

Ejercicios estructura secuencial

Ejercicio 1

Escribir un programa que pregunte al usuario su nombre, y luego lo salude.

Ejercicio 2

Calcular el perímetro y área de un rectángulo dada su base y su altura.

Ejercicio 3

Dados los catetos de un triángulo rectángulo, calcular su hipotenusa.

Ejercicio 4

Dados dos números, mostrar la suma, resta, división y multiplicación de ambos.

Ejercicio 5

Escribir un programa que convierta un valor dado en grados Fahrenheit a grados Celsius.

Recordar que la fórmula para la conversión es:

$$C = (F - 32) * 5 / 9$$

Ejercicio 6

Calcular la media de tres números pedidos por teclado.

Ejercicio 7

Realiza un programa que reciba una cantidad de minutos y muestre por pantalla a cuantas horas y minutos corresponde.

Por ejemplo: 1000 minutos son 16 horas y 40 minutos.

Ejercicio 8

Un vendedor recibe un sueldo base mas un 10% extra por comisión de sus ventas, el vendedor desea saber cuanto dinero obtendrá por concepto de comisiones por las tres ventas que realiza en el mes y el total que recibirá en el mes tomando en cuenta su sueldo base y comisiones.

Ejercicio 9

Una tienda ofrece un descuento del 15% sobre el total de la compra y un cliente desea saber cuanto deberá pagar finalmente por su compra.

Ejercicio 10

Un alumno desea saber cual será su calificación final en la materia de Algoritmos. Dicha calificación se compone de los siguientes porcentajes:

- 55% del promedio de sus tres calificaciones parciales.
- 30% de la calificación del examen final.
- 15% de la calificación de un trabajo final.

Ejercicio 11

Pide al usuario dos números y muestra la “distancia” entre ellos (el valor absoluto de su diferencia, de modo que el resultado sea siempre positivo).

Ejercicio 12

Pide al usuario dos pares de números x_1, y_1 y x_2, y_2 , que representen dos puntos en el plano. Calcula y muestra la distancia entre ellos.

Ejercicio 13

Realizar un algoritmos que lea un número y que muestre su raíz cuadrada y su raíz cúbica. Python3 no tiene ninguna función predefinida que permita calcular la raíz cúbica, ¿Cómo se puede calcular?

Ejercicio 14

Dado un número de dos cifras, diseñe un algoritmo que permita obtener el número invertido. Ejemplo, si se introduce 23 que muestre 32.

Ejercicio 15

Dadas dos variables numéricas A y B, que el usuario debe teclear, se pide realizar un algoritmo que intercambie los valores de ambas variables y muestre cuanto valen al final las dos variables.

Ejercicio 16

Dos vehículos viajan a diferentes velocidades (v_1 y v_2) y están distanciados por una distancia d . El que está detrás viaja a una velocidad mayor. Se pide hacer un algoritmo para ingresar la distancia entre los dos vehículos (km) y sus respectivas velocidades (km/h) y con esto determinar y mostrar en que tiempo (minutos) alcanzará el vehículo más rápido al otro.

Ejercicio 17

Un ciclista parte de una ciudad A a las HH horas, MM minutos y SS segundos. El tiempo de viaje hasta llegar a otra ciudad B es de T segundos. Escribir un algoritmo que determine la hora de llegada a la ciudad B.

Ejercicio 18

Pedir el nombre y los dos apellidos de una persona y mostrar las iniciales.

Ejercicio 19

Escribir un algoritmo para calcular la nota final de un estudiante, considerando que: por cada respuesta correcta 5 puntos, por una incorrecta -1 y por respuestas en blanco 0. Imprime el resultado obtenido por el estudiante.

Ejercicio 20

Diseñar un algoritmo que nos diga el dinero que tenemos (en euros y céntimos) después de pedirnos cuantas monedas tenemos (de 2€, 1€, 50 céntimos, 20 céntimos o 10 céntimos).

Ejercicios estructuras alternativas

Ejercicio 1

Algoritmo que pida dos números e indique si el primero es mayor que el segundo o no.

Ejercicio 2

Algoritmo que pida un número y diga si es positivo, negativo o 0.

Ejercicio 3

Escribe un programa que lea un número e indique si es par o impar.

Ejercicio 4

Crea un programa que pida al usuario dos números y muestre su división si el segundo no es cero, o un mensaje de aviso en caso contrario.

