla adjunta).

Escribir un programa que lea una frase y la codifique en código Morse. También escriba un programa que lea una frase en código Morse y la convierta en su equivalente en castellano. La separación entre letras se indicará mediante “/”, mientras que la separación entre palabras se indicará mediante “//”.

Carácter	Código	Carácter	Código	Carácter	Código
A	.-	B	-...	C	-.-.
D	-..	E	.	F	..-.
G	--.	H	I	..
J	.---	K	-.-	L	.-..
M	--	N	-.	Q	--.-
O	---	P	.-.-	T	-
R	.-.	S	...	W	.-.-
U	..-	V	...-	Z	--..
X		Y	-.--		
1	.----	2	..---	3	...--
4-	5	6	-....
7	--...	8	---..	9	----.
0	-----	?	..--..	,	--..--

27. Escribir un programa que permita convertir grados Fahrenheit a Celsius y grados Celsius a Fahrenheit. El programa presentará el siguiente menú:

1. Conversión de Celsius a Fahrenheit
2. Conversión de Fahrenheit a Celsius
3. Salir del programa.

Nota: Cada conversión se efectuará por medio de funciones, una que convertirá de grados Celsius a grados Fahrenheit y otra que haga justo lo contrario.

28. Realizar un programa que escriba todos los números enteros menores que un cierto entero N y que a su vez sean múltiplos de dos números enteros A y B introducidos por teclado.

Nota: Utilizar para ello una función que admita dos parámetros I, J e indique si I es múltiplo de J.

29. Escribir una función (con su correspondiente programa de prueba) que tome un valor entero y devuelva el número con sus dígitos en reversa. Por ejemplo, dado el número 7631 la función deberá devolver 1367.

30. Escribir un programa que calcule masa radioactiva de carbono 14 que queda después de **t** años. La fórmula que determina la masa restante en el tiempo **t** es

$$M_t = m * \left(\frac{1}{2}\right)^{\frac{t}{h}}$$

donde: t es el tiempo en años

Mt es la masa que permanece en el tiempo t

m es la masa original

h es la vida media en años

Para el carbono 14 la vida media es de 5700 años; si la masa original es de 300 gramos, imprimir una tabla de la masa para $t = 500, 1000, 1500, 2000, \dots, 10000$ años.

Nota : Se deberá utilizar un subprograma que permita evaluar la expresión $(1/2)^{t/h}$ para los diferentes valores de t .

31. Escribir una función que tome el tiempo introducido por el usuario en el formato (horas:minutos:segundos) y lo convierta en segundos. El programa utilizará esta función para calcular la diferencia en segundos entre dos tiempos introducidos por el usuario.

Nota : No utilizar la librería `time.h`.

32. Escribir un programa que calcule el número combinatorio

$$\binom{M}{N} = \frac{M!}{N!(M-N)!}$$

Utilizar para ello una función que calcule el factorial de un número.

Nota: La función para calcular el factorial de un número puede ser iterativa o recursiva. (Una función recursiva es una función que se llama a sí misma)

33. Escribir una función que calcule el factorial de un número y utilizar ésta en un programa que muestre el siguiente menú.

1. Cálculo de e
2. Cálculo de e^x
3. Salir

Nota 1: El cálculo de e debe hacerse con la siguiente expresión matemática:

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots$$

Nota 2: e^x puede calcularse mediante la fórmula:

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

Nota 3: La precisión con la que se obtiene el resultado (e o e^x) depende del último valor añadido en la correspondiente serie.

34. El desarrollo en serie de Taylor de la función coseno es:

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$$

donde el ángulo x viene expresado en radianes.

Escribir un programa que calcule el valor aproximado de coseno (x), utilizando para ello los N primeros términos de la serie de Taylor. El número de términos N de la serie dependerá del valor del último, es decir, se añadirán términos a la serie hasta que el valor absoluto del último término añadido sea menor que 0.0005.

35. Se desea realizar un programa que permita hacer cálculos con polinomios de segundo grado. El programa deberá presentar un menú en pantalla que facilite las siguientes opciones:

1. Leer un polinomio
2. Escribir un polinomio en su forma habitual

3. Evaluar un polinomio en un punto
4. Calcular el polinomio derivado
5. Salir

Se deberán utilizar los siguientes subprogramas:

LEER mediante este subprograma se introducen los números enteros que representan los coeficientes del polinomio.

ESCRIBE es un subprograma que permite la escritura de un polinomio en la forma:

$$Ax^2 + Bx + C$$

Ejemplos válidos: $5x^2+3x+5$ $5x^2-3x+5$ $5x^2-5$

Ejemplos no válidos: $5x^2+ -3x+5$ $5x^2-0x+5$

EVALUA: es un subprograma que retorna el valor de un polinomio para un número real que es introducido como parámetro.

DERIVAR es un subprograma que calcula el polinomio derivada de uno dado.

Nota: No está permitido el uso de variables globales.

