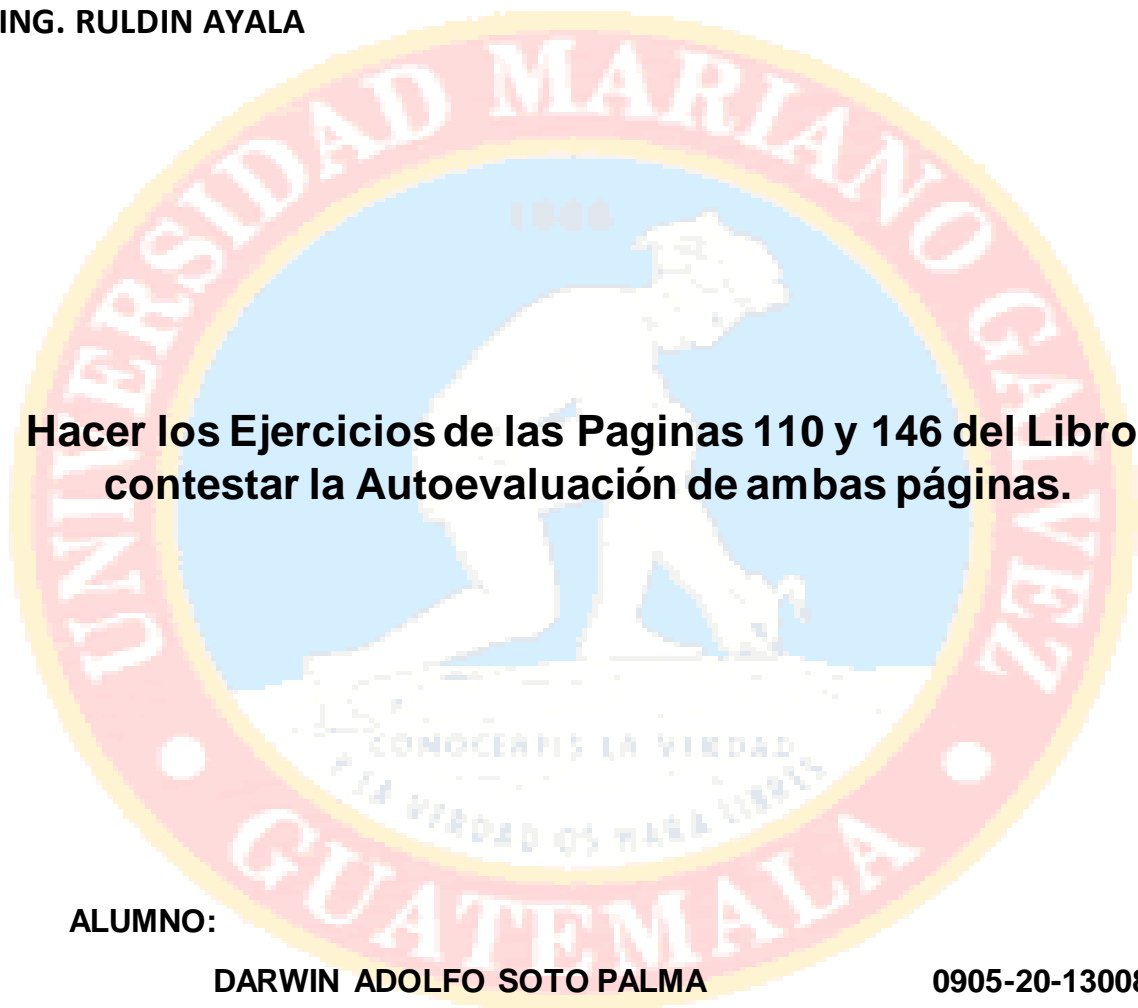




UNIVERSIDAD MARIANO GÁLVEZ DE GUATEMALA, CAMPUS JUTIAPA
FACULTAD DE INGENIERIA EN SISTEMAS Y CIENCIAS DE LA COMPUTACIÓN
CURSO: PROGRAMACIÓN 1
TERCER SEMESTRE 2021, SECCIÓN "B"
ING. RULDIN AYALA



**Hacer los Ejercicios de las Paginas 110 y 146 del Libro,
contestar la Autoevaluación de ambas páginas.**

ALUMNO:

DARWIN ADOLFO SOTO PALMA

0905-20-13008

JALPATAGUA, JUTIAPA, FEBRERO DEL 2,021

Test De Autoevaluación

1. ¿Qué son las estructuras selectivas?