Ejercicio 5

Escribe un programa que pida un nombre de usuario y una contraseña y si se ha introducido "pepe" y "asdasd" se indica "Has entrado al sistema", sino se da un error.

Ejercicio 6

Programa que lea una cadena por teclado y compruebe si es una letra mayúscula.

Ejercicio 7

Realiza un algoritmo que calcule la potencia, para ello pide por teclado la base y el exponente. Pueden ocurrir tres cosas:

- El exponente sea positivo, sólo tienes que imprimir la potencia.
- El exponente sea 0, el resultado es 1.
- El exponente sea negativo, el resultado es 1/potencia con el exponente positivo.

Ejercicio 8

Algoritmo que pida dos números 'nota' y 'edad' y un carácter 'sexo' y muestre el mensaje 'ACEPTADA' si la nota es mayor o igual a cinco, la edad es mayor o igual a dieciocho y el sexo es 'F'. En caso de que se cumpla lo mismo, pero el sexo sea 'M', debe imprimir 'POSIBLE'. Si no se cumplen dichas condiciones se debe mostrar 'NO ACEPTADA'.

Ejercicio 9

Algoritmo que pida tres números y los muestre ordenados (de mayor a menor);

Ejercicio 10

Algoritmo que pida los puntos centrales x_1, y_1, x_2, y_2 y los radios r_1, r_2 de dos circunferencias y las clasifique en uno de estos estados:

- exteriores
- tangentes exteriores
- secantes
- tangentes interiores
- interiores
- concéntricas

Ejercicio 11

Programa que lea 3 datos de entrada A, B y C. Estos corresponden a las dimensiones de los lados de un triángulo. El programa debe determinar que tipo de triangulo es, teniendo en cuenta los siguiente:

- Si se cumple Pitágoras entonces es triángulo rectángulo
- Si sólo dos lados del triángulo son iguales entonces es isósceles.
- Si los 3 lados son iguales entonces es equilátero.
- Si no se cumple ninguna de las condiciones anteriores, es escaleno.

Ejercicio 12

Escribir un programa que lea un año indicar si es bisiesto. Nota: un año es bisiesto si es un número divisible por 4, pero no si es divisible por 100, excepto que también sea divisible por 400.

Ejercicio 13

Escribe un programa que pida una fecha (día, mes y año) y diga si es correcta.

Ejercicio 14

La asociación de vinicultores tiene como política fijar un precio inicial al kilo de uva, la cual se clasifica en tipos A y B, y además en tamaños 1 y 2. Cuando se realiza la venta del producto, ésta es de un solo tipo y tamaño, se requiere determinar cuánto recibirá un productor por la uva que entrega en un embarque, considerando lo siguiente: si es de tipo A, se le cargan 20 céntimos al precio inicial cuando es de tamaño 1; y 30 céntimos si es de tamaño 2. Si es de tipo B, se rebajan 30 céntimos cuando es de tamaño 1, y 50 céntimos cuando es de tamaño 2. Realice un algoritmo para determinar la ganancia obtenida.

Ejercicio 15

El director de una escuela está organizando un viaje de estudios, y requiere determinar cuánto debe cobrar a cada alumno y cuánto debe pagar a la compañía de viajes por el servicio. La forma de cobrar es la siguiente: si son 100 alumnos o más, el costo por cada alumno es de 65 euros; de 50 a 99 alumnos, el costo es de 70 euros, de 30 a 49, de 95 euros, y si son menos de 30, el costo de la renta del autobús es de 4000 euros, sin importar el número de alumnos.

Realice un algoritmo que permita determinar el pago a la compañía de autobuses y lo que debe pagar cada alumno por el viaje.

Ejercicio 16

La política de cobro de una compañía telefónica es: cuando se realiza una llamada, el cobro es por el tiempo que ésta dura, de tal forma que los primeros cinco minutos cuestan 1 euro, los siguientes tres, 80 céntimos, los siguientes dos minutos, 70 céntimos, y a partir del décimo minuto, 50 céntimos.

Además, se carga un impuesto de 3 % cuando es domingo, y si es otro día, en turno de mañana, 15 %, y en turno de tarde, 10 %. Realice un algoritmo para determinar cuánto debe pagar por cada concepto una persona que realiza una llamada.

Ejercicio 17

Realiza un programa que pida por teclado el resultado (dato entero) obtenido al lanzar un dado de seis caras y muestre por pantalla el número en letras (dato cadena) de la cara opuesta al resultado obtenido.