36. Escribir un programa que realice la descomposición en factores primos de un número introducido por teclado.

El programa deberá ir escribiendo la tabla de los factores primos, a medida que los va calculando, tal como muestra el ejemplo siguiente:

Introduce un N° entero -> **84**

N° | Factores primos

-- | -----

84 | 2

42 | 2

21 | 3

7 | 7

1 |

Nota: Deberá utilizar una función que nos diga si un número es primo o no.

37. Escribir un programa que cuente de un texto introducido por teclado:

Longitud del texto

N.º de dígitos

N.º de letras

N.º de otros caracteres

Nota: Se deben crear las funciones para cada caso, es decir no puede utilizar isdigit, isalpha, strlen, etc..

38. Escribir un programa que lea dos números complejos y permita realizar con ellos las siguientes operaciones aritméticas: suma, resta, multiplicación y división

Nota 1: Se debe crear una función de permita leer un número complejo (su parte real y su parte imaginaria).

Nota 2: La suma y resta de números complejos se obtiene sumando, o restando, las partes reales y las partes complejas. El producto de dos números complejos se obtiene multiplicando sus módulos y sumando sus argumentos. El cociente de dos números complejos se obtiene dividiendo sus módulos y restando sus argumentos.

39. Escribir un programa que emplee un argumento de la línea de comandos para realizar una conversión decimal a hexadecimal; es decir, el número decimal se introducirá en la línea de comandos, siguiendo al nombre del programa. Ejemplo:

C:> decihex 128 111

Deci= 128 Hex= 80

Deci= 111 Hex= 6F

C:>

Nota 1: Al igual que en el ejemplo, el programa deberá ser capaz de convertir varios números en una llamada.

Nota 2: En caso de que el programa no reciba argumentos deberá devolver un mensaje de error.

Nota 3: La función atoi() convierte de cadenas a enteros.

40. Escribir un programa que permita al usuario realizar las siguientes operaciones:

1. Mostrar la fecha y hora por pantalla
2. Sacar por pantalla el contenido de un fichero ASCII.
3. Sacar por pantalla el contenido de un directorio
4. Limpiar la pantalla
5. Salir del programa

Para ello se hará uso de los comandos del sistema operativo:

En Linux: date, cat, ls, clear

En DOS: date, time, type, dir, cls

Nota: indicar como comentario el S.O utilizado

41. Escriba las funciones que permitan calcular el valor de la constante π a partir de las siguientes series matemáticas, no olvide anexar el programa de prueba correspondiente.

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

$$\pi = \sqrt{\sum_{i=1}^{\infty} \frac{24}{(i)^2}} = \sqrt{\left(24 + \frac{24}{2^2} + \frac{24}{3^2} + \frac{24}{4^2} + \dots \right)}$$

$$\pi = 4 * \frac{2}{3} * \frac{4}{3} * \frac{4}{5} * \frac{6}{5} * \frac{6}{7} * \frac{8}{7} * \dots$$

42. Escriba una función y el programa de prueba que dado un parámetro $<>0$ devuelva el valor de "y", donde "y" se calcula

$$y = \frac{1}{x^4 \left(\frac{e^{1.325}}{x} - 1 \right)}$$

43. Realizar un programa que permita realizar las operaciones básicas (sumar, restar, multiplicar y dividir) números complejos.

El programa deberá utilizar una variable que represente el número complejo en su forma polar, con sus dos componentes módulo y argumento.

44. Crear un programa que lea las siguientes variables proporcionadas desde teclado con el siguiente formato: BI6755CC140075----FIESTAFORD, donde,

Posición de los caracteres	Campo
1-8	Matricula
9	Cilindros
10-12	Potencia
12-14	Año
15-24	Modelo
25-34	Marca

Y las introduzca en la correspondiente estructura. El programa deberá mostrar la estructura obtenida para comprobar que la conversión ha sido correcta.

Nota: Construir una función que muestre por pantalla la estructura recibida. La estructura deberá pasarse por referencia.

45. Disponemos de la información correspondiente a una jaula de un Zoo en una variable de tipo registro con los siguientes campos:

Numero de jaula Entero
Especie del animal Cadena de caracteres
Nombre del animal Cadena de caracteres
Edad Entero
Peso Real
Kilogramos de comida diaria Real
Frecuencia de limpieza de jaula Entero (veces al día)
Estado de la salud del animal Carácter (B, R, M -> Buena, Regular o Mala)
Descendencia Sí o No
Peligroso Sí o No

Por motivos de transferencia de información a otros organismos necesitamos descomponer la información contenida en esa variable en dos variables diferentes:

Una va a contener los datos de mantenimiento de la jaula del animal: Numero de jaula, kilogramos de comida diaria real, frecuencia de limpieza de jaula, peligroso. Otra, los datos del animal: Especie del animal, nombre del animal, edad, peso, estado de salud del animal y descendencia.

Escribir un programa que lea una variable de información global y la descompongan en dos variables, una de información de mantenimiento y otra de información del animal y visualice ambas variables.

Nota: Dentro de los subprogramas que considere necesarios deberá haber uno, llamado descomponer, que tome un registro y lo descomponga en dos.

