Alejandro Martínez García
Profesor Particular

Practica. Programación:

BUCLES

Caso de estudio ‘For, Do While, While’.

Práctica 1.

Alejandro Martínez García
Profesor Particular

Contenido

Alejandro Martínez García
Profesor Particular

BLOQUE DE EJERCICIOS

1. EJERCICIOS (FOR)

Comentado [A1]: Se realiza una actualización donde, se reordena en orden de dificultad, y se agrega subsecciones en 3 de los ejercicios a modo de aclarar y reconocer las diferentes formas de realizar el ejercicio.

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

1.4.1. Está permitido el uso de condicionales.

1.4.2. No está permitido el uso de condicionales.

1.5. Escribe un algoritmo que muestre por pantalla el número de impares entre 0 y 540.

1.5.1. Está permitido el uso de condicionales.

1.5.2. No está permitido el uso de condicionales.

1.6. Escribe un algoritmo que imprime un conjunto de números del [100,-100] con un decremento de 20.

1.7. Escribe un algoritmo que solicite un número, e imprima por pantalla la tabla de multiplicar de ese número.

1.7.1. La tabla de multiplicar se muestra del 0 al 10.

1.7.2. La tabla de multiplicar se muestra del 10 al 0.

1.7.3. La tabla de multiplicar se muestra el 0 a N, siendo N un número solicitado por el programa hacia el usuario.

1.8. Escribe un algoritmo que solicite un número, calcule la factorial (n!) y muestre el resultado al final del bucle. (El factorial de un número es ese número multiplicado por sus anteriores => [5! = 5*4*3*2*1]).

1.9. Diseña un algoritmo que determine si un número N (siendo N un número solicitado por el programa al usuario) es primo (un número primo sólo puede ser divisible por él mismo y por 1).

Alejandro Martínez García
Profesor Particular

2. BUCLES FOR (ANIDADOS)

2.1. Escribe un algoritmo que, solicitando un número, imprima todas las tablas de multiplicar desde el 1 hasta el número indicado *[Si se ingresa 5, se mostrarán las tablas de multiplicar desde el 1 hasta el 5].*

2.2. Imprime un triángulo rectángulo con el carácter y altura solicitados *[Si se ingresa carácter: '*' y altura 5, se mostrarán lo siguiente].*

```
*
**
***
****
*****
```

2.3. Imprime todos los números primos que hay en un rango solicitado al usuario *[Si se ingresa el número 15 tendrá que imprimir todos los números primos del 1 al 15.]*.

2.4. Reloj digital (Simulación de horas, minutos y segundos).

```
00:00:00
00:00:01
.
.
.
23:59:59
```

Alejandro Martínez García
Profesor Particular

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).

Alejandro Martínez García
Profesor Particular

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 N 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”.