- Nota 1: En las caras opuestas de un dado de seis caras están los números: 1-6, 2-5 y 3-4.
- Nota 2: Si el número del dado introducido es menor que 1 o mayor que 6, se mostrará el mensaje: "ERROR: número incorrecto."

Ejemplo:

```
Introduzca número del dado: 5
En la cara opuesta está el "dos".
```

Ejercicio 18

Realiza un programa que pida el día de la semana (del 1 al 7) y escriba el día correspondiente. Si introducimos otro número nos da un error.

Ejercicio 19

Escribe un programa que pida un número entero entre uno y doce e imprima el número de días que tiene el mes correspondiente.

Ejercicio 20

Una compañía de transporte internacional tiene servicio en algunos países de América del Norte, América Central, América del Sur, Europa y Asia. El costo por el servicio de transporte

se basa en el peso del paquete y la zona a la que va dirigido. Lo anterior se muestra en la tabla:

Zona	Ubicación	Costo/gramo
1	América del Norte	24.00 euros
2	América Central	20.00 euros
3	América del Sur	21.00 euros
4	Europa	10.00 euros
5	Asia	18.00 euros

Parte de su política implica que los paquetes con un peso superior a 5 kg no son transportados, esto por cuestiones de logística y de seguridad.

Realice un algoritmo para determinar el cobro por la entrega de un paquete o, en su caso, el rechazo de la entrega.

Ejercicios estructuras repetitivas

Ejercicio 1

Crea una aplicación que pida un número y calcule su factorial (El factorial de un número es el producto de todos los enteros entre 1 y el propio número y se representa por el número seguido de un signo de exclamación. Por ejemplo $5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$),

Ejercicio 2

Crea una aplicación que permita adivinar un número. La aplicación genera un número aleatorio del 1 al 100. A continuación va pidiendo números y va respondiendo si el número a adivinar es mayor o menor que el introducido, a demás de los intentos que te quedan (tienes 10 intentos para acertarlo). El programa termina cuando se acierta el número (además te dice en cuantos intentos lo has acertado), si se llega al límite de intentos te muestra el número que había generado.

Ejercicio 3

Algoritmo que pida números hasta que se introduzca un cero. Debe imprimir la suma y la media de todos los números introducidos.

Ejercicio 4

Realizar un algoritmo que pida números (se pedirá por teclado la cantidad de números a introducir). El programa debe informar de cuantos números introducidos son mayores que 0, menores que 0 e iguales a 0.

Ejercicio 5

Algoritmo que pida caracteres e imprima 'VOCAL' si son vocales y 'NO VOCAL' en caso contrario, el programa termina cuando se introduce un espacio.

Ejercicio 6

Escribir un programa que imprima todos los números pares entre dos números que se le pidan al usuario.

Ejercicio 7

Realizar una algoritmo que muestre la tabla de multiplicar de un número introducido por teclado.

Ejercicio 8

Escribe un programa que pida el límite inferior y superior de un intervalo. Si el límite inferior es mayor que el superior lo tiene que volver a pedir.

A continuación se van introduciendo números hasta que introduzcamos el 0. Cuando termine el programa dará las siguientes informaciones:

- La suma de los números que están dentro del intervalo (intervalo abierto).
- Cuántos números están fuera del intervalo.
- He informa si hemos introducido algún número igual a los límites del intervalo.

Ejercicio 9

Escribe un programa que dados dos números, uno real (base) y un entero positivo (exponente), saque por pantalla el resultado de la potencia. No se puede utilizar el operador de potencia.

Ejercicio 10

Algoritmo que muestre la tabla de multiplicar de los números 1,2,3,4 y 5.

Ejercicio 11

Escribe un programa que diga si un número introducido por teclado es o no primo. Un número primo es aquel que sólo es divisible entre él mismo y la unidad. Nota: Es suficiente probar hasta la raíz cuadrada del número para ver si es divisible por algún otro número.

Ejercicio 12

Realizar un algoritmo para determinar cuánto ahorrará una persona en un año, si al final de cada mes deposita cantidades variables de dinero; además, se quiere saber cuánto lleva ahorrado cada mes.

Ejercicio 13

Una empresa tiene el registro de las horas que trabaja diariamente un empleado durante la semana (seis días) y requiere determinar el total de éstas, así como el sueldo que recibirá por las horas trabajadas.