Son Utilizadas para tomar decisiones lógicas por lo que también se les llama estructuras de decisión o alternativas, las estructuras de selección siempre evalúan una condición si esta llegara a ser positiva ejecuta una o varias instrucciones.

2. ¿Cuándo se ejecuta la sentencia de if?

Permite condicionar la ejecución de uno o varios bloques de sentencias al cumplimiento de una o varias condiciones. La ejecución de esta es la siguiente.

- Sí el resultado es True se ejecutará el bloque de sentencias.
- Si el resultado es False no se ejecutará el bloque de sentencias.

3. ¿Qué colocamos entre paréntesis en if?

La condición se encierra entre paréntesis (), se separa por un espacio en blanco de la palabra "if" y sin finalizar '; '. El bloque de instrucciones que se ejecuta solo si se cumple la condición, lo incluimos entre dos llaves de apertura y cierre '{' '}' esto para indicar el inicio y final.

4. ¿Qué es una expresión relacional?

Son quienes aceptan diversos tipos de argumentos y resultado que es la respuesta a la pregunta, es siempre del tipo cierto o falso es decir un resultado booleano.

5. ¿Cuáles son los operadores de una expresión relacional?

Son símbolos que se usan para comparar dos valores, si el resultado de la operación es correcto la expresión considera es verdadera en dado contrario es falsa

6. ¿A qué valores posibles puede evaluar una expresión relacional o lógica?

Números enteros o Racionales

7. ¿Qué es una expresión lógica?

La Expresión lógica o Booleana es aquella cuyo valor es siempre verdadero o falso y que las variables lógicas pueden tomar solo estos dos valores.

8. ¿Cuáles son los operadores de las expresiones lógicas?

- AND: Verdadero sólo si los dos elementos son verdaderos.
- OP: Verdadero si cualquiera de los elementos es verdadero.
- NOT: Cambia el valor de falso a verdadero y viceversa.



9. ¿Cómo funciona el switch?

Analiza múltiples opciones y pasa el control al bloque de instrucciones que sigue a la sentencia "case" del valor correcto, al contrario que en otros lenguajes como en C o Pascal la orden "switch" de C# permite comparar cadenas de texto.

10. ¿Qué es la variable de comparación y cómo se coloca?

- El Operador de Igualdad (==) compara referencias "Es decir si dos variables apuntan a ser el mismo objeto".
- El Operador Equals Compara valores (es decir si dos objetos son iguales).

11. ¿Cómo definimos los casos y cómo usamos break?

En C# es recomendable utilizar "Break" al final de cada caso para evitar errores por si se nos ha olvidado colocar algún Break, con la única excepción de que un caso no haga absolutamente nada excepto dejar pasar el control al siguiente caso.



1. Hacer un programa que le pida al usuario un número y la computadora responda si es par o impar.

```
using System;

namespace Programa_que_detecte_si_es_Par_o_Impar
{
    class Program
    {
        static void Main(string[] args)
        {
            int num;

            Console.WriteLine("\t\t Te Gustaria saber si es un Numero Par o Impar");

            Console.WriteLine("\n\tIngrese un numero: ");
            num = Int32.Parse(Console.ReadLine());

            if (num % 2 == 0)
            {
                Console.WriteLine("\n\n\tEl numero " + num + ".Genial!!! Es Numero
PAR");
            }
            else
            {
                Console.WriteLine("\n\n\tEl numero " + num + ". Lo Siento!!! Es numero
Impar");
            }

            Console.WriteLine("\n\n\tPresione < ENTER > para continuar . . .");
            Console.ReadKey();
        }
    }
}
```



