Instituto Tecnologico de Monterrey

Maestría en Inteligencia Artificial Aplicada

Curso: Pruebas de Software y aseguramiento de la Calidad

Clave: TC4017.10

Profesor Título: Dr. Gerardo Padilla Zárate

Profesora Asistente: Mtra. Viridiana Rodríguez González

Estudiante: Francisco Javier Ramírez Arias

Matrícula: A01316379

##Actividad: Actividad 4.2

##Ejercicios de Programación #2

##Descripción: Convertidor de números decimales a binario y hexadecimal.

```
## Esta linea nos permite guardar el codigo en
## un archivo físico que nombraremos computeStatistics

%%writefile convertNumbers.py

## Nos indica elinterprete que va a utilizar para
## Ejecutar el archivo, en este caso python3

#!/usr/bin/env python3

"""
convertNumbers.py

Conversión de números decimales de un archivo de texto a binario y
hexadecimal.
El programa es ejecutado de la línea de comando e imprime los
```

```
resultados en la
pantalla y a un archivo nombrado ConvertionResults.txt.
"""

import sys
import time

## Definición de función
def read_numbers_from_file(file_path):
    """
    Lee los números del archivo de texto.
    """
    numbers = []

    try:
        with open(file_path, "r", encoding="utf-8") as file:
            for line_number, line in enumerate(file, start=1):
                value = line.strip()
                if not value:
                    continue
                try:
                    numbers.append(int(value))
                except ValueError:
                    print(
                        f"Warning: Invalid data on line {line_number}:
'{value}'"
                    )
    except FileNotFoundError:
        print(f"Error: File '{file_path}' not found.")
        sys.exit(1)

    return numbers   ## Regresa los números

## Definición de función
def decimal_to_binary(number):
    """
    Convierte un entero decimal a binario utilizando el complemento a
2
    a través de puros algoritmos y funciones básicas.
    """

    # Caso especial: cero
    if number == 0:
        return "0"

    # Determinar si es negativo
    is_negative = number < 0
    number = abs(number)

    # 1. Conversión decimal → binario (valor absoluto)
```

```
    binary_digits = []
    temp = number

    while temp > 0:
        remainder = temp % 2
        binary_digits.append(str(remainder))
        temp //= 2

    binary_digits.reverse()
    binary_value = "".join(binary_digits)

    # Si es positivo, regresamos el binario directo
    if not is_negative:
        return binary_value

    # 2. Complemento a 2 para números negativos

    # Asegurar un bit extra para el signo
    bits = len(binary_value) + 1
    binary_value = binary_value.zfill(bits)

    # Invertir bits (complemento a 1)
    inverted = []
    for bit in binary_value:
        if bit == '0':
            inverted.append('1')
        else:
            inverted.append('0')

    # Sumar 1 (complemento a 2)
    carry = 1
    for i in range(len(inverted) - 1, -1, -1):
        if inverted[i] == '1' and carry == 1:
            inverted[i] = '0'
            carry = 1
        elif inverted[i] == '0' and carry == 1:
            inverted[i] = '1'
            carry = 0

    return "".join(inverted)

## Definición de función
def decimal_to_binary_and_hexadecimal(number):
    """
    Conversión de decimal a:
    - binarios (utiliza complemento a 2, si es negativo)
    - hexadecimal (de la representación binaria, )

    Regresa: (decimal, binario, hexadecimal)
    """
```

```
    hex_symbols = "0123456789ABCDEF"

    # Caso especial
    if number == 0:
        return 0, "0", "0"

    # Conversión de decimal a binario
    is_negative = number < 0
    abs_number = abs(number)

    # Conversión decimal → binario (valor absoluto)
    binary_digits = []
    temp = abs_number

    while temp > 0:
        binary_digits.append(str(temp % 2))
        temp //= 2

    binary_digits.reverse()
    binary_value = "".join(binary_digits)

    # Si es negativo, aplicar complemento a 2
    if is_negative:
        # Agregar bit de signo
        bits = len(binary_value) + 1
        binary_value = binary_value.zfill(bits)

        # Complemento a 1
        inverted = []
        for bit in binary_value:
            inverted.append('1' if bit == '0' else '0')

        # Sumar 1 (complemento a 2)
        carry = 1
        for i in range(len(inverted) - 1, -1, -1):
            if inverted[i] == '1' and carry == 1:
                inverted[i] = '0'
            elif inverted[i] == '0' and carry == 1:
                inverted[i] = '1'
                carry = 0

        binary_value = "".join(inverted)

    # Ajuste correcto de los bits
    sign_bit = '1' if is_negative else '0'
    while len(binary_value) % 4 != 0:
        binary_value = sign_bit + binary_value

    # Conversión binario → hexadecimal
```

```
    hex_digits = []

    for i in range(0, len(binary_value), 4):
        nibble = binary_value[i:i+4]
        value = 0
        for bit in nibble:
            value = value * 2 + int(bit)
        hex_digits.append(hex_symbols[value])

    hex_value = "".join(hex_digits)

    return number, binary_value, hex_value

## Definición de función
def write_results(output_lines_to_write):
    """
    Escribe los resultados en el archivo ConvertionResults.txt.
    """
    with open("ConvertionResults.txt", "w", encoding="utf-8") as file:
        for line in output_lines_to_write:
            file.write(line + "\n")

## Programa principal
def main():
    """
    Punto de entrada del programa.
    """
    start_time = time.perf_counter() # Inicialización del temporizador

    if len(sys.argv) != 2:
        print("Usage: python convertNumbers.py <input_file>")
        sys.exit(1)

    file_path = sys.argv[1]
    numbers = read_numbers_from_file(file_path)

    if not numbers:
        print("Error: No valid numeric data found.")
        sys.exit(1)

    output_lines = []
    output_lines.append("Decimal | Binary | Hexadecimal")
    output_lines.append("-" * 40)

    for number in numbers:
        dec, binary_value, hex_value =
decimal_to_binary_and_hexadecimal(number)
        line = f"{dec} | {binary_value} | {hex_value}"
        output_lines.append(line)
        print(line)
```

```
    end_time = time.perf_counter()           # Finalización de
temporizador
    elapsed_time = end_time - start_time     # Tiempo de ejecución

    time_line = f"Execution Time (seconds): {elapsed_time}"
    output_lines.append("-" * 40)
    output_lines.append(time_line)

    print(time_line)
    write_results(output_lines)


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

