

Archivo Editar Selección Ver Ir Ejecutar ... TP_Grupal

EXPLORADOR

TP_GRUPAL

tp_grupal.py

Bienvenido tp_grupal.py X

```
1 # Definición de Funciones
2
3 def decimal_binario(decimal): # Función que convierte un numero decimal en binario
4     binario = ""
5     while decimal > 0:
6         residuo = decimal % 2
7         binario = str(residuo) + binario
8         decimal //= 2
9     return validar_binario(binario)
10
11 def validar_binario(binario):
12     if binario: # Verifica si la cadena 'binario' no está vacía
13         return binario
14     else:
15         return "0"
16
17 def binario_decimal(binario):
18     decimal = 0
19     potencia = len(binario) - 1
20     for digito in binario:
21         decimal += int(digito) * (2 ** potencia)
22         potencia -= 1
23     return decimal
24
25 # Programa principal
26 numero_decimal = 10
27 numero_binario = decimal_binario(numero_decimal)
28 print(f"Decimal {numero_decimal} a binario: {numero_binario}")
29
30 binario = "11001"
31 decimal = binario_decimal(binario)
32 print(f"Binario {binario} a decimal: {decimal}")
33
```

Creación del proyecto.

LÍNEA DE TIEMPO
ESQUEMA
MYSQL

0 0 0 Live Share

Lín. 1, col. 1 Espacios: 4 UTF-8 CRLF {} Python 3.13.2 Go Live

Archivo Editar Selección Ver Ir Ejecutar ... TP_Grupal

EXPLORADOR

TP_GRUPAL

tp_grupal.py

Bienvenido tp_grupal.py X

```
1 # Definición de Funciones
2
3 def decimal_binario(decimal): # Función que convierte un numero decimal en binario
4     binario = "" # Se genera una cadena de texto vacía
5     while decimal > 0:
6         residuo = decimal % 2 #Calcula el residuo de la división del número ingresado y 2
7         binario = str(residuo) + binario # Va formando el string de 0 y 1 que componen al número
8         decimal //= 2 # Calcula el cociente exacto entre el número ingresado y 2
9         return validar_binario(binario) # LLama a la función validar_binario.
10
11 def validar_binario(binario):
12     if binario: # Verifica si la cadena 'binario' no está vacía
13         return binario
14     else:
15         return "0"
16
17 def binario_decimal(binario):
18     decimal = 0
19     potencia = len(binario) - 1 # Calcula la potencia más significativa del número binario
20     for digito in binario: # Genera un ciclo que recorre el número decimal y genera la sumatoria de las potencias de 2
21         decimal += int(digito) * (2 ** potencia) # Sumatoria de las potencias de 2 por el dígito de esa posición
22         potencia -= 1 # Disminuye en uno la potencia
23     return decimal
24
25 # Programa principal
26 numero_decimal = 10
27 numero_binario = decimal_binario(numero_decimal)
28 print(f"Decimal {numero_decimal} a binario: {numero_binario}")
29
30 binario = "11001"
31 decimal = binario_decimal(binario)
32 print(f"Binario {binario} a decimal: {decimal}")
33
```

