

Practica. Programación:

BUCLES

Caso de estudio 'For, Do While, While'.

Práctica 1.

Contenido

BLOQUE DE EJERCICIOS..... 3

1. EJERCICIOS (FOR) 3

2. EJERCICIOS DO-WHILE 4

3. EJERCICIOS WHILE..... 5

4. PROYECTO FINAL 6

BLOQUE DE EJERCICIOS

1. EJERCICIOS (FOR)

- 1.1. Escribe un algoritmo que imprima por pantalla los números del 0 al 10.
- 1.2. Escribe un algoritmo que imprima por pantalla los números del 1 al 100
- 1.3. Escribe un algoritmo que imprima por pantalla los números del 75 al 0 (en orden decreciente).
- 1.4. Escribe un algoritmo que imprima por pantalla los números pares del 0 al 1.000 (no está permitido el uso de condicionales).
- 1.5. Escribe un algoritmo que realice la suma de los 100 primeros números (Ejemplo 0, 1 => (0+1), 2 => (1+2), 3 => (1+2+3) ...)
- 1.6. Escribe un algoritmo que muestre por pantalla el número de impares entre 0 y 540.
- 1.7. Escribe un algoritmo que solicite un número, e imprima por pantalla la tabla de multiplicar de ese número.
- 1.8. Escribe un algoritmo que solicite un número, calcule la factorial (n!) y muestre el resultado al final del bucle. (La factorial de un número es ese número multiplicado por sus anteriores => [5! = 5*4*3*2*1]).
- 1.9. Escribe un algoritmo que imprime un conjunto de números del [100,-100] con un decremento de 20.
- 1.10. Diseña un algoritmo que determine si un número N es primo (un número primo sólo puede ser divisible por él mismo y por 1).

1. EJERCICIOS DO-WHILE

- 1.1. Escribe un algoritmo que lea palabras de forma indefinida y finalizará cuando se introduzca la palabra “fin”, mostrando al finalizar la cantidad de palabras leídas.
- 1.2. Escribe un algoritmo que lea números enteros indefinidamente hasta que se introduzca el número 0.
- 1.3. Escribe un algoritmo que lea números enteros indefinidamente hasta que se introduzca el número 0, cuando finalice el bucle tiene que imprimir la suma de los números anteriormente leídos.
- 1.4. Escribe un algoritmo que lea números enteros indefinidamente hasta que se introduzca el número 0, cuando finalice el bucle tiene que imprimir el promedio de los números anteriormente leídos.
- 1.5. Escribe un algoritmo que lea números enteros indefinidamente hasta que se introduzca el número 0, cuando finalice el bucle tiene que imprimir el número mayor y menor de todos los números leídos.
- 1.6. Escribe un algoritmo que elija un número entre 1 y 100, El usuario debe adivinar el número, y el programa debe dar pistas (más alto o más bajo), hasta que el usuario lo acierte. (Al final debe indicar el número de intentos que ha necesitado).

2. EJERCICIOS WHILE.

- 2.1. Escribe un algoritmo que almacene una cadena de caracteres en una variable, y pregunte al usuario indefinidamente hasta que se introduzca la contraseña correcta, indicando "Login exitoso".
- 2.2. Escribe un programa que muestre el eco de todo lo que el usuario introduzca hasta que el usuario escriba "salir" que terminará.
- 2.3. Escriba un algoritmo que pida dos números enteros. El programa pedirá de nuevo el segundo número mientras no sea mayor que el primero. El programa terminará escribiendo los dos números.
- 2.4. Escriba un algoritmo que pida números enteros mientras sean cada vez más grandes.
- 2.5. Escriba un algoritmo que pida un valor límite positivo y a continuación pida números hasta que la suma de los números introducidos supere el límite inicial.
- 2.6. Escriba un algoritmo que pida primero dos números enteros (mínimo y máximo) y que después pida números enteros situados entre ellos. El programa terminará cuando se escriba un número que no esté comprendido entre los dos valores iniciales. El programa termina escribiendo la cantidad de números escritos.
- 2.7. Escriba un programa que pida números pares mientras el usuario indique que quiere seguir introduciendo números. Para indicar que quiere seguir escribiendo números, el usuario deberá contestar S o s a la pregunta.

3. PROYECTO FINAL

3.1. Escribe un algoritmo que simule el juego de Piedra, Papel, Tijera.

- Debe incluir: Sistema de turnos, contador de victorias por jugador, cuando el contador de uno de los 2 jugadores sea igual a 3, no debe reproducir más partidas y debe poner quien ha ganado y quien ha perdido.
- Opcionalmente puede incluir el nombre del jugador A, el B se llamará CPU, y cuando ponga el ganador mostrar “Ha ganado {nombre jugador} / CPU”.