```
Writing convertNumbers.py
```

Conversión de los datos del archivo TC1.txt

```
!python convertNumbers.py TC1.txt
```

```
6980368 | 011010101000001100010000 | 6A8310
5517055 | 010101000010111011111111 | 542EFF
1336159 | 000101000110001101011111 | 14635F
6750185 | 011001101111111111101001 | 66FFE9
1771937 | 000110110000100110100001 | 1B09A1
360952 | 01011000000111111000 | 581F8
5672561 | 010101101000111001110001 | 568E71
916583 | 11011111110001100111 | DFC67
2700138 | 001010010011001101101010 | 29336A
9645053 | 100100110010101111111101 | 932BFD
1181110 | 000100100000010110110110 | 1205B6
1492185 | 000101101100010011011001 | 16C4D9
4018595 | 001111010101000110100011 | 3D51A3
7654888 | 011101001100110111101000 | 74CDE8
7062453 | 011010111100001110110101 | 6BC3B5
2478010 | 001001011100111110111010 | 25CFBA
6134768 | 010111011001101111110000 | 5D9BF0
8420417 | 100000000111110001000001 | 807C41
2917489 | 001011001000010001110001 | 2C8471
3340773 | 001100101111100111100101 | 32F9E5
1115956 | 000100010000011100110100 | 110734
9172192 | 100010111111010011100000 | 8BF4E0
6271996 | 010111111011001111111100 | 5FB3FC
8686939 | 100001001000110101011011 | 848D5B
50986 | 1100011100101010 | C72A
9376410 | 100011110001001010011010 | 8F129A
```

5962327 | 010110101111101001010111 | 5AFA57
7686891 | 011101010100101011101011 | 754AEB
6615183 | 011001001111000010001111 | 64F08F
1864844 | 000111000111010010001100 | 1C748C
3329962 | 001100101100111110101010 | 32CFAA
3942794 | 001111000010100110001010 | 3C298A
2614836 | 001001111110011000110100 | 27E634
7406772 | 011100010000010010110100 | 7104B4
2384190 | 001001000110000100111110 | 24613E
398347 | 0110000101000000 1011 | 6140B
8698503 | 100001001011101010000111 | 84BA87
9551696 | 100100011011111101010000 | 91BF50
1019556 | 11111000111010100100 | F8EA4
1677430 | 000110011001100001110110 | 199876
3479629 | 001101010001100001001101 | 35184D
9309008 | 100011100000101101010000 | 8E0B50
5266170 | 010100000101101011111010 | 505AFA
4094340 | 001111100111100110000100 | 3E7984
1754055 | 000110101100001111000111 | 1AC3C7
5861132 | 010110010110111100001100 | 596F0C
4471329 | 010001000011101000100001 | 443A21
8826052 | 100001101010110011000100 | 86ACC4
7469325 | 011100011111100100001101 | 71F90D
1973172 | 000111100001101110110100 | 1E1BB4
53145 | 1100111110011001 | CF99
3897508 | 001110110111100010100100 | 3B78A4
7773386 | 011101101001110011001010 | 769CCA
6089829 | 010111001110110001100101 | 5CEC65
4223424 | 010000000111000111000000 | 4071C0
9761752 | 100101001111001111011000 | 94F3D8
7930799 | 011110010000001110101111 | 7903AF
3597495 | 001101101110010010110111 | 36E4B7
9302948 | 100011011111001110100100 | 8DF3A4
2288712 | 001000101110110001001000 | 22EC48
197187 | 0011000000 1001000011 | 30243
5266939 | 010100000101110111111011 | 505DFB
221545 | 00110110000101101001 | 36169
7957027 | 011110010110101000100011 | 796A23
3195361 | 001100001100000111100001 | 30C1E1
7106269 | 011011000110111011011101 | 6C6EDD
9633312 | 100100101111111000100000 | 92FE20
9713704 | 100101000011100000101000 | 943828
91925 | 00010110011100010101 | 16715
4418686 | 010000110110110001111110 | 436C7E
9682250 | 100100111011110101001010 | 93BD4A
2583824 | 001001110110110100010000 | 276D10
4979126 | 010010111111100110110110 | 4BF9B6
6280954 | 010111111101011011111010 | 5FD6FA
1228610 | 000100101011111101000010 | 12BF42

705518 | 10101100001111101110 | AC3EE
1017653 | 11111000011100110101 | F8735
500098 | 01111010000110000010 | 7A182
7210727 | 01101110000011011100111 | 6E06E7
4250898 | 01000000110111010010010 | 40DD12
4055028 | 00111101110111111110100 | 3DDFF4
2754240 | 00101010000011011000000 | 2A06C0
452999 | 01101110100110000111 | 6E987
6831458 | 01101000001111010110010 | 683D62
207636 | 00110010101100010100 | 32B14
7280635 | 01101111000101111111011 | 6F17FB
3308937 | 00110010011111010110001001 | 327D89
4303570 | 01000001101010101010010 | 41AAD2
8375055 | 01111111110010110000111 | 7FCB0F
1457960 | 00010110000111111100101000 | 163F28
5197625 | 01001111010011110011001 | 4F4F39
3144371 | 00101111111110101010011 | 2FFAB3
6674138 | 01100101110101101101010 | 65D6DA
2692106 | 00101001000101000001010 | 29140A
2276769 | 00100010101111010110100001 | 22BDA1
1940971 | 00011101100111010111101011 | 1D9DEB
3288264 | 00110010000101100110001000 | 322CC8
4819803 | 01001001100010110101011 | 498B5B
9431005 | 10001111111100111111011101 | 8FE7DD
3992019 | 00111100111010011101011 | 3CE9D3
8184782 | 01111100111000111001110 | 7CE3CE
2847975 | 00101011011101001110011 | 2B74E7
7891025 | 01111000011010000101001 | 786851
900082 | 11011011101111110010 | DBBF2
1831532 | 00011011111100100110100 | 1BF26C
8428253 | 10000000100110101101101 | 809ADD
2905752 | 00101100010101101001100 | 2C5698
9763121 | 10010100111110010010001 | 94F931
257890 | 00111110111101100010 | 3EF62
4699761 | 01000111101101100110001 | 47B671
359541 | 01010111110001110101 | 57C75
3446150 | 00110100100101011000110 | 349586
4246818 | 01000000110011010010010 | 40CD22
1848840 | 00011100001101100001000 | 1C3608
938480 | 11100101000111110000 | E51F0
227465 | 00110111100010001001 | 37889
3923488 | 00111011110111000100000 | 3BDE20
8915697 | 10001000000101011110001 | 880AF1
4309094 | 01000001110000000110110 | 41C066
3891251 | 00111011011000000110011 | 3B6033
8627956 | 10000011101001101110100 | 83A6F4
5884613 | 01011001110010101100101 | 59CAC5
1659318 | 00011001010100011011011O | 1951B6
1834855 | 00011011111111110100111 | 1BFF67

3386751 | 001100111010110101111111 | 33AD7F
3000166 | 001011011100011101100110 | 2DC766
9135626 | 100010110110011000001010 | 8B660A
4869260 | 010010100100110010001100 | 4A4C8C
76550 | 00010010101100000110 | 12B06
432271 | 01101001100010001111 | 6988F
251028 | 00111101010010010100 | 3D494
7016218 | 011010110000111100011010 | 6B0F1A
6896099 | 011010010011100111100011 | 6939E3
8386350 | 011111111111011100101110 | 7FF72E
8637147 | 100000111100101011011011 | 83CADB
936705 | 11100100101100000001 | E4B01
6602175 | 011001001011110110111111 | 64BDBF
1429181 | 000101011100111010111101 | 15CEBD
8395138 | 100000000001100110000010 | 801982
6132809 | 010111011001010001001001 | 5D9449
5936917 | 010110101001011100010101 | 5A9715
2878578 | 001010111110110001110010 | 2BEC72
158885 | 00100110110010100101 | 26CA5
2441957 | 001001010100001011100101 | 2542E5
5914794 | 010110100100000010101010 | 5A40AA
3999272 | 001111010000011000101000 | 3D0628
3142897 | 001011111111010011110001 | 2FF4F1
8151159 | 011111000110000001110111 | 7C6077
5147564 | 010011101000101110101100 | 4E8BAC
4595374 | 010001100001111010101110 | 461EAE
4234951 | 010000001001111011000111 | 409EC7
7880605 | 011110000011111110011101 | 783F9D
7009921 | 011010101111011010000001 | 6AF681
695580 | 10101001110100011100 | A9D1C
7370443 | 011100000111011011001011 | 7076CB
7921729 | 011110001110000001000001 | 78E041
8419625 | 100000000111100100101001 | 807929
7024080 | 011010110010110111010000 | 6B2DD0
3905988 | 001110111001100111000100 | 3B99C4
1767599 | 000110101111100010101111 | 1AF8AF
935136 | 11100100010011100000 | E44E0
635788 | 10011011001110001100 | 9B38C
8807719 | 100001100110010100100111 | 866527
317375 | 01001101011110111111 | 4D7BF
9975410 | 100110000011011001110010 | 983672
2727968 | 001010011010000000100000 | 29A020
7444399 | 011100011001011110101111 | 7197AF
4065675 | 001111100000100110001011 | 3E098B
9925720 | 100101110111010001011000 | 977458
2293633 | 001000101111111110000001 | 22FF81
7734826 | 011101100000011000101010 | 76062A
1065463 | 000100000100000111110111 | 1041F7
1105617 | 000100001101111011010001 | 10DED1

