

Práctica 1. Introducción a C

PI.1. Escribe un programa que lea una medida de tiempo expresada en horas, minutos y segundos con valores arbitrarios. El programa debe transformar esa medida en una expresión correcta.

Ejemplo: Introduce un tiempo en horas minutos y segundos: 3 118 195
El tiempo introducido es 5:1:15

PI.2. Escribe un programa que pida una fecha (día mes y año) y devuelva la fecha correspondiente al día siguiente (en este ejercicio suponemos que todos los años el mes de febrero tiene 28 días).

Ejemplo: Introduce una fecha (día mes año): 31 3 1056
Mañana es día 1/4/1056

PI.3. Diseña un programa que halle la descomposición en números primos de un valor entero.

Ejemplo: Introduce un numero entero: 252
252 se descompone en $2 \times 2 \times 3 \times 3 \times 7$

PI.4. Escribe un programa que lea números enteros por teclado hasta que se introduzca un cero. El programa debe devolver la suma de todos aquellos números introducidos que sean positivos.

Ejemplo: Introduce un numero: 1
Introduce un numero: -2
Introduce un numero: 3
Introduce un numero: 0
La suma de los numeros positivos es 4

PI.5. Escribe un programa que presente un menú en pantalla y lea una opción repitiendo el proceso indefinidamente hasta que la opción sea no válida.

Ejemplo: Elige una opcion:
1 – Opcion 1
2 – Opcion 2
3 – Opcion 3
1
Has elegido la opcion 1
Elige una opcion:
1 – Opcion 1
2 – Opcion 2
3 – Opcion 3
3
Has elegido la opcion 3
Elige una opcion:
1 – Opcion 1
2 – Opcion 2
3 – Opcion 3
5

PI.6. Escribe un programa que lea números enteros por teclado hasta que se introduzca un número negativo. El programa debe mostrar cuál es el número máximo y cuántas veces se ha introducido.

Ejemplo: Introduce un numero: 8
Introduce un numero: 1
Introduce un numero: 8
Introduce un numero: -5
El numero mayor es 8 y se ha introducido 2 veces

PI.7. Para un valor entero de entrada, escribe un programa que dibuje en pantalla un gráfico como el del ejemplo.

Ejemplo: Introduce un numero entero: 4
1
12
123
1234

PI.8. Escribe un programa que multiplique dos números enteros utilizando la técnica de la multiplicación rusa. En esta técnica, el primero de los elementos a multiplicar se divide entre dos a la vez que el segundo número se multiplica por dos hasta que el primer número llegue a valer 1. El resultado final se obtiene sumando todos aquellos segundos multiplicandos que se correspondan con un primer multiplicando impar.

Por ejemplo, si multiplicamos 25 por 14
25 14 → Como 25 es impar, sumamos 14
12 28
6 56
3 112 → Como 3 es impar, sumamos 112
1 224 → Como 1 es impar, sumamos 224
La suma total es 14+112+224=350

Ejemplo: Introduce dos números enteros: 25 14
La multiplicación es 25 x 14 = 350

PI.9. Un trío pitagórico se define como un conjunto de tres números: a, b y c que cumplen con la relación $a^2 + b^2 = c^2$. Escribe un programa que, dado un número entero, devuelva todos los tríos pitagóricos existentes tales que sus tres números son menores que el introducido. Tener en cuenta que el trío (3,4,5) es igual que (4,3,5) por lo que sólo debe imprimirse una vez.

Ejemplo: Introduce un numero entero: 15
(3,4,5) (5,12,13) (6,8,10)

PI.10. Un tablero de ajedrez es una cuadrícula de 8 filas y 8 columnas. Dos de las piezas del juego son el alfil y la torre. El alfil se desplaza en diagonal mientras que la torre lo hace en horizontal o en vertical. Una pieza puede ser “capturada” por otra si está en una casilla a la cual la otra puede moverse.

Escribe un programa que pida las posiciones en el tablero de un alfil y una torre e indique qué ficha puede capturar a la otra. Suponemos que las posiciones introducidas son válidas.

Ejemplo: Fila alfil: 7

Columna alfil: 6
Fila torre: 4
Columna torre: 3
El alfil captura a la torre

Ejemplo: Fila alfil: 3
Columna alfil: 3
Fila torre: 8
Columna torre: 5
Ninguna ficha captura a la otra

PI.11. Escribe un programa que encuentre y escriba por pantalla el elemento mayor y menor de una matriz de 4x4 posiciones, así como las posiciones que ocupan.

Ejemplo: Introduce los números de la matriz: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
El elemento mayor es 16 y esta en la posición (4,4)
El elemento menor es 1 y esta en la posición (1,1)

PI.12. Escribe un programa que calcule y visualice la media de un vector de 10 enteros.