Ejercicio 14

Una persona se encuentra en el kilómetro 70 de una carretera, otra se encuentra en el km 150, los coches tienen sentido opuesto y tienen la misma velocidad. Realizar un programa para determinar en qué kilómetro de esa carretera se encontrarán.

Ejercicio 15

Una persona adquirió un producto para pagar en 20 meses. El primer mes pagó 10 €, el segundo 20 €, el tercero 40 € y así sucesivamente. Realizar un algoritmo para determinar cuánto debe pagar mensualmente y el total de lo que pagó después de los 20 meses.

Ejercicio 16

Una empresa les paga a sus empleados con base en las horas trabajadas en la semana. Realice un algoritmo para determinar el sueldo semanal de N trabajadores y, además, calcule cuánto pagó la empresa por los N empleados.

Ejercicio 17

Una empresa les paga a sus empleados con base en las horas trabajadas en la semana. Para esto, se registran los días que trabajó y las horas de cada día. Realice un algoritmo para determinar el sueldo semanal de N trabajadores y además calcule cuánto pagó la empresa por los N empleados.

Ejercicio 18

Hacer un programa que muestre un cronometro, indicando las horas, minutos y segundos.

Ejercicio 19

Realizar un ejemplo de menú, donde podemos escoger las distintas opciones hasta que seleccionamos la opción de “Salir”.

Ejercicio 20

Mostrar en pantalla los N primero número primos. Se pide por teclado la cantidad de números primos que queremos mostrar.

Ejercicios cadena de caracteres

Ejercicio 1

Escribir por pantalla cada carácter de una cadena introducida por teclado.

Ejercicio 2

Realizar un programa que comprueba si una cadena leída por teclado comienza por una subcadena introducida por teclado.

Ejercicio 3

Pide una cadena y un carácter por teclado (valida que sea un carácter) y muestra cuantas veces aparece el carácter en la cadena.

Ejercicio 4

Suponiendo que hemos introducido una cadena por teclado que representa una frase (palabras separadas por espacios), realiza un programa que cuente cuantas palabras tiene.

Ejercicio 5

Si tenemos una cadena con un nombre y apellidos, realizar un programa que muestre las iniciales en mayúsculas.

Ejercicio 6

Realizar un programa que dada una cadena de caracteres por caracteres, genere otra cadena resultado de invertir la primera.

Ejercicio 7

Pide una cadena y dos caracteres por teclado (valida que sea un carácter), sustituye la aparición del primer carácter en la cadena por el segundo carácter.

Ejercicio 8

Realizar un programa que lea una cadena por teclado y convierta las mayúsculas a minúsculas y viceversa.

Ejercicio 9

Realizar un programa que compruebe si una cadena **contiene** una subcadena. Las dos cadenas se introducen por teclado.

Ejercicio 10

Introducir una cadena de caracteres e indicar si es un palíndromo. Una palabra palíndroma es aquella que se lee igual adelante que atrás.

Ejercicios de listas

Ejercicio 1

Realizar un programa que inicialice una lista con 10 valores aleatorios (del 1 al 10) y posteriormente muestre en pantalla cada elemento de la lista junto con su cuadrado y su cubo.

Ejercicio 2

Crea una lista e inicializala con 5 cadenas de caracteres leídas por teclado. Copia los elementos de la lista en otra lista pero en orden inverso, y muestra sus elementos por la pantalla.

Ejercicio 3

Se quiere realizar un programa que lea por teclado las 5 notas obtenidas por un alumno (comprendidas entre 0 y 10). A continuación debe mostrar todas las notas, la nota media, la nota más alta que ha sacado y la menor.

Ejercicio 4

Programa que declare una lista y la vaya llenando de números hasta que introduzcamos un número negativo. Entonces se debe imprimir el vector (sólo los elementos introducidos).

Ejercicio 5

Hacer un programa que inicialice una lista de números con valores aleatorios (10 valores), y posterior ordene los elementos de menor a mayor.

Ejercicio 6

Crea un programa que pida un número al usuario un número de mes (por ejemplo, el 4) y diga cuántos días tiene (por ejemplo, 30) y el nombre del mes. Debes usar listas. Para simplificarlo vamos a suponer que febrero tiene 28 días.