46. Crear un programa que permita introducir cierta información relativa a los vuelos diarios que parten de un aeropuerto en un arreglo formado por registros. Cada registro contendrá la siguiente información sobre el vuelo correspondiente:

Número de vuelo (No tiene por qué coincidir con el índice del arreglo)

Hora de partida (En dos campos): Hora: 0..23 y Minutos: 0..59

Origen del vuelo: Cadena de caracteres

Destino del vuelo: Cadena de caracteres

Número de pasajeros: Entero

Una vez introducidos los datos de todos los vuelos se preguntará si se desea obtener información de algún vuelo. En caso de que el usuario responda afirmativamente se pedirá el número de vuelo. El programa buscará el vuelo y accederá a la información que contiene a partir de su número de vuelo, mostrando por pantalla todos sus datos.

El programa se ejecutará repetitivamente hasta que el usuario indique que no desea obtener más información de ningún vuelo.

Nota: Se deben diseñar las funciones que visualicen un vuelo, busquen un vuelo en el arreglo, introduzcan la información de un vuelo en el array, etc...

47. Una compañía utiliza aviones para fumigar las cosechas contra una gran variedad de plagas. Lo que la compañía cobra a los granjeros depende de contra qué es lo que desean fumigar, y de cuantos m2 de tierra quieren que se fumiguen de acuerdo con la siguiente tabla:

Tipo1: Fumigación contra malas hierbas 18 € / m2

Tipo2: Fumigación contra langostas 36 € / m2

Tipo3: Fumigación contra gusanos 54 € / m2

Tipo4: Fumigación contra todo lo anterior 90 € / m2

Además, si el área a fumigar es mayor que 10.000 m2, el granjero goza de un descuento del 7%.

Se trata de escribir un programa que lea los datos de un conjunto de granjeros y al final calcule, para cada uno de ellos, la factura correspondiente. De cada granjero se tendrá la siguiente información:

Nombre

Tipo de fumigación (código entre 1 y 4)

Número de m2 que se van a fumigar.

Fecha de factura (a su vez, con los componentes: minutos, hora, día, mes y año)

Nota: Los componentes de la fecha de la factura deben ser leídos en el sistema(utilizando funciones de la librería time.h)

48. Escriba un programa que genere un vector de 50 números enteros entre 1 y 1000 de forma aleatoria. A continuación, partiendo de ese vector genere otro de 50 estructuras tal que cada estructuras tenga un campo para el número y otro (cadena de caracteres) para indicar si dicho número es primo o no. Por ejemplo:

203 672 23 319 426 1 862 373 273 203

203 672 23 319 426 1 862 373 273 203

NO NO SI NO NO SI NO SI NO NO

Finalmente muestre en pantalla la suma de todos los números del array que sean primos.

49. Escribir un programa que cree una base de datos temporal sobre el personal de agentes de policía. La base de datos almacenará cuatro datos acerca de cada persona:

- Nombre (Array de caracteres)
- Apellido (Array de caracteres)
- Código (Entero)
- Categoría (Entero: 0: detective, 1: ayudante, 2: director)

El programa preguntará acerca de cuántos datos se añadirán a la base de datos y luego mostrará los datos de todos los agentes introducidos.

50. Programa que implemente una agenda. Se guardarán los siguientes datos por persona:

- Nombre: Máximo 15 caracteres.
- Apellidos: Máximo 35 caracteres.
- Sobrenombre: Máximo 10 caracteres.
- Teléfono: Máximo 10 caracteres.
- Fecha de nacimiento: 8 caracteres (con el formato dd/mm/aa, pudiendo ser espacios los dígitos del año)

Se creará un menú con las siguientes opciones:

1. Alta de una nueva persona a la agenda con los correspondientes datos.
2. Eliminar a una persona de la agenda.
3. Búsqueda de un nombre en particular (La búsqueda se hará por sobrenombre)
4. Listado de todas las personas empezando por la primera introducida (Para pasar a la siguiente se deberá pulsar intro).
0. Salir

Nota. Valide por medio de funciones que los elementos ingresados en numero sean dígitos y que la cadena recibida en fecha acepte el formato establecido.

51. Se deberá escribir un programa que permita introducir y consultar la tabla periódica de los elementos químicos. Para ello, se construirá un menú con las siguientes opciones:

1. Introducir elementos de la tabla periódica. Preguntará cuántos elementos de la tabla periódica se quieren introducir e irá preguntando sucesivamente por ellos
2. Listar todos los elementos de tabla periódica. Muestra la información acerca de todos los elementos químicos introducidos ordenados por número atómico
3. Mostrar elemento de la tabla periódica por número atómico, pedirá el número atómico del elemento que se quiere consultar y mostrará por pantalla la información correspondiente a dicho elemento
4. Mostrar elemento de la tabla periódica por símbolo, pedirá el símbolo del elemento que se quiere consultar y mostrará por pantalla la información correspondiente a dicho elemento
5. Salir

Se guardará la siguiente información sobre cada elemento.

1. Símbolo del elemento
2. Nombre completo
3. Peso Atómico