Se agrega descripción de cada línea del código

> LÍNEA DE TIEMPO
> ESQUEMA
> MYSQL

0 0 Live Share

Lín. 10, col. 1 Espacios: 4 UTF-8 CRLF {} Python 3.13.2 Go Live

EXPLORADOR

TP_GRUPAL

tp_grupal.py

LÍNEA DE TIEMPO

ESQUEMA

MYSQL

```

tp_grupal.py > decimal_binario
1 """Conversión de Números:
2 Desarrollar un programa que convierta números decimales a binarios y, de forma opcional, también de binario a decimal.
3 Extensión: Validar la entrada y mostrar mensajes de error ante datos incorrectos."""
4
5 # Definición de Funciones
6
7 def decimal_binario(decimal): # Función que convierte un numero decimal en binario
8     binario = "" # Se genera una cadena de texto vacía
9     while decimal > 0:
10         residuo = decimal % 2 #Calcula el residuo de la división del número ingresado y 2
11         binario = str(residuo) + binario # Va formando el string de 0 y 1 que componen al numero binario final
12         decimal //= 2 # Calcula el cociente exacto entre el número ingresado y 2
13     return validar_binario(binario) # LLama a la función validar_
14
15 def validar_binario(binario):
16     if binario: # Verifica si la cadena 'binario' no está vacía
17         return binario
18     else:
19         return "0"
20
21 def binario_decimal(binario):
22     decimal = 0
23     potencia = len(binario) - 1 # Calcula la potencia más significativa del número binario
24     for digito in binario: # Genera un ciclo que recorre el número decimal y genera la sumatoria de las potencias de 2
25         decimal += int(digito) * (2 ** potencia) # Sumatoria de las potencias de 2 por el dígito de esa posición
26         potencia -= 1 # Disminuye en uno la potencia
27     return decimal
28
29 # Programa principal
30 numero_decimal = 10
31 numero_binario = decimal_binario(numero_decimal)
32 print(f"Decimal {numero_decimal} a binario: {numero_binario}")
33
34 binario = "11001"
35 decimal = binario_decimal(binario)
36 print(f"Binario {binario} a decimal: {decimal}")
    
```

Se agrega
enunciado

Visual Studio Code interface showing a Python file named `TpGrupal_MatematicaProgramacion.py` and a terminal window.

File Explorer (EXPLORADOR):

- TPGRUPAL_MATEMATICAPROGR...
- HistorialDeCambio... U
- TpGrupal_MatematicaPr...

Code Editor:

```
1 """Conversión de Números:
2 Desarrollar un programa que convierta números decimales a binarios y, de forma opcional, también de binario a decimal.
3 Extensión: Validar la entrada y mostrar mensajes de error ante datos incorrectos."""
4 ###Integrantes:
5 # * Contreras, Agustín - DNI:
6 # * Cortes, Leonardo - DNI:
7 # * Cuello, Franco - DNI:
8 # * Dagatti, German - DNI: 31230293
9 # * Demiryi, Nicolás - DNI:
10 # ###
11
12 # Definición de Funciones
13
14 def decimal_binario(decimal): # Función que convierte un número decimal a binario
15     binario = "" # Se genera una cadena de texto vacía
16     while decimal > 0:
17         residuo = decimal % 2 #Calcula el residuo de la división del número ingresado y 2
18         binario = str(residuo) + binario # Va formando el string de 0 y 1 que componen al numero binario final
19         decimal //= 2 # Calcula el cociente exacto entre el número ingresado y 2
20     return validar_binario(binario) # LLama a la función validar_binario.
21
22 def validar_binario(binario):
23     if binario: # Verifica si la cadena 'binario' no está vacía
24         return binario
25     else:
```

Terminal:

```
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.03 KiB | 1.03 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/gdagatti/TpGrupal_MatematicaProgramacion.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

Ger@DESKTOP-4M6GRQJ MINGW64 ~/OneDrive/UTN_TUPaD/1 Cuatrimestre/Matematica I/TpGrupal_MatematicaProgramacion (main)
$
```

Terminal Output:

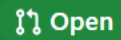
```
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.03 KiB | 1.03 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/gdagatti/TpGrupal_MatematicaProgramacion.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Terminal Prompt:

```
Ger@DESKTOP-4M6GRQJ MINGW64 ~/OneDrive/UTN_TUPaD/1 Cuatrimestre/Matematica I/TpGrupal_MatematicaProgramacion (main)
$
```

Se agregan
integrantes del
proyectos.
Se carga proyecto a
GitHub

Update TpGrupal_MatematicaProgramacion.py #1

[Edit](#)[Code](#)**Open**

Cybert18 wants to merge 1 commit into `gdagatti:main` from `Cybert18:patch-1`

Trabajo colaborativo en
GitHub

[Conversation](#) 0[Commits](#) 1[Checks](#) 0[Files changed](#) 1[+5 -5](#)

Cybert18 commented 10 hours ago

First-time contributor

Modifique el return de la primer funcion



Update TpGrupal_MatematicaProgramacion.py

Verified

688446d



No conflicts with base branch

Merging can be performed automatically.

Merge pull request

You can also merge this with the command line. [View command line instructions.](#)



Add a comment

Write

Preview

H

B

I

≡

<>

🔗

≡

≡

≡

📎

@

👤

↩

📄

Reviewers



Suggestions

gdagatti

[Request](#)

Still in progress? [Convert to draft](#)

Assignees



No one—[assign yourself](#)

Labels



None yet

Projects



None yet

Milestone



No milestone

Open Cybert18 wants to merge 1 commit into gdagatti:main from Cybert18:patch-1

Conversation 0 Commits 1 Checks 0 Files changed 1

+5 -5

despace

Review changes

Update TpGrupal_MatematicaProgramacion.py

Modifique el return de la primer funcion

Cybert18 authored 10 hours ago Verified

commit 688446d269ba343be262fbfe1f695fe23805ada5

10 TpGrupal_MatematicaProgramacion.py

...

@@ -3,7 +3,7 @@