Ejemplo: Introduce 10 enteros: 5 8 4 2 6 3 1 7 8 4
La media de todos los valores es 4.8

PI.13. Diseña un programa que lea una cadena de caracteres y la almacene en un vector. Posteriormente debe imprimirla, poniendo en mayúsculas el primer carácter de la misma así como todo carácter que siga a un punto. No hace falta almacenar la nueva cadena.

Ejemplo: Introduce una cadena de caracteres: hola.me llamo.maria
Hola.Me llamo.Maria

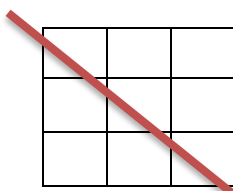
PI.14. Escribe un programa que lea elementos enteros en una matriz de 3x3 posiciones. Luego debe cambiar los elementos de la matriz, consiguiendo que sea simétrica respecto a la diagonal principal. Finalmente debe imprimirla con formato de matriz.

Ejemplo: Introduce 9 numeros enteros: 1 2 3 4 5 6 7 8 9
La matriz simétrica es:
1 4 7
2 5 8
3 6 9

Explicación: La matriz original es

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

La simétrica con respecto a la diagonal principal es:



| | | |
|---|---|---|
| 1 | 4 | 7 |
| 2 | 5 | 8 |
| 3 | 6 | 9 |

P1.15. Escribe un programa que cree un vector de 10 posiciones y lo rellene con números aleatorios entre 11 y 50. Para ello utiliza la función `rand()` (y adicionalmente la función `srand` si quieres que los valores sean distintos en cada ejecución). Debe mostrar los números almacenados. Posteriormente, organizar los números del vector de tal forma que todos los pares estén juntos y después todos los números impares. Mostrar otra vez el vector por pantalla.

Ejemplo: El vector inicial es:

15 41 32 49 17 22 26 36 31 45

El vector reordenado es:

36 26 32 22 17 49 41 15 45

P1.16. Escribe un programa que pida al usuario el número de datos que quiere introducir. Posteriormente reservar memoria para una tabla de ese tamaño (dinámicamente) y almacenar los datos que introduzca el usuario. Finalmente, se deben mostrar por pantalla los elementos almacenados.

Ejemplo: Cuantos datos quieres introducir: 4

Introduce 4 enteros: 4 8 2 1

Los datos introducidos son 4 8 2 1

P1.17. Escribe un programa que pida al usuario un número entero (n). Si el número es par, crear un matriz cuadrada de $(n+1)$ elementos de lado. Si el número es impar, crear una matriz cuadrada de n elementos de lado. Rellenar la matriz como en los ejemplos mostrados. Finalmente, se deben mostrar por pantalla los elementos almacenados.

$n=3$

| | | | | |
|---|---|---|---|---|
| * | * | * | * | * |
| * | | | | * |
| * | | * | | * |
| * | | | | * |
| * | * | * | * | * |

$n=8$

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| * | * | * | * | * | * | * | * | * |
| * | | | | | | | | * |
| * | | * | * | * | * | * | | * |
| * | | * | | | | * | | * |
| * | | * | | * | | * | | * |
| * | | * | | | | * | | * |
| * | | * | * | * | * | * | | * |
| * | | | | | | | | * |
| * | * | * | * | * | * | * | * | * |

Ejemplo: Introduce un numero: 6

| | | | | | | |
|---|---|---|---|---|---|---|
| * | * | * | * | * | * | * |
| * | | * | * | * | * | * |
| * | | * | * | * | * | * |
| * | | * | * | * | * | * |
| * | | * | * | * | * | * |
| * | | * | * | * | * | * |
| * | * | * | * | * | * | * |

PI.18. Hacer un programa que pida al usuario las dimensiones de la matriz que quiere crear (filas y columnas). A partir de ellos, se debe crear un array bidimensional de las longitudes introducidas cuyo nombre sea 'suma'. Posteriormente, se le deben asignar valores de tal forma que cada elemento contenga la suma del número de su fila y del número de su columna (tener en cuenta que las filas y las columnas comienzan en 0, por lo que el primer elemento debe tener como valor 0). Por último, mostrar el contenido de la matriz por pantalla.

Ejemplo: Introduce las dimensiones de la matriz: 3 4

| | | | |
|---|---|---|---|
| 0 | 1 | 2 | 3 |
| 1 | 2 | 3 | 4 |
| 2 | 3 | 4 | 5 |

PI.19. Hacer un programa que pida al usuario las dimensiones de dos matrices. Rellenarlas con valores pedidos al usuario. Posteriormente, crear otra matriz con tantas filas como el máximo de filas de las anteriores y tantas columnas como el máximo de columnas de las anteriores. Rellenar los valores de esta nueva matriz, de tal forma que los elementos que tengan valor en las dos matrices anteriores, tomen el valor de la suma. Los elementos que no existiesen en la matriz 1, toman valor 2; los elementos que no existiesen en la matriz 2, toman valor 1 y los elementos que no existiesen en ninguna matriz, toman valor 0. Por último, mostrar el contenido de la matriz por pantalla.

