

EJERCICIOS INTRODUCCIÓN A LA PROGRAMACIÓN

Estructuras Secuenciales

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

Ejercicio 4

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 5

Calcular la media de tres números pedidos por teclado.

Ejercicio 6

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 7

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

Ejercicio 8

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 9

Pide al usuario dos pares de números (x1,y2) y (x2,y2), que representen dos puntos en el plano. Calcula y muestra la distancia entre ellos.

Ejercicio 10

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

Ejercicio 11

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

Ejercicio 12

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 13

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.

Estructuras alternativas

Ejercicio 1

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

Ejercicio 2

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

Ejercicio 3

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 4

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 $1/\text{potencia}$ con el exponente positivo.

Ejercicio 5

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 6

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 7

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 8

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 9

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

Ejercicio 10

Escribe un programa que resuelva una ecuación de segundo grado del tipo $AX^2 + BX + C$

Estructuras repetitivas

Ejercicio 1

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, además 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 2

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 3

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 4

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

Ejercicio 5

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 6

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 7

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 8

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 9

Realizar un ejemplo de menú, donde podemos escoger las distintas opciones (Opción A, Opción B, Salir) hasta que seleccionamos la opción de "Salir".

Ejercicio 11

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

Ejercicio 12

Realizar un programa que pida el número de filas a mostrar y comience por la mayor fila mostrando tantos asteriscos como la fila actual hasta llegar a cero.

[Ejercicios de caracteres](#)

Ejercicio 1

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

Ejercicio 2

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 3

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 4

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

Ejercicio 5

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

Ejercicio 6

Escribe un método que permita cifrar/descifrar una cadena usando el cifrado Cesar, es decir, a cada carácter de la cadena se le suma una cifra en el abecedario. Se pide por teclado la palabra a cifrar/descifrar, la clave (cifra) y lo que se desea hacer, cifrar o descifrar.

Ejercicio 7

Escribe un programa que dada una cadena de texto (hola45mundo44) muestre el resultado de sumar todas las cifras numéricas independientes que aparezcan en la cadena.

Ejercicio 8

Escribe un programa que muestre la frecuencia de las vocales en un texto

Arrays

Ejercicio 1

Crear un vector de 5 elementos de cadenas de caracteres, inicializa el vector con datos leídos por el teclado. Copia los elementos del vector en otro vector, pero en orden inverso, y muéstralo por la pantalla.

Ejercicio 2

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 un vector. Para simplificarlo vamos a suponer que febrero tiene 28 días.

Ejercicio 3

Programa que declare tres vectores 'vector1', 'vector2' y 'vector3' de cinco enteros cada uno, pida valores para 'vector1' y 'vector2' y calcule $\text{vector3} = \text{vector1} + \text{vector2}$.

Ejercicio 4

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 5

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 6

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: Vector para guardar los nombres de los conductores.

kms: Tabla para guardar los kilómetros que realizan cada día de la semana.

Se quiere generar un nuevo vector ("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.

Funciones y procedimientos

Ejercicio 1

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

Ejercicio 2

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 3

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 4

Crea una función que calcule la sucesión de Fibonacci pidiendo como parámetro cuántos números de la sucesión quieres mostrar

Ejercicio 5

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 denominador es 1, se muestra sólo el numerador.

Calcular_mcd: Esta función recibe dos números y devuelve el máximo común divisor.

Simplificar_fracción: Esta función simplifica la fracción, para ello hay que dividir numerador y denominador por el MCD del numerador y denominador.

Sumar_fracciones: Función que recibe dos fracciones $n1/d1$ 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: numerador= $n1*d2-d1*n2$ y denominador= $d1*d2$. Se debe simplificar la fracción resultado.

Multiplicar_fracciones: Función que recibe dos fracciones y calcula el producto, para ello numerador= $n1*n2$ y denominador= $d1*d2$. Se debe simplificar la fracción resultado.

Dividir_fracciones: Función que recibe dos fracciones y calcula el cociente, para ello numerador= $n1*d2$ y denominador= $d1*n2$. Se debe simplificar la fracción resultado.

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

Sumar dos fracciones: En esta opción se piden dos fracciones y se muestra el resultado.

Restar dos fracciones: En esta opción se piden dos fracciones y se muestra la resta.

Multiplicar dos fracciones: En esta opción se piden dos fracciones y se muestra el producto.

Dividir dos fracciones: En esta opción se piden dos fracciones y se muestra el cociente.

Salir

Ejercicio 6

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 un arreglo (vector) de cadena de caracteres con tamaño 10, por lo tanto, la pila no podrá tener más de 10 elementos.

Vamos a crear varias funciones para trabajar con la pila:

InicializarPila: Como tenemos un arreglo de 10 elementos de cadenas tenemos que inicializarlo e introducir un carácter (por ejemplo, un * que indique que ese elemento del arreglo no corresponde con un dato de la pila. Esta función inicializa el vector con ese carácter.

LongitudPila: Función que recibe una pila y devuelve el número de elementos que tiene.

EstaVaciaPila: 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 está 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 anteriores, que nos muestre un menú, con las siguientes opciones:

Añadir elemento a la pila

Sacar elemento de la pila

Longitud de la pila

Mostrar pila

Salir