2. Hacer un programa que transforme de grados a radianes o de radianes a grados dependiendo de lo que necesite el usuario.

```
using System;

namespace Transformar_Grados_a_Radianes_o_Radianes_a_Grados
{
    class Program
    {
        static void Main(string[] args)
        {
            double grados, radianes, pi = 3.1416;
            string valor = "";

            Console.Write("Escriba la Cantidad del Grado a Convertir:");
            valor = Console.ReadLine();
            grados = Convert.ToDouble(valor);

            radianes = gradosRadianes(grados, pi);

            Console.WriteLine("{0}° Equivale a {1} Radianes", grados, radianes);
        }

        static double gradosRadianes(double n1, double n2)
        {
            double r = 0.0;

            r = (n1 * n2) / 180;
            return r;
        }
    }
}
```




```
        case 7;
            Console.WriteLine();
            Console.WriteLine("El numero que ingreso corresponde al dia
Domingo");
            break;
        default:
            Console.WriteLine();
            Console.WriteLine("El numero que ingreso esta fuera de rango");
            break;
    }
    Console.ReadKey();
}
}
```

4. Hacer un programa que pueda calcular el perímetro y el área de cualquier polígono regular, pero que le pregunte al usuario qué desea calcular.

```
using System;

namespace Area_de_Poligono
{
    class Program
    {
        static void Main(string[] args)
        {
            int lado;
            Console.WriteLine("Deseas Calcular el are de un Poligono");

            Console.WriteLine("Puedes Ingresar un lado: ");
            lado = int.Parse(Console.ReadLine ());

            Console.Write("El Perimetro del Poligono regular de 5 lados es : " + lado
* 5);

            Console.ReadKey();

        }
    }
}
```

Test De Autoevaluación

1. ¿Qué es un ciclo?

Es una secuencia de instrucciones de código que se ejecuta repetidas veces, hasta que la condición asignada a dicho deja de cumplirse, entre ellos los más utilizados son:

- Bucle while
- Bucle for
- Bucle

2. ¿Cuáles son las partes del ciclo for?

Está conformado por tres partes:

- Inicio del Ciclo, desde el punto donde comienza.
- Condición, Limite del ciclo, el cual se repite mientras la condición sea verdadera.
- Incremento o decremento, el cual puede ser de 1 en 1 o de x en x, de forma positiva o negativa.

3. ¿Cómo colocamos el valor inicial de conteo en un ciclo for?

For (int i = Valor Inicial;

4. ¿Cómo colocamos el valor final de conteo en un ciclo for?

i <= Valor final; i = i + paso)

5. ¿Cómo se lleva a cabo el incremento en un ciclo for?

Se utiliza en las ocasiones en las cuales se conoce la cantidad de veces que se quiere ejecutar un bloque de instrucciones.

6. ¿Cómo funciona el ciclo do while?

Es una estructura de control cíclica que permite ejecutar de manera repetitiva un bloque de instrucciones sin evaluar de forma inmediata, sino que se evaluara después de ejecutar por primera vez el bloque de instrucciones.

7. ¿Por qué el ciclo while se lleva a cabo al menos una vez?

Para verificar que esté controlado por la condición, si esa condición se cumple, se ejecutaran las condiciones de ese bucle.

8. ¿Se necesita punto y coma al finalizar el ciclo do while?

No es necesario, ya que si se escribe punto indicará que dicha estructura termina en ese punto y no repetirá las instrucciones.



9. ¿Cómo funciona el ciclo while?

Permite ejecutar un número de veces determinado una serie de instrucciones.

10. ¿Cuántas veces se puede repetir el ciclo while?

Este se ejecuta siempre una vez y al final se evalúa la condición para decidir si se ejecuta otra vez o se termina su ejecución.

1. Hacer un programa que muestre la tabla de multiplicar del 1 al 10 de cualquier número.

```
using System;

namespace Tablas_de_Multiplicar
{
    class Program
    {
        static void Main(string[] args)
        {
            for (int i = 1; i <= 10; i++)
            {
                Console.WriteLine("\nTabla de multiplicar del {0}", i);
                Console.WriteLine("-----");
                for (int j = 1; j <= 10; j++)
                {
                    Console.WriteLine("{0} * {1} = {2}", i, j, (i *
j));
                }
            }
        }
    }
}
```



2. Hacer un programa que calcule el resultado de un número elevado a cualquier potencia.

```
using System;

namespace Tablas_de_Multiplicar
{
    class Program
    {
        static void Main(string[] args)
        {
            for (int i = 1; i <= 10; i++)
            {
                Console.WriteLine("\nTabla de multiplicar del {0}", i);
                Console.WriteLine("-----");
                for (int j = 1; j <= 10; j++)
                {
                    Console.WriteLine("{0} * {1} = {2}", i, j, (i *
j));
                }
            }
        }
    }
}
```