```
5325800 | 01010001010000111101000 | 5143E8
3822527 | 00111010010100111011111 | 3A53BF
5503858 | 01010011111110110110010 | 53FB72
9214055 | 10001100100110000110011 | 8C9867
6521769 | 01100011100000111010001 | 6383A9
7923796 | 01111000111010000101010 | 78E854
5250236 | 01010000000111001011100 | 501CBC
1083154 | 00010000100001110001010 | 108712
472141 | 01110011010001001101 | 7344D
9597454 | 10010010011100100000111 | 92720E
1581679 | 00011000001000100110111 | 18226F
656751 | 10100000010101101111 | A056F
345464 | 01010100010101111000 | 54578
4281218 | 01000001010100111000010 | 415382
6558883 | 01100100000101001010011 | 6414A3
3852986 | 00111010110010101011010 | 3ACABA
6263187 | 01011111100100011001011 | 5F9193
5828308 | 01011000111011011010100 | 58EED4
8058535 | 01111010111011010100111 | 7AF6A7
9035191 | 10001001110111011011111 | 89DDB7
7922103 | 01111000111000011011111 | 78E1B7
9366003 | 10001110111010011111011 | 8EE9F3
4555717 | 01000101100000111100101 | 4583C5
3526753 | 00110101110100000110001 | 35D061
3176815 | 00110000011110010110111 | 30796F
858440 | 11010001100101001000 | D1948
2250854 | 00100010010110000110110 | 225866
Execution Time (seconds): 0.00507847299999753
```

Conversión de los datos del archivo TC2.txt

```
!python convertNumbers.py TC2.txt

7116776 | 01101100100101111101000 | 6C97E8
1666340 | 00011001011011010010100 | 196D24
8886983 | 10000111100110101100111 | 879AC7
839365 | 11001100111011000101 | CCEC5
924280 | 11100001101001111000 | E1A78
1026310 | 11111010100100000110 | FA906
1615293 | 00011000101001011011101 | 18A5BD
1063875 | 00010000001110111000011 | 103BC3
679035 | 10100101110001111011 | A5C7B
5201970 | 01001111011000000110010 | 4F6032
593979 | 10010001000000111011 | 9103B
801371 | 11000011101001011011 | C3A5B
3796878 | 00111001111011111000111 | 39EF8E
7489201 | 01110010010001101011001 | 7246B1
9740020 | 10010100100111101110100 | 949EF4
```

9128737 | 100010110100101100100001 | 8B4B21
5473463 | 010100111000010010110111 | 5384B7
8701957 | 100001001100100000000101 | 84C805
8238050 | 011111011011001111100010 | 7DB3E2
8679038 | 100001000110111001111110 | 846E7E
385912 | 0101111000110111111000 | 5E378
5867340 | 010110011000011101001100 | 59874C
4894542 | 010010101010111101001110 | 4AAF4E
8999451 | 100010010101001000011011 | 89521B
4392535 | 010000110000011001010111 | 430657
2078131 | 000111111011010110110011 | 1FB5B3
3070124 | 001011101101100010101100 | 2ED8AC
7451998 | 011100011011010101011110 | 71B55E
5635510 | 010101011111110110110110 | 55FDB6
1233932 | 000100101101010000001100 | 12D40C
6089867 | 010111001110110010001011 | 5CEC8B
1792316 | 000110110101100100111100 | 1B593C
6298637 | 011000000001110000001101 | 601C0D
2408038 | 001001001011111001100110 | 24BE66
8510100 | 100000011101101010010100 | 81DA94
991581 | 11110010000101011101 | F215D
6455739 | 011000101000000110111011 | 6281BB
7829175 | 011101110111011010110111 | 7776B7
6328931 | 011000001001001001100011 | 609263
9982305 | 100110000101000101100001 | 985161
2292022 | 001000101111100100110110 | 22F936
1070496 | 000100000101010110100000 | 1055A0
8518360 | 100000011111101011011000 | 81FAD8
6378470 | 011000010101001111100110 | 6153E6
7756258 | 011101100101100111100010 | 7659E2
7111767 | 011011001000010001010111 | 6C8457
2685897 | 001010001111101111001001 | 28FBC9
617946 | 10010110110111011010 | 96DDA
246280 | 00111100001000001000 | 3C208
6075442 | 010111001011010000110010 | 5CB432
991323 | 11110010000001011011 | F205B
6754202 | 011001110000111110011010 | 670F9A
3103178 | 001011110101100111001010 | 2F59CA
3228842 | 001100010100010010101010 | 3144AA
6786623 | 011001111000111000111111 | 678E3F
5329085 | 010100010101000010111101 | 5150BD
2334647 | 001000111001111110110111 | 239FB7
7253396 | 011011101010110110010100 | 6EAD94
6516953 | 011000110111000011011001 | 6370D9
5972725 | 010110110010001011110101 | 5B22F5
5828037 | 010110001110110111000101 | 58EDC5
6642265 | 011001010101101001011001 | 655A59
2559699 | 001001110000111011010011 | 270ED3
6597631 | 011001001010101111111111 | 64ABFF

5826202 | 010110001110011010011010 | 58E69A
4642335 | 010001101101011000011111 | 46D61F
1862900 | 000111000110110011110100 | 1C6CF4
6039635 | 010111000010100001010011 | 5C2853
3358744 | 001100110100000000011000 | 334018
4965249 | 010010111100001110000001 | 4BC381
6621475 | 011001010000100100100011 | 650923
3336153 | 001100101110011111011001 | 32E7D9
1044743 | 11111111000100000111 | FF107
6201477 | 010111101010000010000101 | 5EA085
6297100 | 011000000001011000001100 | 60160C
885678 | 11011000001110101110 | D83AE
8546891 | 100000100110101001001011 | 826A4B
6683923 | 011001011111110100010011 | 65FD13
7064472 | 011010111100101110011000 | 6BCB98
3430021 | 001101000101011010000101 | 345685
9764122 | 100101001111110100011010 | 94FD1A
9958605 | 100101111111010011001101 | 97F4CD
1006022 | 11110101100111000110 | F59C6
5276496 | 010100001000001101010000 | 508350
8876232 | 100001110111000011001000 | 8770C8
3364648 | 001100110101011100101000 | 335728
3539979 | 001101100000010000001011 | 36040B
3003857 | 001011011101010111010001 | 2DD5D1
4481634 | 010001000110001001100010 | 446262
9035781 | 100010011110000000000101 | 89E005
3767021 | 001110010111101011101101 | 397AED
4993901 | 010011000011001101101101 | 4C336D
8698377 | 100001001011101000001001 | 84BA09
4626501 | 010001101001100001000101 | 469845
1755352 | 000110101100100011011000 | 1AC8D8
2755340 | 001010100000101100001100 | 2A0B0C
5150616 | 010011101001011110011000 | 4E9798
9345169 | 100011101001100010010001 | 8E9891
848948 | 11001111010000110100 | CF434
2708748 | 001010010101010100001100 | 29550C
8372451 | 011111111100000011100011 | 7FC0E3
3523015 | 001101011100000111000111 | 35C1C7
2624283 | 001010000000101100011011 | 280B1B
8795348 | 100001100011010011010100 | 8634D4
2318100 | 001000110101111100010100 | 235F14
4750861 | 010010000111111000001101 | 487E0D
3770597 | 001110011000100011100101 | 3988E5
1053352 | 000100000001001010101000 | 1012A8
833307 | 11001011011100011011 | CB71B
4632526 | 010001101010111111001110 | 46AFCE
2212917 | 001000011100010000110101 | 21C435
5053189 | 010011010001101100000101 | 4D1B05
412218 | 01100100101000111010 | 64A3A

4901564 | 010010101100101010111100 | 4ACABC
5094606 | 010011011011110011001110 | 4DBCCE
8070285 | 011110110010010010001101 | 7B248D
6847905 | 011010000111110110100001 | 687DA1
4311824 | 010000011100101100010000 | 41CB10
3009421 | 001011011110101110001101 | 2DEB8D
1629105 | 000110001101101110110001 | 18DBB1
8431899 | 100000001010100100011011 | 80A91B
7582456 | 011100111011001011111000 | 73B2F8
7332147 | 011011111110000100110011 | 6FE133
314183 | 0100110010110100011 | 4CB47
7632568 | 011101000111011010111000 | 7476B8
8161327 | 011111001000100000101111 | 7C882F
6961087 | 011010100011011110111111 | 6A37BF
6898282 | 011010010100001001101010 | 69426A
9749303 | 100101001100001100110111 | 94C337
4427419 | 010000111000111010011011 | 438E9B
4368829 | 010000101010100110111101 | 42A9BD
6439318 | 011000100100000110010110 | 624196
6473603 | 011000101100011110000011 | 62C783
6446249 | 011000100101110010101001 | 625CA9
9818836 | 100101011101001011010100 | 95D2D4
9438069 | 100100000000001101110101 | 900375
2809499 | 001010101101111010011011 | 2ADE9B
3460304 | 001101001100110011010000 | 34CCD0
8430904 | 100000001010010100111000 | 80A538
9774785 | 100101010010011011000001 | 9526C1
9560603 | 100100011110001000011011 | 91E21B
3440732 | 001101001000000001011100 | 34805C
7053263 | 011010111001111111001111 | 6B9FCF
287398 | 01000110001010100110 | 462A6
4043644 | 001111011011001101111100 | 3DB37C
2943820 | 001011001110101101001100 | 2CEB4C
9030962 | 100010011100110100110010 | 89CD32
7537540 | 011100110000001110000100 | 730384
7203657 | 011011011110101101001001 | 6DEB49
5639400 | 010101100000110011101000 | 560CE8
9448142 | 100100000010101011001110 | 902ACE
9954392 | 100101111110010001011000 | 97E458
2856859 | 001010111001011110011011 | 2B979B
1332118 | 000101000101001110010110 | 145396
6984725 | 011010101001010000010101 | 6A9415
1925762 | 000111010110001010000010 | 1D6282
2138171 | 001000001010000000111011 | 20A03B
347447 | 01010100110100110111 | 54D37
8839390 | 100001101110000011011110 | 86E0DE
3951652 | 001111000100110000100100 | 3C4C24
3755102 | 001110010100110001011110 | 394C5E
6867687 | 011010001100101011100111 | 68CAE7