Ejemplo: Introduce las dimensiones de la matriz 1: 3 5

Introduce las dimensiones de la matriz 2: 6 2

Introduce los elementos de la matriz 1:

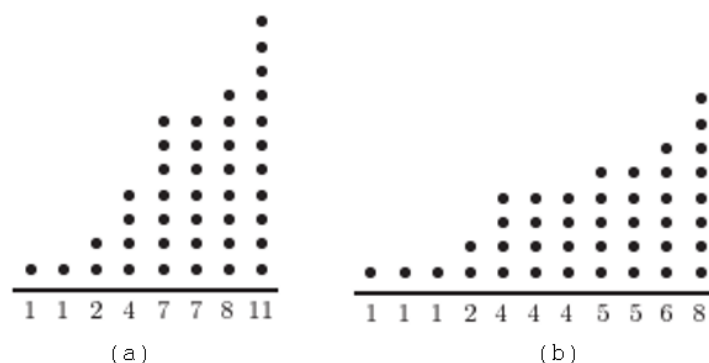
2 5 3 1 0 8 4 7 5 1 2 6 3 2 5

Introduce los elementos de la matriz 2:

5 8 4 7 6 2 5 9 0 1 2 4

| | | | | |
|----|----|---|---|---|
| 7 | 13 | 1 | 1 | 1 |
| 12 | 11 | 1 | 1 | 1 |
| 8 | 8 | 1 | 1 | 1 |
| 2 | 2 | 0 | 0 | 0 |
| 2 | 2 | 0 | 0 | 0 |
| 2 | 2 | 0 | 0 | 0 |

PI.20. Consideramos un "histograma" entero no decreciente representado mediante una lista en la que se almacenan las alturas de sus columnas (figura a). Si hacemos que las columnas pasen a ser filas y viceversa, se obtienen el histograma y la figura representativa de figura b.



Escribe un programa que pida al usuario la lista del histograma inicial y devuelva la lista de su histograma transpuesto.

Ejemplo: Introduce el numero de elementos del histograma: 8
Introduce los elementos: 1 1 2 4 7 7 8 11
El histograma transpuesto es:
1 1 1 2 4 4 4 5 5 6 8

PI.21. Tenemos el siguiente problema: m personas están sentadas en círculo, y son ejecutadas contando cada n personas. La persona que queda última es la que sobrevive. Por ejemplo, si tenemos 12 personas y ejecutamos contando cada 3 personas, el orden de ejecuciones será: 3, 6, 9, 12, 4, 8, 1, 7, 2, 11, 5, siendo la persona que se salva la número 10.

Ejemplo: Introduce el numero de personas: 12
Introduce cada cuantos se produce una ejecucion: 3
La persona que se salva es la numero 10

PI.22. Introduce por teclado una tabla de estructuras que contenga los nombres y las notas de teoría y de prácticas de 5 alumnos y saque por pantalla un listado de los aprobados. Se consideran aprobados aquellos alumnos cuya media entre teoría y práctica sea de al menos 5 puntos.

Ejemplo: Introduce el nombre nota de teoría y nota de practicas: Juan 6 8
Introduce el nombre nota de teoría y nota de practicas: Maria 3 5
Introduce el nombre nota de teoría y nota de practicas: Antonio 3 6
Introduce el nombre nota de teoría y nota de practicas: Pepe 8 3
Introduce el nombre nota de teoría y nota de practicas: Ana 4 9
Los alumnos aprobados son: Juan Pepe Ana

PI.23. Escribe un programa que pida al usuario el número de polígonos que quiere introducir. De cada uno de ellos debe almacenar el número de lados y si es regular o no. Finalmente, imprimir por pantalla el número de lados de los polígonos, mostrando primero aquellos que sean regulares y después los irregulares.

Ejemplo: Introduce el numero de poligonos: 4
Introduce el numero de lados y 0 si es regular, 1 si es irregular: 6 1
Introduce el numero de lados y 0 si es regular, 1 si es irregular: 4 0
Introduce el numero de lados y 0 si es regular, 1 si es irregular: 3 0
Introduce el numero de lados y 0 si es regular, 1 si es irregular: 4 1
Los polígonos regulares tienen 4, 3 lados
Los polígonos irregulares tienen 6, 4 lados

PI.24. Escribir un programa que haga una llamada a la función absoluto y muestre por pantalla la solución de ésta. La función absoluto debe calcular y devolver el valor absoluto de un valor entero introducido como argumento.