```
3 3 Extensión: Validar la entrada y mostrar mensajes de error ante datos incorrectos.""
4 4 ###Integrantes:
5 5 # * Contreras, Agustín - DNI:
6 - # * Cortes, Leonardo - DNI:
6 + # * Cortes, Leonardo - DNI: 34711991
7 7 # * Cuello, Franco - DNI:
8 8 # * Dagatti, German - DNI: 31230293
9 9 # * Demiryi, Nicolás - DNI:
```

@@ -16,8 +16,8 @@ def decimal_binario(decimal): # Función que convierte un número decimal a bina

```
16 16 while decimal > 0:
17 17     residuo = decimal % 2 #Calcula el residuo de la división del número ingresado y 2
18 18     binario = str(residuo) + binario # Va formando el string de 0 y 1 que componen al numero binario final
19 - decimal //= 2 # Calcula el cociente exacto entre el número ingresado y 2
20 - return validar_binario(binario) # LLama a la función validar_binario.
19 + decimal //= 2 # Actualiza el valor del decimal dividiendolo por 2
20 + return validar_binario(binario) # LLama a la función validar_binario
```

Problemas al momento de
hacer el primer Merge..
Todo tranquilo, se pudo
solucionar

Visual Studio Code interface showing a merge conflict resolution process.

Left Panel (Source Control):

- Repository: `2f09e3c - Merge branch 'main' ... (1 archivo)`
- Branch: `main`
- Commit: `2f09e3c`
- Changes: 2 files changed, 15 insertions(+), 9 deletions(-)
- Commit Message: `Se agrega panel de selección de opciones, y posibilidad de ingreso de número`

Editor (TpGrupal_MatematicaProgramacion.py):

```
def binario_decimal(binario):# Función que convierte un número binario
    decimal = 0
    potencia = len(binario) - 1 # Calcula la potencia más significativa
    for digito in binario: # Genera un ciclo que recorre el número deci
        decimal += int(digito) * (2 ** potencia) # Sumatoria de las pot
        potencia -= 1 # Disminuye en uno la potencia
```

Terminal:

```
PS C:\Users\Ger\OneDrive\UTN_TUPaD\1 Cuatrimestre\Matematica I\TpGrupal_MatematicaProgramacion> python -u "c:\Users\Ger\OneDrive\UTN_TUPaD\1 Cuatrimestre\Matematica I\TpGrupal_MatematicaProgramacion\TpGrupal_MatematicaProgramacion.py"
Si quiere convertir un numero de DECIMAL a BINARIO, presione D
Si quiere convertir un número de BINARIO a DECIMAL, presione B
Ingrese la opcion seleccionada: b
Ingrese su número en base 2: 1111101
Binario 1111101 a decimal: 125
PS C:\Users\Ger\OneDrive\UTN_TUPaD\1 Cuatrimestre\Matematica I\TpGrupal_MatematicaProgramacion> git add .
PS C:\Users\Ger\OneDrive\UTN_TUPaD\1 Cuatrimestre\Matematica I\TpGrupal_MatematicaProgramacion> git commit -m "Se agrega panel de selección de opciones, y posibilidad de ingreso de número"
[main 29c34ac] Se agrega panel de selección de opciones, y posibilidad de ingreso de número
2 files changed, 15 insertions(+), 9 deletions(-)
PS C:\Users\Ger\OneDrive\UTN_TUPaD\1 Cuatrimestre\Matematica I\TpGrupal_MatematicaProgramacion> git push origin main
To https://github.com/gdagatti/TpGrupal_MatematicaProgramacion.git
! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'https://github.com/gdagatti/TpGrupal_MatematicaProgramacion.git'
hint: Updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
PS C:\Users\Ger\OneDrive\UTN_TUPaD\1 Cuatrimestre\Matematica I\TpGrupal_MatematicaProgramacion> git pull
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (4/4), 1.92 KiB | 30.00 KiB/s, done.
From https://github.com/gdagatti/TpGrupal_MatematicaProgramacion
aaad0ae..7e0994a  main      -> origin/main
Auto-merging TpGrupal_MatematicaProgramacion.py
CONFLICT (content): Merge conflict in TpGrupal_MatematicaProgramacion.py
Automatic merge failed; fix conflicts and then commit the result.
PS C:\Users\Ger\OneDrive\UTN_TUPaD\1 Cuatrimestre\Matematica I\TpGrupal_MatematicaProgramacion> git push origin main
To https://github.com/gdagatti/TpGrupal_MatematicaProgramacion.git
! [rejected]        main -> main (non-fast-forward)
error: failed to push some refs to 'https://github.com/gdagatti/TpGrupal_MatematicaProgramacion.git'
```

Right Panel (Output):

- bash
- Code
- Python

Bottom Bar:

- main
- 0
- Live Share

Problemas al momento de hacer el primer Merge..
Todo tranquilo, se pudo solucionar

Visual Studio Code interface showing a Python file named `TpGrupal_MatematicaProgramacion.py` with a function `decimal_binario` and a main program.