```
4383175 | 01000010111000011100111 | 42E1C7
7125508 | 01101100101110100000100 | 6CBA04
7799732 | 01110111000001110110100 | 7703B4
8784156 | 10000110000100100011100 | 86091C
1372809 | 00010100111100101000100 1 | 14F289
1327984 | 00010100010000110111000 0 | 144370
9205087 | 10001100011101010101111 1 | 8C755F
6524241 | 01100011100011010101000 1 | 638D51
9316072 | 10001110001001101110100 0 | 8E26E8
127744 | 0001111100110000000 0 | 1F300
8388820 | 10000000000000011010100 | 8000D4
6883557 | 01101001000010001110010 1 | 6908E5
2291239 | 00100010111101100010011 1 | 22F627
3101270 | 00101111010100100101011 0 | 2F5256
1382825 | 00010101000110011010100 1 | 1519A9
9873376 | 10010110101001111110000 0 | 96A7E0
6873500 | 01101000111000011001110 0 | 68E19C
8185634 | 01111100111001110010001 0 | 7CE722
2464786 | 00100101100111000001001 0 | 259C12
8387154 | 01111111111110100101001 0 | 7FFA52
5581444 | 01010101001010101000010 0 | 552A84
8097911 | 01111011100100000111011 1 | 7B9077
8882527 | 10000111100010010111111 | 87895F
8941444 | 10001000011011111000010 0 | 886F84
4942703 | 01001011011010110110111 1 | 4B6B6F
101144 | 0001100010110001100 0 | 18B18
7471180 | 01110010000000001001100 | 72004C
1932131 | 00011101011110110100001 1 | 1D7B63
8052752 | 01111010111000000010000 | 7AE010
6359493 | 01100001000010011100010 1 | 6109C5
1967646 | 00011110000001100001111 0 | 1E061E
6575052 | 01100100010100111100110 0 | 6453CC
2323342 | 00100011011100111000111 0 | 23738E
6735760 | 01100110110001111001000 0 | 66C790
8895858 | 10000111101111010111001 0 | 87BD72
4238091 | 01000000101010110000101 1 | 40AB0B
7093069 | 01101100001110110100110 1 | 6C3B4D
39 | 00100111 | 27
Execution Time (seconds): 0.005159722000001921
```

Conversión de los datos del archivo TC3.txt

```
!python convertNumbers.py TC3.txt

-39 | 11011001 | D9
-36 | 11011100 | DC
8 | 1000 | 8
34 | 00100010 | 22
```

17 | 00010001 | 11
49 | 00110001 | 31
5 | 0101 | 5
39 | 00100111 | 27
0 | 0 | 0
33 | 00100001 | 21
12 | 1100 | C
-6 | 1010 | A
27 | 00011011 | 1B
-4 | 1100 | C
-38 | 11011010 | DA
26 | 00011010 | 1A
49 | 00110001 | 31
29 | 00011101 | 1D
42 | 00101010 | 2A
-16 | 11110000 | F0
-28 | 11100100 | E4
34 | 00100010 | 22
20 | 00010100 | 14
0 | 0 | 0
25 | 00011001 | 19
45 | 00101101 | 2D
3 | 0011 | 3
-46 | 11010010 | D2
-46 | 11010010 | D2
29 | 00011101 | 1D
33 | 00100001 | 21
29 | 00011101 | 1D
26 | 00011010 | 1A
-5 | 1011 | B
-36 | 11011100 | DC
12 | 1100 | C
45 | 00101101 | 2D
-50 | 11001110 | CE
0 | 0 | 0
-6 | 1010 | A
-39 | 11011001 | D9
35 | 00100011 | 23
26 | 00011010 | 1A
-35 | 11011101 | DD
-42 | 11010110 | D6
14 | 1110 | E
-3 | 1101 | D
22 | 00010110 | 16
-47 | 11010001 | D1
12 | 1100 | C
-17 | 11101111 | EF
-31 | 11100001 | E1
-25 | 11100111 | E7