Ejercicio 7

Programa que declare tres listas 'lista1', 'lista2' y 'lista3' de cinco enteros cada uno, pida valores para 'lista1' y 'lista2' y calcule lista3=lista1+lista2.

Ejercicio 8

Queremos guardar los nombres y la edades de los alumnos de un curso. Realiza un programa que introduzca el nombre y la edad de cada alumno. El proceso de lectura de datos terminará cuando se introduzca como nombre un asterisco (*) Al finalizar se mostrará los siguientes datos:

- Todos los alumnos mayores de edad.
- Los alumnos mayores (los que tienen más edad)

Ejercicio 9

Queremos guardar la temperatura mínima y máxima de 5 días. Realiza un programa que de la siguiente información:

- La temperatura media de cada día
- Los días con menos temperatura
- Se lee una temperatura por teclado y se muestran los días cuya temperatura máxima coincide con ella. Si no existe ningún día se muestra un mensaje de información.

Ejercicio 10

Diseñar el algoritmo correspondiente a un programa, que:

- Crea una tabla (lista con dos dimensiones) de 5x5 enteros.
- Carga la tabla con valores numéricos enteros.
- Suma todos los elementos de cada fila y todos los elementos de cada columna visualizando los resultados en pantalla.

Ejercicio 11

Diseñar el algoritmo correspondiente a un programa, que:

- Crea una tabla bidimensional de longitud 5x5 y nombre 'diagonal'.
- Carga la tabla de forma que los componentes pertenecientes a la diagonal de la matriz tomen el valor 1 y el resto el valor 0.
- Muestra el contenido de la tabla en pantalla.

Ejercicio 12

Diseñar el algoritmo correspondiente a un programa, que:

- Crea una tabla bidimensional de longitud 5x15 y nombre 'marco'.
- Carga la tabla con dos únicos valores 0 y 1, donde el valor uno ocupará las posiciones o elementos que delimitan la tabla, es decir, las más externas, mientras que el resto de los elementos contendrán el valor 0.

```
1111111111111111
1000000000000001
1000000000000001
1000000000000001
1111111111111111
```

- Visualiza el contenido de la matriz en pantalla.

Ejercicio 13

De una empresa de transporte se quiere guardar el nombre de los conductores que tiene, y los kilómetros que conducen cada día de la semana.

Para guardar esta información se van a utilizar dos arreglos:

- Nombre: Lista para guardar los nombres de los conductores.
- kms: Tabla para guardar los kilómetros que realizan cada día de la semana.

Se quiere generar una nueva lista ("total_kms") con los kilómetros totales que realiza cada conductor.

Al finalizar se muestra la lista con los nombres de conductores y los kilómetros que ha realizado.

Ejercicio 14

Crear un programa que lea los precios de 5 artículos y las cantidades vendidas por una empresa en sus 4 sucursales. Informar:

- Las cantidades totales de cada artículo.
- La cantidad de artículos en la sucursal 2.
- La cantidad del artículo 3 en la sucursal 1.
- La recaudación total de cada sucursal.
- La recaudación total de la empresa.
- La sucursal de mayor recaudación.

Ejercicio 15

Crear un programa de ordenador para gestionar los resultados de la quiniela de fútbol. Para ello vamos a utilizar dos tablas:

- Equipos: Que es una tabla de cadenas donde guardamos en cada columna el nombre de los equipos de cada partido. En la quiniela se indican 15 partidos.
- Resultados: Es una tabla de enteros donde se indica el resultado. También tiene dos columnas, en la primera se guarda el número de goles del equipo que está guardado en la primera columna de la tabla anterior, y en la segunda los goles del otro equipo.

El programa ira pidiendo los nombres de los equipos de cada partido y el resultado del partido, a continuación se imprimirá la quiniela de esa jornada.

¿Qué modificación habría que hacer en las tablas para guardar todos los resultados de todas las jornadas de la temporada?

Ejercicio 16

Vamos a crear un programa que tenga el siguiente menú:

- 1.Añadir número a la lista: Me pide un número de la lista y lo añade al final de la lista.