Ejemplo: Introduce un numero: -5
El valor absoluto de -5 es 5

PI.25. Escribir un programa que haga una llamada a la función multiplo. La función multiplo recibe como argumento dos números enteros y escribe por pantalla verdadero si el primero es múltiplo del segundo, y falso si no lo es.

Ejemplo: Introduce dos números enteros: 10 5

10 es múltiplo de 2

PI.26. Escribe un programa que pida al usuario el tamaño de un vector y los números enteros necesarios para rellenarlo. Posteriormente debe llamar a una función que ordene los elementos según su valor absoluto. Finalmente, el programa principal debe mostrar los valores ordenados.

Ejemplo: Introduce el tamaño del vector: 7

Introduce los elementos: 8 -2 6 3 -4 0 -7

Los elementos ordenados son: 0 -2 3 -4 6 -7 8

PI.27. Escribe un programa que llame a una función recursiva que muestre en pantalla los dígitos de un entero positivo en orden inverso.

Ejemplo: Introduce un número entero: 10523

32501

PI.28. Escribe un programa que llame a una función recursiva para calcular un número combinatorio. Éste se debe calcular mediante la siguiente ecuación

$$comb(x, y) = \begin{cases} 1 & \text{si } y = 0 \text{ o } x = y \\ comb(x-1, y-1) + comb(x-1, y) & \text{en otro caso} \end{cases}$$

Ejemplo: Introduce los valores del número combinatorio: 5 3

El valor es 10

PI.29. Escribe un programa que calcule la división de dos números de forma recursiva. (Suponemos que el primer número es mayor que el segundo, y que es un múltiplo, es decir, el resto de la división siempre es 0. Estas condiciones no hace falta comprobarlas).

Ejemplo: Introduce el dividendo y el divisor: 10 2

La división es 5

PI.30. Escribe un programa que llame a una función recursiva que calcule el máximo común divisor de dos enteros positivos: m y n. Si m es mayor que n, la función recursiva puede ser de la forma:

$$mcd(m, n) = \begin{cases} m & \text{si } n = 0 \\ mcd(n, m \bmod n) & \text{si } n > 0 \end{cases}$$

Ejemplo: Introduce dos números enteros: 40 180

El mcd de 40 y 180 es 20

PI.31. Un granjero ha comprado una pareja de conejos para criarlos y luego venderlos. La pareja de conejos produce una nueva pareja cada mes y la nueva pareja tarda un mes más en ser también productiva. Escribe un programa que pida al usuario el número de un mes y devuelva, mediante una llamada a una función recursiva, el número de conejos que tendrá el granjero en ese mes.

Ejemplo: Introduce el mes: 3

El granjero tiene 8 parejas de conejos

P1.32. Escribe un programa que, dado un vector de números enteros introducidos por el usuario, calcule mediante una función recursiva el máximo de dicho vector.

Ejemplo: Introduce el numero de elementos del vector: 7

Introduce los elementos del vector: 5 8 1 4 6 3 2

El máximo de los elementos es 8

P1.33. Escribe un programa que pida un conjunto de números enteros al usuario, y otro número entero a buscar. Mediante una función recursiva, el programa debe devolver el número de veces que aparece este último elemento en la lista.

Ejemplo: Introduce el numero de elementos del vector: 8

Introduce los elementos: 2 4 4 6 1 2 4 3

Introduce el elemento a buscar: 4

El numero 4 aparece 3 veces en el vector

P1.34. Dada la siguiente función recursiva, estúdiala para comprender su funcionamiento. Posteriormente, crea una función iterativa equivalente. Escribe un programa principal que compruebe que ambas funciones se comportan de la misma manera.

```
void mifuncion (int v[], int i){
    int aux;
    if (i>0){
        aux = v[0];
        v[0] = v[i];
        v[i] = aux;
        mifuncion (v, i-1);
    }
}
```

P1.35. Dada la siguiente función recursiva, estúdiala para comprender su funcionamiento. Posteriormente, crea una función iterativa equivalente. Escribe un programa principal que compruebe que ambas funciones se comportan de la misma manera.

```
void mifuncion (int v[], int j, int k){
    int aux;
    if (j<k){
        mifuncion (v, j+1, k-1);
        aux = v[j];
        v[j] = v[k];
        v[k] = aux;
    }
}
```


P1.36. Dada la siguiente función recursiva, estúdiala para comprender su funcionamiento. Posteriormente, crea una función iterativa equivalente. Escribe un programa principal que compruebe que ambas funciones se comportan de la misma manera.

```
void mifuncion (int v[], int N, int pos){
    int aux;
    if (2*pos < N){
        aux = v[N-pos-1];
        v[N-pos-1] = v[pos];
        v[pos] = aux;
        mifuncion (v, N, pos+1);
    }
}
```