```
-35 | 11011101 | DD
-18 | 11101110 | EE
14 | 1110 | E
37 | 00100101 | 25
-8 | 11111000 | F8
-41 | 11010111 | D7
-50 | 11001110 | CE
-9 | 11110111 | F7
2 | 0010 | 2
12 | 1100 | C
-39 | 11011001 | D9
-17 | 11101111 | EF
0 | 0 | 0
27 | 00011011 | 1B
28 | 00011100 | 1C
45 | 00101101 | 2D
-44 | 11010100 | D4
28 | 00011100 | 1C
35 | 00100011 | 23
-33 | 11011111 | DF
43 | 00101011 | 2B
-4 | 1100 | C
23 | 00010111 | 17
-49 | 11001111 | CF
49 | 00110001 | 31
-7 | 1001 | 9
-38 | 11011010 | DA
-45 | 11010011 | D3
5 | 0101 | 5
-21 | 11101011 | EB
26 | 00011010 | 1A
-2 | 1110 | E
-24 | 11101000 | E8
6 | 0110 | 6
-45 | 11010011 | D3
-4 | 1100 | C
15 | 1111 | F
0 | 0 | 0
-17 | 11101111 | EF
-35 | 11011101 | DD
43 | 00101011 | 2B
45 | 00101101 | 2D
-35 | 11011101 | DD
-8 | 11111000 | F8
28 | 00011100 | 1C
47 | 00101111 | 2F
34 | 00100010 | 22
-24 | 11101000 | E8
-42 | 11010110 | D6
```

```
-13 | 11110011 | F3
14 | 1110 | E
-45 | 11010011 | D3
-35 | 11011101 | DD
27 | 00011011 | 1B
-12 | 11110100 | F4
-19 | 11101101 | ED
-20 | 11101100 | EC
-14 | 11110010 | F2
20 | 00010100 | 14
30 | 00011110 | 1E
12 | 1100 | C
-36 | 11011100 | DC
-47 | 11010001 | D1
-27 | 11100101 | E5
-3 | 1101 | D
6 | 0110 | 6
36 | 00100100 | 24
9 | 1001 | 9
-16 | 11110000 | F0
36 | 00100100 | 24
-27 | 11100101 | E5
7 | 0111 | 7
-6 | 1010 | A
-5 | 1011 | B
5 | 0101 | 5
47 | 00101111 | 2F
23 | 00010111 | 17
-48 | 11010000 | D0
-24 | 11101000 | E8
40 | 00101000 | 28
-4 | 1100 | C
-18 | 11101110 | EE
-19 | 11101101 | ED
14 | 1110 | E
15 | 1111 | F
1 | 0001 | 1
24 | 00011000 | 18
-9 | 11110111 | F7
-50 | 11001110 | CE
18 | 00010010 | 12
15 | 1111 | F
30 | 00011110 | 1E
44 | 00101100 | 2C
9 | 1001 | 9
19 | 00010011 | 13
33 | 00100001 | 21
41 | 00101001 | 29
-24 | 11101000 | E8
```

```
43 | 00101011 | 2B
48 | 00110000 | 30
-23 | 11101001 | E9
7 | 0111 | 7
28 | 00011100 | 1C
8 | 1000 | 8
-2 | 1110 | E
37 | 00100101 | 25
42 | 00101010 | 2A
-17 | 11101111 | EF
-6 | 1010 | A
36 | 00100100 | 24
22 | 00010110 | 16
27 | 00011011 | 1B
31 | 00011111 | 1F
-47 | 11010001 | D1
-43 | 11010101 | D5
0 | 0 | 0
27 | 00011011 | 1B
30 | 00011110 | 1E
12 | 1100 | C
-15 | 11110001 | F1
-19 | 11101101 | ED
-42 | 11010110 | D6
-37 | 11011011 | DB
31 | 00011111 | 1F
-10 | 11110110 | F6
18 | 00010010 | 12
33 | 00100001 | 21
-15 | 11110001 | F1
17 | 00010001 | 11
25 | 00011001 | 19
-25 | 11100111 | E7
-5 | 1011 | B
33 | 00100001 | 21
-13 | 11110011 | F3
33 | 00100001 | 21
-10 | 11110110 | F6
47 | 00101111 | 2F
47 | 00101111 | 2F
-13 | 11110011 | F3
-32 | 11100000 | E0
1 | 0001 | 1
1 | 0001 | 1
-25 | 11100111 | E7
-33 | 11011111 | DF
16 | 00010000 | 10
17 | 00010001 | 11
```

```
4 | 0100 | 4
Execution Time (seconds): 0.0022856079999939993
```

Conversión de los datos del archivo TC4.txt

```
!python convertNumbers.py TC4.txt

Warning: Invalid data on line 8: 'ABC'
Warning: Invalid data on line 21: 'ERR'
Warning: Invalid data on line 41: 'VAL'
-39 | 11011001 | D9
-36 | 11011100 | DC
8 | 1000 | 8
34 | 00100010 | 22
17 | 00010001 | 11
49 | 00110001 | 31
5 | 0101 | 5
0 | 0 | 0
33 | 00100001 | 21
12 | 1100 | C
-6 | 1010 | A
27 | 00011011 | 1B
-4 | 1100 | C
-38 | 11011010 | DA
26 | 00011010 | 1A
49 | 00110001 | 31
29 | 00011101 | 1D
42 | 00101010 | 2A
-16 | 11110000 | F0
34 | 00100010 | 22
20 | 00010100 | 14
0 | 0 | 0
25 | 00011001 | 19
45 | 00101101 | 2D
3 | 0011 | 3
-46 | 11010010 | D2
-46 | 11010010 | D2
29 | 00011101 | 1D
33 | 00100001 | 21
29 | 00011101 | 1D
26 | 00011010 | 1A
-5 | 1011 | B
-36 | 11011100 | DC
12 | 1100 | C
45 | 00101101 | 2D
-50 | 11001110 | CE
0 | 0 | 0
```

```
-6 | 1010 | A
Execution Time (seconds): 0.0008601460000079442
```

Instalación de Pylint