2. Añadir número de la lista en una posición: Me pide un número y una posición, y si la posición existe en la lista lo añade a ella (la posición se pide a partir de 1).
3. Longitud de la lista: te muestra el número de elementos de la lista.
4. Eliminar el último número: Muestra el último número de la lista y lo borra.
5. Eliminar un número: Pide una posición, y si la posición existe en la lista lo borra de ella (la posición se pide a partir de 1).
6. Contar números: Te pide un número y te dice cuantas apariciones hay en la lista.
7. Posiciones de un número: Te pide un número y te dice en que posiciones está (contando desde 1).
8. Mostrar números: Muestra los números de la lista
9. Salir

Ejercicio 17

Crear un programa que añada números a una lista hasta que introducimos un número negativo. A continuación debe crear una nueva lista igual que la anterior pero eliminando los números duplicados. Muestra esta segunda lista para comprobar que hemos eliminados los duplicados.

Ejercicio 18

Escriba un programa que permita crear una lista de palabras y que, a continuación de tres opciones:

- Contar: Me pide una cadena, y me dice cuantas veces aparece en la lista
- Modificar: Me pide una cadena, y otra cadena a modificar, y modifica todas las apariciones de la primera por la segunda en la lista.
- Eliminar: Me pide una cadena, y la elimina de la lista.
- Mostrar: Muestra la lista de cadenas
- Terminar

Ejercicios de diccionarios

Ejercicio 1

Escribe un programa python que pida un número por teclado y que cree un diccionario cuyas claves sean desde el número 1 hasta el número indicado, y los valores sean los cuadrados de las claves.

Ejercicio 2

Escribe un programa que lea una cadena y devuelva un diccionario con la cantidad de apariciones de cada carácter en la cadena.

Ejercicio 3

Vamos a crear un programa en python donde vamos a declarar un diccionario para guardar los precios de las distintas frutas. El programa pedirá el nombre de la fruta y la cantidad que se ha vendido y nos mostrará el precio final de la fruta a partir de los datos guardados en el diccionario. Si la fruta no existe nos dará un error. Tras cada consulta el programa nos preguntará si queremos hacer otra consulta.

Ejercicio 4

Codifica un programa en python que nos permita guardar los nombres de los alumnos de una clase y las notas que han obtenido. Cada alumno puede tener distinta cantidad de notas. Guarda la información en un diccionario cuya claves serán los nombres de los alumnos y los valores serán listas con las notas de cada alumno.

El programa pedirá el número de alumnos que vamos a introducir, pedirá su nombre e irá pidiendo sus notas hasta que introduzcamos un número negativo. Al final el programa nos mostrará la lista de alumnos y la nota media obtenida por cada uno de ellos. Nota: si se introduce el nombre de un alumno que ya existe el programa nos dará un error.

Ejercicio 5

Escribir un programa que implemente una agenda. En la agenda se podrán guardar nombres y números de teléfono. El programa nos dará el siguiente menú:

- Añadir/modificar: Nos pide un nombre. Si el nombre se encuentra en la agenda, debe mostrar el teléfono y, opcionalmente, permitir modificarlo si no es correcto. Si el nombre no se encuentra, debe permitir ingresar el teléfono correspondiente.

- Buscar: Nos pide una cadena de caracteres, y nos muestras todos los contactos cuyos nombres comiencen por dicha cadena.
- Borrar: Nos pide un nombre y si existe nos preguntará si queremos borrarlo de la agenda.
- Listar: Nos muestra todos los contactos de la agenda.

Implementar el programa con un diccionario.

Ejercicios de funciones

Ejercicio 1

Crea una función **EscribirCentrado**, que reciba como parámetro un texto y lo escriba centrado en pantalla (suponiendo una anchura de 80 columnas; pista: deberás escribir $40 - \text{longitud}/2$ espacios antes del texto). Además subraya el mensaje utilizando el carácter **=**.

Ejercicio 2

Crea un programa que pida dos números enteros al usuario y diga si alguno de ellos es múltiplo del otro. Crea una función **EsMultiplo** que reciba los dos números, y devuelve si el primero es múltiplo del segundo.

Ejercicio 3

Crear una función que calcule la temperatura media de un día a partir de la temperatura máxima y mínima. Crear un programa principal, que utilizando la función anterior, vaya pidiendo la temperatura máxima y mínima de cada día y vaya mostrando la media. El programa pedirá el número de días que se van a introducir.

Ejercicio 4

Crea una función **“ConvertirEspaciado”**, que reciba como parámetro un texto y devuelve una cadena con un espacio adicional tras cada letra. Por ejemplo, **“Hola, tú”** devolverá **“H o l a , t ú ”**. Crea un programa principal donde se use dicha función.