```
30 potencia = len(binario) - 1 # Calcula la potencia más significativa del número binario
31 for digito in binario: # Genera un ciclo que recorre el número decimal y genera la sumatoria de las potencias de 2
32     decimal += int(digito) * (2 ** potencia) # Sumatoria de las potencias de 2 por el dígito de esa posición
33     potencia -= 1 # Disminuye en uno la potencia
34 return decimal
35
36 # Programa principal
37 print("Si quiere convertir un numero de DECIMAL a BINARIO, presione D")
38 print("Si quiere convertir un número de BINARIO a DECIMAL, presione B")
39 opcion=input("Ingrese la opcion seleccionada: ").upper()
40 if opcion == "D":
41     numero_decimal = int(input("Ingrese su número en base 10: "))
42     numero_binario = decimal_binario(numero_decimal)
43     print(f"Decimal {numero_decimal} a binario: {numero_binario}")
44 elif opcion == "B":
45     binario = input("Ingrese su número en base 2: ")
46     decimal = binario_decimal(binario)
47     print(f"Binario {binario} a decimal: {decimal}")
48 else:
49     print("La opción escogida no es válida")
50     exit()
51
```

A red bracket highlights the `decimal_binario` function and its usage in the main program.

A text box on the right states: "A continuación vamos a mostrar como se utilizó IA (Copilot) Para optimizar y mejorar el programa. Nos centralizamos en el programa principal."

The terminal output shows the execution of the program:

```
PS C:\Users\Ger\OneDrive\UTN_TUPaD\1 Cuatrimestre\Matematica I\TpGrupal_MatematicaProgramacion> python -u "c:\Users\Ger\OneDrive\UTN_TUPaD\1 Cuatrimestre\Matematica I\TpGrupal_MatematicaProgramacion\TpGrupal_MatematicaProgramacion.py"
Si quiere convertir un numero de DECIMAL a BINARIO, presione D
Si quiere convertir un número de BINARIO a DECIMAL, presione B
Ingrese la opcion seleccionada: b
Ingrese su número en base 2: 1111101
Binario 1111101 a decimal: 125
PS C:\Users\Ger\OneDrive\UTN_TUPaD\1 Cuatrimestre\Matematica I\TpGrupal_MatematicaProgramacion>
```


Hoy

Optimización de Código e...

Representación Binaria de...

Ayer

Conversión Decimal-Binari...

¿Cómo optimizar esta secuencia de código en python?

¡Por supuesto! Comparte tu código y con gusto te ayudaré a optimizarlo. Puedo sugerirte mejoras en rendimiento, legibilidad y eficiencia. 🚀