```
pip install pylint

Collecting pylint
  Downloading pylint-4.0.4-py3-none-any.whl.metadata (12 kB)
Collecting astroid<=4.1.dev0,>=4.0.2 (from pylint)
  Downloading astroid-4.0.3-py3-none-any.whl.metadata (4.4 kB)
Requirement already satisfied: dill>=0.3.6 in
/usr/local/lib/python3.12/dist-packages (from pylint) (0.3.8)
Collecting isort!=5.13,<8,>=5 (from pylint)
  Downloading isort-7.0.0-py3-none-any.whl.metadata (11 kB)
Collecting mccabe<0.8,>=0.6 (from pylint)
  Downloading mccabe-0.7.0-py2.py3-none-any.whl.metadata (5.0 kB)
Requirement already satisfied: platformdirs>=2.2 in
/usr/local/lib/python3.12/dist-packages (from pylint) (4.5.1)
Requirement already satisfied: tomlkit>=0.10.1 in
/usr/local/lib/python3.12/dist-packages (from pylint) (0.13.3)
Downloading pylint-4.0.4-py3-none-any.whl (536 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 536.4/536.4 kB 26.1 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 276.4/276.4 kB 16.9 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 94.7/94.7 kB 7.3 MB/s eta
0:00:00
ccabe-0.7.0-py2.py3-none-any.whl (7.3 kB)
Installing collected packages: mccabe, isort, astroid, pylint
Successfully installed astroid-4.0.3 isort-7.0.0 mccabe-0.7.0 pylint-
4.0.4
```

#Análisis del Código

```
!pylint convertNumbers.py

************* Module convertNumbers
convertNumbers.py:1:0: C0103: Module name "convertNumbers" doesn't
conform to snake_case naming style (invalid-name)
convertNumbers.py:103:0: R0914: Too many local variables (17/15) (too-
many-locals)

-----------------------------------
Your code has been rated at 9.83/10
```

#**Código Mejorado**

```
## Esta linea nos permite guardar el codigo en
## un archivo físico que nombraremos convertNumbers

%%writefile convertNumbers.py

## Indica al interprete que va a utilizar para
## ejecutar el archivo, en este caso python3

#!/usr/bin/env python3

"""
convertNumbers.py

Conversión de números decimales de un archivo de texto a binario y
hexadecimal.
El programa es ejecutado de la línea de comando e imprime los
resultados en la
pantalla y a un archivo nombrado ConvertionResults.txt.
"""


import sys
import time

## Definición de función
def read_numbers_from_file(file_path):
    """
    Lee los números del archivo de texto.
    """
    numbers = []

    try:
        with open(file_path, "r", encoding="utf-8") as file:
            for line_number, line in enumerate(file, start=1):
                value = line.strip()
                if not value:
                    continue
                try:
                    numbers.append(int(value))
                except ValueError:
                    print(
                        f"Warning: Invalid data on line {line_number}:
'{value}'"
                    )
    except FileNotFoundError:
        print(f"Error: File '{file_path}' not found.")
        sys.exit(1)

    return numbers

## Definición de función
```

```python
def decimal_to_binary(number):
    """
    Convierte un entero decimal a binario utilizando el complemento a
2
    a través de puros algoritmos y funciones básicas.
    """

    # Caso especial: cero
    if number == 0:
        return "0"

    # PASO #1
    # Determina el signo del número
    is_negative = number < 0
    # Se trabaja con el valor absoluto para
    # realizar la conversión binaria
    number = abs(number)

    # PASO #2
    # Conversión decimal → binario (valor absoluto)
    # Por medio de división repetida entre 2
    binary_digits = []
    temp = number

    while temp > 0:
        remainder = temp % 2                # Bit menos significativo
        binary_digits.append(str(remainder))
        temp //= 2                          # División entre 2

    # Inversión de la lista para tener el orden binario correcto
    binary_digits.reverse()
    binary_value = "".join(binary_digits)

    # PASO #3
    # Si es positivo, regresamos el binario directo
    if not is_negative:
        return binary_value

    # PASO #4
    # Complemento a 2 para números negativos
    # Se agrega un bit extra para manejo del signo
    bits = len(binary_value) + 1
    binary_value = binary_value.zfill(bits)

    # PASO #4(A)
    # Se calcula el complemento uno, esta operación
    # consiste en invertir cada bit:
    #  '0' se vuelve '1'
    #  '1' se vuelve '0'
    inverted = []
```

```
    for bit in binary_value:
        # Si el bit actual es '0' se invierte a '1'
        if bit == '0':
            inverted.append('1')
        # Si el bit actual es '1' se inviete a '0'
        else:
            inverted.append('0')

    # PASO #4(B)
    # Se suma 1 para obtener el complemento a 2
    # Esta suma comienza con el bit menos significativo (derecha)
    # y se propaga el acarreo a la izquierda cuando se necesite.
    # Se inicializa el acarreo con 1, debido a que sumamos 1.
    carry = 1
    # Recorremos la lista de bits de derecha a izquierda
    for i in range(len(inverted) - 1, -1, -1):

        # Caso 1:
        # Si rl bit actual es '1' y hay acarreo,
        # el resultado es '0' y el acarreo se mantiene '1'
        if inverted[i] == '1' and carry == 1:
            inverted[i] = '0'
            carry = 1

        # Caso 2:
        # Si el bit actual es '0' y hay acarreo,
        # el resultado es '1' y el acarreo no se presenta
        elif inverted[i] == '0' and carry == 1:
            inverted[i] = '1'
            carry = 0

    # PASO #5
    # se une la lista dentro de la cadena binaria final
    # Lista de caracteres se convierte a caracteres binario
    # representando el complemento a 2.
    return "".join(inverted)

## Definición de función
def decimal_to_binary_and_hexadecimal(number):
    """
    Conversión de decimal a:
    - binarios (utiliza complemento a 2, si es negativo)
    - hexadecimal (de la representación binaria, )

    Regresa: (decimal, binario, hexadecimal)
    """

    hex_symbols = "0123456789ABCDEF"

    # Caso especial
```

```
    if number == 0:
        return 0, "0", "0"

    # Se obtiene el complemento a 2, para obtener el valor decimal en
binario
    binary_value = decimal_to_binary(number)

    # Ajuste de bits para hexadecimal
    # Determina el bit de signo.
    # El bit inicial de `binary_value` será el bit de signo
    # (0 para positivo, 1 para negativo).
    # Necesitamos extender este bit de signo para que la
    # longitud sea un múltiplo de 4.
    sign_bit = binary_value[0] if binary_value else '0'

    while len(binary_value) % 4 != 0:
        binary_value = sign_bit + binary_value

    # Binario a hexadecimal
    hex_digits = []

    for i in range(0, len(binary_value), 4):
        nibble = binary_value[i:i+4]
        value = 0
        for bit in nibble:
            value = value * 2 + int(bit)
        hex_digits.append(hex_symbols[value])

    hex_value = "".join(hex_digits)

    return number, binary_value, hex_value

## Definición de función
def write_results(output_lines_to_write):
    """
    Write the given lines to ConvertionResults.txt.
    """
    with open("ConvertionResults.txt", "w", encoding="utf-8") as file:
        for line in output_lines_to_write:
            file.write(line + "\n")

## Definición de función programa principall
def main():
    """
    Punto inicial del programa
    """
    start_time = time.perf_counter()    #Inicialización de timer

    if len(sys.argv) != 2:
        print("Usage: python convertNumbers.py <input_file>")
        sys.exit(1)
```

```
    file_path = sys.argv[1]
    numbers = read_numbers_from_file(file_path)

    if not numbers:
        print("Error: No valid numeric data found.")
        sys.exit(1)

    output_lines = []
    output_lines.append("Decimal | Binary | Hexadecimal")
    output_lines.append("-" * 40)

    for number in numbers:
        dec, binary_value, hex_value =
decimal_to_binary_and_hexadecimal(number)
        line = f"{dec} | {binary_value} | {hex_value}"
        output_lines.append(line)
        print(line)                          #Impresión de los
resultados

    end_time = time.perf_counter()           #Finalizacion del timer
    elapsed_time = end_time - start_time     #Cálculo de tiempo de
ejecución

    time_line = f"Execution Time (seconds): {elapsed_time}"
    output_lines.append("-" * 40)
    output_lines.append(time_line)

    print(time_line)
    write_results(output_lines)              #Escritura de los
resultados a archivo


if __name__ == "__main__":
    main()