Ejercicio 5

Crea una función **“calcularMaxMin”** que recibe una lista con valores numéricos y devuelve el valor máximo y el mínimo. Crea un programa que pida números por teclado y muestre el máximo y el mínimo, utilizando la función anterior.

Ejercicio 6

Diseñar una función que calcule el área y el perímetro de una circunferencia. Utiliza dicha función en un programa principal que lea el radio de una circunferencia y muestre su área y perímetro.

Ejercicio 7

Crear una subrutina llamada "Login", que recibe un nombre de usuario y una contraseña y te devuelve Verdadero si el nombre de usuario es "usuario1" y la contraseña es "asdasd".

Además recibe el número de intentos que se ha intentado hacer login y si no se ha podido hacer login incrementa este valor.

Crear un programa principal donde se pida un nombre de usuario y una contraseña y se intente hacer login, solamente tenemos tres oportunidades para intentarlo.

Ejercicio 8

Crear una función recursiva que permita calcular el factorial de un número. Realiza un programa principal donde se lea un entero y se muestre el resultado del factorial.

Ejercicio 9

Crear una función que calcule el MCD de dos números por el método de Euclides. El método de Euclides es el siguiente:

- Se divide el número mayor entre el menor.
- Si la división es exacta, el divisor es el MCD.
- Si la división no es exacta, dividimos el divisor entre el resto obtenido y se continúa de esta forma hasta obtener una división exacta, siendo el último divisor el MCD.

Crear un programa principal que lea dos números enteros y muestre el MCD.

Ejercicio 10

Escribir dos funciones que permitan calcular:

- La cantidad de segundos en un tiempo dado en horas, minutos y segundos.
- La cantidad de horas, minutos y segundos de un tiempo dado en segundos.

Escribe un programa principal con un menú donde se pueda elegir la opción de convertir a segundos, convertir a horas, minutos y segundos o salir del programa.

Ejercicio 11

El día juliano correspondiente a una fecha es un número entero que indica los días que han transcurrido desde el 1 de enero del año indicado. Queremos crear un programa principal que al introducir una fecha nos diga el día juliano que corresponde. Para ello podemos hacer las siguientes subrutinas:

- **LeerFecha**: Nos permite leer por teclado una fecha (día, mes y año).
- **DiasDelMes**: Recibe un mes y un año y nos dice los días de ese mes en ese año.
- **EsBisiesto**: Recibe un año y nos dice si es bisiesto.
- **Calcular_Dia_Juliano**: recibe una fecha y nos devuelve el día juliano.

Ejercicio 12

Vamos a mejorar el ejercicio anterior haciendo una función para validar la fecha. De tal forma que al leer una fecha se asegura que es válida.

Ejercicio 13

Queremos crear un programa que trabaje con fracciones a/b. Para representar una fracción vamos a utilizar dos enteros: **numerador** y **denominador**.

Vamos a crear las siguientes funciones para trabajar con funciones:

- **Leer_fracción**: La tarea de esta función es leer por teclado el numerador y el denominador. Cuando leas una fracción debes simplificarla.
- **Escribir_fracción**: Esta función escribe en pantalla la fracción. Si el dominador es 1, se muestra sólo el numerador.
- **Calcular_mcd**: Esta función recibe dos número y devuelve el máximo común divisor.
- **Simplificar_fracción**: Esta función simplifica la fracción, para ello hay que dividir numerador y dominador por el MCD del numerador y denominador.
- **Sumar_fracciones**: Función que recibe dos funciones $n1/d1$ y $n2/d2$, y calcula la suma de las dos fracciones. La suma de dos fracciones es otra fracción cuyo **numerador**= $n1*d2+d1*n2$ y **denominador**= $d1*d2$. Se debe simplificar la fracción resultado.

- Restar_fracciones**: Función que resta dos fracciones: $\text{numerador} = n1 * d2 - d1 * n2$ y $\text{denominador} = d1 * d2$. Se debe simplificar la fracción resultado.
- Multiplicar_fracciones**: Función que recibe dos fracciones y calcula el producto, para ello $\text{numerador} = n1 * n2$ y $\text{denominador} = d1 * d2$. Se debe simplificar la fracción resultado.
- Dividir_fracciones**: Función que recibe dos fracciones y calcula el cociente, para ello $\text{numerador} = n1 * d2$ y $\text{denominador} = d1 * n2$. Se debe simplificar la fracción resultado.