```
print("Si quiere convertir un numero de DECIMAL a BINARIO, presione D")
print("Si quiere convertir un número de BINARIO a DECIMAL, presione B")
opcion=input("Ingrese la opcion seleccionada: ").upper() if opcion == "D":
numero_decimal = int(input("Ingrese su número en base 10: "))
numero_binario = decimal_binario(numero_decimal) print(f"Decimal
{numero_decimal} a binario: {numero_binario}") elif opcion == "B": binario
= input("Ingrese su número en base 2: ") decimal =
binario_decimal(binario) print(f"Binario {binario} a decimal: {decimal}")
else: print("La opción escogida no es válida") exit()
```

¡Buena lógica! Pero podemos optimizar tu código mejorando la eficiencia y la claridad. Aquí hay algunas mejoras que podrías considerar:

1. **Uso de funciones:** En lugar de tener toda la lógica en el flujo principal, es mejor encapsularla en funciones para modularidad y reutilización.
2. **Validación de entrada:** Asegúrate de que los valores ingresados sean correctos



Enviar un mensaje a Copilot



Hoy

Optimización de Código e...

Representación Binaria de...

Ayer

Conversión Decimal-Binari...

¡Buena lógica! Pero podemos optimizar tu código mejorando la eficiencia y la claridad. Aquí hay algunas mejoras que podrías considerar:

1. **Uso de funciones:** En lugar de tener toda la lógica en el flujo principal, es mejor encapsularla en funciones para modularidad y reutilización.
2. **Validación de entrada:** Asegúrate de que los valores ingresados sean correctos antes de proceder con la conversión.
3. **Simplificación de la salida:** Evita redundancia en el código con estructuras más concisas.

Aquí tienes una versión optimizada:

Python

Copiar

```
def decimal_a_binario(numero):  
    return bin(numero)[2:]  
  
def binario_a_decimal(numero):  
    return int(numero, 2)  
  
def main():  
    print("Seleccione una opción:")  
    print("[D] Convertir de DECIMAL a BINARIO")  
    print("[B] Convertir de BINARIO a DECIMAL")  
  
    opcion = input("Ingrese la opción seleccionada: ").strip().upper()  
  
    if opcion == "D":  
        try:  
            numero_decimal = int(input("Ingrese su número en base 10: "))  
            print(f"Decimal {numero_decimal} a binario: {decimal_a_binario(numero_decimal)}")  
        except ValueError:  
            print("Error: El número ingresado no es un número válido en base 10.")  
    elif opcion == "B":  
        try:  
            numero_binario = input("Ingrese su número en base 2: ")  
            print(f"Binario {numero_binario} a decimal: {binario_a_decimal(numero_binario)}")  
        except ValueError:  
            print("Error: El número ingresado no es un número válido en base 2.")  
    else:  
        print("Opción no válida. Por favor, seleccione D o B.")
```



Enviar un mensaje a Copilot



Hoy

Optimización de Código e...

Representación Binaria de...

Ayer

Conversión Decimal-Binari...

```
except ValueError:
    print("Error: Ingrese un número válido.")

elif opcion == "B":
    binario = input("Ingrese su número en base 2: ").strip()
    if all(d in "01" for d in binario): # Verifica que la entrada sea
        print(f"Binario {binario} a decimal: {binario_a_decimal(binari
    else:
        print("Error: Ingrese un número binario válido.")

else:
    print("La opción escogida no es válida.")

if __name__ == "__main__":
    main()
```

Mejoras aplicadas:

- ✓ **Funciones reutilizables** para cada conversión.
- ✓ **Validaciones de entrada** para evitar errores.
- ✓ **Uso de métodos integrados** (`bin()` y `int(base)`) para convertir valores sin necesidad de escribir lógica extra.
- ✓ **Evitar** `exit()`, pues interrumpe el programa abruptamente; en su lugar, se usa una mejor gestión de opciones.

Esta versión es más robusta y fácil de leer. ¿Te gustaría añadir alguna característica adicional? 🚀

No se utilizaron las funciones integradas `bin()` y `int(base)`, ya que queríamos presentar una opción que mostrara el proceso de cambio de un sistema a otro.



Hoy

Optimización de Código e...

Representación Binaria de...

Ayer

Conversión Decimal-Binari...

desarrolla `if name == "main": main()`

La línea `if __name__ == "__main__": main()` es una convención en Python que define el punto de entrada del programa cuando se ejecuta directamente. Veamos en detalle qué significa:

Explicación:

1. `__name__` en Python:

- Cuando un archivo Python es ejecutado, la variable `__name__` toma el valor `"__main__"`, indicando que el script se está ejecutando como programa principal.
- Si el archivo se importa como módulo en otro script, `__name__` tomará el nombre del archivo en lugar de `"__main__"`.

2. Uso de `if __name__ == "__main__":`

- Esta estructura asegura que `main()` solo se ejecute cuando el script es ejecutado directamente y no cuando es importado como módulo en otro programa.
- Esto permite que las funciones y clases definidas en el archivo sean reutilizadas en otros scripts sin que el código principal se ejecute automáticamente.



Enviar un mensaje a Copilot