Overwriting convertNumbers.py
```

#Conversión de los datos de los archivos TC2.txt y TC3.txt

```
!python convertNumbers.py TC2.txt

7116776 | 111011001001011111101000 | EC97E8
1666340 | 111110010110110100100100 | F96D24
8886983 | 100001111001101011000111 | 879AC7
839365 | 11001100111011000101 | CCEC5
924280 | 11100001101001111000 | E1A78
1026310 | 11111010100100000110 | FA906
1615293 | 11111000101001011011101 | F8A5BD
1063875 | 11110000001110111000011 | F03BC3
```

679035 | 10100101110001111011 | A5C7B
5201970 | 11001111011000000000110010 | CF6032
593979 | 10010001000000111011 | 9103B
801371 | 11000011101001011011 | C3A5B
3796878 | 11111001111101111110001110 | F9EF8E
7489201 | 11110010010000110110110001 | F246B1
9740020 | 10010100100111101111010100 | 949EF4
9128737 | 10001011010010110010000 1 | 8B4B21
5473463 | 11010011100001001011011 1 | D384B7
8701957 | 10000100110010000000101 | 84C805
8238050 | 11111101101100111110001 0 | FDB3E2
8679038 | 10000100011011100111110 | 846E7E
385912 | 11011110001101111000 | DE378
5867340 | 11011001100001110100 1100 | D9874C
4894542 | 11001010101011110100 1110 | CAAF4E
8999451 | 10001001010100100001 1011 | 89521B
4392535 | 11000011000001100101 0111 | C30657
2078131 | 11111111101101011011 0011 | FFB5B3
3070124 | 11101110110110001010 1100 | EED8AC
7451998 | 11110001101101010101 1110 | F1B55E
5635510 | 11010101111111011011 0110 | D5FDB6
1233932 | 11110010110101000000 1100 | F2D40C
6089867 | 11011100111011001000 1011 | DCEC8B
1792316 | 11111011010110010011 1100 | FB593C
6298637 | 11100000000111000000 1101 | E01C0D
2408038 | 11100100101111100110 0110 | E4BE66
8510100 | 10000001110110101001 0100 | 81DA94
991581 | 11110010000101011101 | F215D
6455739 | 11100010100000011011 1011 | E281BB
7829175 | 11110111011101101011 0111 | F776B7
6328931 | 11100000100100100110 0011 | E09263
9982305 | 10011000010100010110 0001 | 985161
2292022 | 11100010111110010011 0110 | E2F936
1070496 | 11110000010101011010 0000 | F055A0
8518360 | 10000001111110101101 1000 | 81FAD8
6378470 | 11100001010100111110 0110 | E153E6
7756258 | 11110110010110011110 0010 | F659E2
7111767 | 11101100100000100010 1111 | EC8457
2685897 | 11101000111110111100 1001 | E8FBC9
617946 | 10010110110111011010 | 96DDA
246280 | 11111100001000001000 | FC208
6075442 | 11011100101101000011 0010 | DCB432
991323 | 11110010000001011011 | F205B
6754202 | 11100111000011111001 1010 | E70F9A
3103178 | 11101111010110011100 1010 | EF59CA
3228842 | 11110001010001001010 1010 | F144AA
6786623 | 11100111100011100011 1111 | E78E3F
5329085 | 11010001010100001011 1101 | D150BD
2334647 | 11100011100111111011 0111 | E39FB7

7253396 | 111011101010110110010100 | EEAD94
6516953 | 111000110111000011011001 | E370D9
5972725 | 110110110010001011110101 | DB22F5
5828037 | 110110001110110111000101 | D8EDC5
6642265 | 111001010101101001011001 | E55A59
2559699 | 111001110000111011010011 | E70ED3
6597631 | 111001001010101111111111 | E4ABFF
5826202 | 110110001110011010011010 | D8E69A
4642335 | 110001101101011000011111 | C6D61F
1862900 | 111111000110110011110100 | FC6CF4
6039635 | 110111000010100001010011 | DC2853
3358744 | 111100110100000000011000 | F34018
4965249 | 110010111100001110000001 | CBC381
6621475 | 111001010000100100100011 | E50923
3336153 | 111100101110011111011001 | F2E7D9
1044743 | 11111111000100000111 | FF107
6201477 | 110111101010000010000101 | DEA085
6297100 | 111000000001011000001100 | E0160C
885678 | 11011000001110101110 | D83AE
8546891 | 100000100110101001001011 | 826A4B
6683923 | 111001011111110100010011 | E5FD13
7064472 | 111010111100101110011000 | EBCB98
3430021 | 111101000101011010000101 | F45685
9764122 | 100101001111110100011010 | 94FD1A
9958605 | 100101111111010011001101 | 97F4CD
1006022 | 11110101100111000110 | F59C6
5276496 | 110100001000001101010000 | D08350
8876232 | 100001110111000011001000 | 8770C8
3364648 | 111100110101011100101000 | F35728
3539979 | 111101100000010000001011 | F6040B
3003857 | 111011011101010111010001 | EDD5D1
4481634 | 110001000110001001100010 | C46262
9035781 | 100010011110000000000101 | 89E005
3767021 | 111110010111101011101101 | F97AED
4993901 | 110011000011001101101101 | CC336D
8698377 | 100001001011101000001001 | 84BA09
4626501 | 110001101001100001000101 | C69845
1755352 | 111110101100100011011000 | FAC8D8
2755340 | 111010100000101100001100 | EA0B0C
5150616 | 110011101001011110011000 | CE9798
9345169 | 100011101001100010010001 | 8E9891
848948 | 11001111010000110100 | CF434
2708748 | 111010010101010100001100 | E9550C
8372451 | 111111111100000011100011 | FFC0E3
3523015 | 111101011100000111000111 | F5C1C7
2624283 | 111010000000101100011011 | E80B1B
8795348 | 100001100011010011010100 | 8634D4
2318100 | 111000110101111100010100 | E35F14
4750861 | 110010000111111000001101 | C87E0D

3770597 | 111110011000100011100101 | F988E5
1053352 | 111100000001001010101000 | F012A8
833307 | 11001011011100011011 | CB71B
4632526 | 110001101010111111001110 | C6AFCE
2212917 | 111000011100010000110101 | E1C435
5053189 | 110011010001101100000101 | CD1B05
412218 | 11100100101000111010 | E4A3A
4901564 | 110010101100101010111100 | CACABC
5094606 | 110011011011110011001110 | CDBCCE
8070285 | 111110110010010010001101 | FB248D
6847905 | 111010000111110110100001 | E87DA1
4311824 | 110000011100101100010000 | C1CB10
3009421 | 111011011110101110001101 | EDEB8D
1629105 | 111110001101101110110001 | F8DBB1
8431899 | 100000001010100100011011 | 80A91B
7582456 | 111100111011001011111000 | F3B2F8
7332147 | 111011111110000100110011 | EFE133
314183 | 11001100101101000111 | CCB47
7632568 | 111101000111011010111000 | F476B8
8161327 | 111111001000100000101111 | FC882F
6961087 | 111010100011011110111111 | EA37BF
6898282 | 111010010100001001101010 | E9426A
9749303 | 100101001100001100110111 | 94C337
4427419 | 110000111000111010011011 | C38E9B
4368829 | 110000101010100110111101 | C2A9BD
6439318 | 111000100100000110010110 | E24196
6473603 | 111000101100011110000011 | E2C783
6446249 | 111000100101110010101001 | E25CA9
9818836 | 100101011101001011010100 | 95D2D4
9438069 | 100100000000001101110101 | 900375
2809499 | 111010101101111010011011 | EADE9B
3460304 | 111101001100110011010000 | F4CCD0
8430904 | 100000001010010100111000 | 80A538
9774785 | 100101010010011011000001 | 9526C1
9560603 | 100100011110001000011011 | 91E21B
3440732 | 111101001000000001011100 | F4805C
7053263 | 111010111001111111001111 | EB9FCF
287398 | 11000110001010100110 | C62A6
4043644 | 111111011011001101111100 | FDB37C
2943820 | 111011001110101101001100 | ECEB4C
9030962 | 100010011100110100110010 | 89CD32
7537540 | 111100110000001110000100 | F30384
7203657 | 111011011110101101001001 | EDEB49
5639400 | 110101100000110011101000 | D60CE8
9448142 | 100100000010101011001110 | 902ACE
9954392 | 100101111110010001011000 | 97E458
2856859 | 111010111001011110011011 | EB979B
1332118 | 111101000101001110010110 | F45396
6984725 | 111010101001010000010101 | EA9415