3. Hacer un programa que encuentre los números primos que existen entre el 1 y el 1000.

```
using System;

namespace NUMEROS_PRIMOS_DE_1_AL_1000
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.SetCursorPosition(0, 0);
            Console.WriteLine("Muestrame los Numeros Primos del 1 al 1000");
            Console.SetCursorPosition(1, 1);
            Console.WriteLine("");
            int cuentap = 0;
            for (int i = 0; i <= 1000; i++)
            {
                if (primo(i))
                    cuentap++;
            }
            int x = 2, y = 2;
            int[] vecp = new int[cuentap];
            int indice = 0;
            for (int i = 1; i <= 1000; i++)
            {
                if (primo(i))
                {
                    vecp[indice] = i;
                    Console.SetCursorPosition(x, y);
                    Console.WriteLine(i);
                    indice++;
                    y++;

                    if (y == 25)
                    {
                        x += 5;
                        y = 2;
                    }
                }
            }
            Console.ReadKey();
        }
        public static bool primo(int n)
        {
            int div = 2;
            while (div < n)
            {
                if (n % div == 0)
                    return false;
                div++;
            }
            return true;
        }
    }
}
```



4. Hacer un programa que calcule el promedio de edad de un grupo de personas y diga cuál es la de edad más grande y cuál es la más joven.

```
using System;
```

```
namespace Promedio_Edad_Por_Persona
```

```
{
```

```
    class Program
```

```
    {
```

```
        static void Main(string[] args)
```

```
        {
```

```
            int cant, edad, mayor = 0, menor = 0, acumulador = 0;
```

```
            Console.Write("Cantidad de Personas: ");
```

```
            cant = int.Parse(Console.ReadLine());
```

```
            for (int i = 0; i < cant; i++)
```

```
            {
```

```
                Console.Write(": ");
```

```
                edad = int.Parse(Console.ReadLine());
```

```
                acumulador = acumulador + edad;
```

```
                if (i == 0)
```

```
                {
```

```
                    mayor = edad;
```

```
                    menor = edad;
```

```
                }
```

```
                else if (i != 0)
```

```
                {
```

```
                    if (edad < menor)
```

```
                    {
```

```
                        menor = edad;
```

```
                    }
```

```
                    if (edad > mayor)
```

```
                    {
```

```
                        mayor = edad;
```

```
                    }
```

```
                }
```

```
            }
```

```
            Console.Write("El Promedio de edades del Grupo es:" + acumulador);
```

```
            Console.WriteLine();
```

```
            Console.Write(" El de Mayor edad tiene:" + mayor);
```

```
            Console.WriteLine();
```

```
            Console.Write("El de Menor Edad tiene:" + menor);
```

```
            Console.WriteLine();
```

```
            Console.ReadKey();
```

```
        }
```

```
    }
```

```
}
```



• Hacer los siguientes ejercicios:

calcular la factorial de un número

```
using System;

namespace Calcular_el_Factorial_de_un_Numero
{
    class Program
    {
        static void Main(string[] args)
        {
            int num;
            Console.WriteLine("Calculo del Factorial de un número \n");
            do
            {
                Console.Write("Puedes ingresar un Numero: ");
                num = int.Parse(Console.ReadLine());
                verificar(num);
            } while (num < 0);
            factorial(num);
            Console.ReadKey();
        }
        static void verificar(int n)
        {
            if (n < 0)
            {
                Console.WriteLine("\nError!!! No existe un numero entero negativo...\n");
            }
        }
        static void factorial(int n)
        {
            int fact = 1;
            for (int i = 1; i <= n; i++)
            {
                fact *= i;
            }
            Console.WriteLine("\nEl Factorial de " + n + " es: " + fact);
        }
    }
}
```



calcular la serie de un numero

```
using System;

namespace FibonacciNumbers
{
    class Program
    {
        static void Main()
        {
            Console.WriteLine("Escribe un numero");

            int cantidad = int.Parse(Console.ReadLine());
            long a = 0;
            long b = 1;
            long aux = 0;

            for (int i = 0; i < cantidad; i++)
            {
                if (i == 0)
                {
                    Console.Write("{0}", a);
                }
                else
                {
                    aux = a;
                    a = b;
                    b = aux + a;
                    Console.Write(", {0}", a);
                }
            }
            Console.WriteLine();
        }
    }
}
```