Crear un programa que utilizando las funciones anteriores muestre el siguiente menú:

1. Sumar dos fracciones: En esta opción se piden dos fracciones y se muestra el resultado.
2. Restar dos fracciones: En esta opción se piden dos fracciones y se muestra la resta.
3. Multiplicar dos fracciones: En esta opción se piden dos fracciones y se muestra la producto.
4. Dividir dos fracciones: En esta opción se piden dos fracciones y se muestra la cociente.
5. Salir

Ejercicio 14

Vamos a crear un programa para trabajar con una pila. Una pila es una estructura de datos que nos permite guardar un conjunto de variables. La característica fundamental es que el último elemento que se añade al conjunto es el primero que se puede sacar.

Para representar una pila vamos a utilizar una lista de cadenas de caracteres.

Vamos a crear varias funciones para trabajar con la pila:

- LongitudPila**: Función que recibe una pila y devuelve el número de elementos que tiene.
- EstaVacíaPila**: Función que recibe una pila y que devuelve si la pila está vacía, no tiene elementos.
- EstaLlenaPila**: Función que recibe una pila y que devuelve si la pila está llena.

- **AddPila**: función que recibe una cadena de caracteres y una pila, y añade la cadena a la pila, si no está llena. Si esta llena muestra un mensaje de error.
- **SacarDeLaPila**: Función que recibe una pila y devuelve el último elemento añadido y lo borra de la pila. Si la pila está vacía muestra un mensaje de error.
- **EscribirPila**: Función que recibe una pila y muestra en pantalla los elementos de la pila.

Realiza un programa principal que nos permita usar las funciones anterior, que nos muestre un menú, con las siguientes opciones:

1. Añadir elemento a la pila
2. Sacar elemento de la pila
3. Longitud de la pila
4. Mostrar pila
5. Salir

Ejercicio 15

Vamos a realizar un programa similar al anterior para trabajar con una cola. Una cola es una estructura de datos que nos permite guardar un conjunto de variables. La característica fundamental es que el primer elemento que se añade al conjunto es el primero que se puede sacar.

En realizada nos sirven todas las funciones del ejercicio anterior menos la función **SacarDeLaCola** que es la que tienes que modificar.

Ejercicios de Programación orientada a objetos

Ejercicio 1

Vamos a crear una clase llamada **Persona**. Sus atributos son: **nombre**, **edad** y **DNI**.

Construye los siguientes métodos para la clase:

- Un constructor, donde los datos pueden estar vacíos.
- Los setters y getters para cada uno de los atributos. Hay que validar las entradas de datos.
- **mostrar()**: Muestra los datos de la persona.

- `esMayorDeEdad()`: Devuelve un valor lógico indicando si es mayor de edad.

Ejercicio 2

Crea una clase llamada `Cuenta` que tendrá los siguientes atributos: titular (que es una persona) y cantidad (puede tener decimales). El titular será obligatorio y la cantidad es opcional. Construye los siguientes métodos para la clase:

- Un constructor, donde los datos pueden estar vacíos.
- Los setters y getters para cada uno de los atributos. El atributo no se puede modificar directamente, sólo ingresando o retirando dinero.
- `mostrar()`: Muestra los datos de la cuenta.
- `ingresar(cantidad)`: se ingresa una cantidad a la cuenta, si la cantidad introducida es negativa, no se hará nada.
- `retirar(cantidad)`: se retira una cantidad a la cuenta. La cuenta puede estar en números rojos.

Ejercicio 3

Vamos a definir ahora una “Cuenta Joven”, para ello vamos a crear una nueva clase `CuentaJoven` que deriva de la anterior. Cuando se crea esta nueva clase, además del titular y la cantidad se debe guardar una bonificación que estará expresada en tanto por ciento. Construye los siguientes métodos para la clase:

- Un constructor.
- Los setters y getters para el nuevo atributo.
- En esta ocasión los titulares de este tipo de cuenta tienen que ser mayor de edad., por lo tanto hay que crear un método `esTitularValido()` que devuelve verdadero si el titular es mayor de edad pero menor de 25 años y falso en caso contrario.
- Además la retirada de dinero sólo se podrá hacer si el titular es válido.
- El método `mostrar()` debe devolver el mensaje de “Cuenta Joven” y la bonificación de la cuenta.

Piensa los métodos heredados de la clase madre que hay que reescribir.