```
1925762 | 111111010110001010000010 | FD6282
2138171 | 111000001010000000111011 | E0A03B
347447 | 11010100110100110111 | D4D37
8839390 | 100001101110000011011110 | 86E0DE
3951652 | 111111000100110000100100 | FC4C24
3755102 | 111110010100110001011110 | F94C5E
6867687 | 111010001100101011100111 | E8CAE7
4383175 | 110000101110000111000111 | C2E1C7
7125508 | 111011001011101000000100 | ECBA04
7799732 | 111101110000001110110100 | F703B4
8784156 | 100001100000100100011100 | 86091C
1372809 | 111101001111001010001001 | F4F289
1327984 | 111101000100001101110000 | F44370
9205087 | 100011000111010101011111 | 8C755F
6524241 | 111000111000110101010001 | E38D51
9316072 | 100011100010011011101000 | 8E26E8
127744 | 1111111001100000000 | FF300
8388820 | 100000000000000011010100 | 8000D4
6883557 | 111010010000100011100101 | E908E5
2291239 | 111000101111011000100111 | E2F627
3101270 | 111011110101001001010110 | EF5256
1382825 | 111101010001100110101001 | F519A9
9873376 | 100101101010011111100000 | 96A7E0
6873500 | 111010001110000110011100 | E8E19C
8185634 | 111111001110011100100010 | FCE722
2464786 | 111001011001110000010010 | E59C12
8387154 | 111111111111101001010010 | FFFA52
5581444 | 110101010010101010000100 | D52A84
8097911 | 111110111001000001110111 | FB9077
8882527 | 100001111000100101011111 | 87895F
8941444 | 100010000110111110000100 | 886F84
4942703 | 110010110110101101101111 | CB6B6F
101144 | 11111000101100011000 | F8B18
7471180 | 111100100000000001001100 | F2004C
1932131 | 111111010111101101100011 | FD7B63
8052752 | 111110101110000000010000 | FAE010
6359493 | 111000010000100111000101 | E109C5
1967646 | 111111100000011000011110 | FE061E
6575052 | 111001000101001111001100 | E453CC
2323342 | 111000110111001110001110 | E3738E
6735760 | 111001101100011110010000 | E6C790
8895858 | 100001111011110101110010 | 87BD72
4238091 | 110000001010101100001011 | C0AB0B
7093069 | 111011000011101101001101 | EC3B4D
39 | 11100111 | E7
Execution Time (seconds): 0.005171652000001359

!python convertNumbers.py TC3.txt
```

-39 | 11011001 | D9
-36 | 11011100 | DC
8 | 1000 | 8
34 | 11100010 | E2
17 | 11110001 | F1
49 | 11110001 | F1
5 | 1101 | D
39 | 11100111 | E7
0 | 0 | 0
33 | 11100001 | E1
12 | 1100 | C
-6 | 1010 | A
27 | 11111011 | FB
-4 | 1100 | C
-38 | 11011010 | DA
26 | 11111010 | FA
49 | 11110001 | F1
29 | 11111101 | FD
42 | 11101010 | EA
-16 | 11110000 | F0
-28 | 11100100 | E4
34 | 11100010 | E2
20 | 11110100 | F4
0 | 0 | 0
25 | 11111001 | F9
45 | 11101101 | ED
3 | 1111 | F
-46 | 11010010 | D2
-46 | 11010010 | D2
29 | 11111101 | FD
33 | 11100001 | E1
29 | 11111101 | FD
26 | 11111010 | FA
-5 | 1011 | B
-36 | 11011100 | DC
12 | 1100 | C
45 | 11101101 | ED
-50 | 11001110 | CE
0 | 0 | 0
-6 | 1010 | A
-39 | 11011001 | D9
35 | 11100011 | E3
26 | 11111010 | FA
-35 | 11011101 | DD
-42 | 11010110 | D6
14 | 1110 | E
-3 | 1101 | D
22 | 11110110 | F6
-47 | 11010001 | D1
12 | 1100 | C

```
-17 | 11101111 | EF
-31 | 11100001 | E1
-25 | 11100111 | E7
-35 | 11011101 | DD
-18 | 11101110 | EE
14 | 1110 | E
37 | 11100101 | E5
-8 | 11111000 | F8
-41 | 11010111 | D7
-50 | 11001110 | CE
-9 | 11110111 | F7
2 | 1110 | E
12 | 1100 | C
-39 | 11011001 | D9
-17 | 11101111 | EF
0 | 0 | 0
27 | 11111011 | FB
28 | 11111100 | FC
45 | 11101101 | ED
-44 | 11010100 | D4
28 | 11111100 | FC
35 | 11100011 | E3
-33 | 11011111 | DF
43 | 11101011 | EB
-4 | 1100 | C
23 | 11110111 | F7
-49 | 11001111 | CF
49 | 11110001 | F1
-7 | 1001 | 9
-38 | 11011010 | DA
-45 | 11010011 | D3
5 | 1101 | D
-21 | 11101011 | EB
26 | 11111010 | FA
-2 | 1110 | E
-24 | 11101000 | E8
6 | 1110 | E
-45 | 11010011 | D3
-4 | 1100 | C
15 | 1111 | F
0 | 0 | 0
-17 | 11101111 | EF
-35 | 11011101 | DD
43 | 11101011 | EB
45 | 11101101 | ED
-35 | 11011101 | DD
-8 | 11111000 | F8
28 | 11111100 | FC
47 | 11101111 | EF
34 | 11100010 | E2
```

```
-24 | 11101000 | E8
-42 | 11010110 | D6
-13 | 11110011 | F3
14 | 1110 | E
-45 | 11010011 | D3
-35 | 11011101 | DD
27 | 11111011 | FB
-12 | 11110100 | F4
-19 | 11101101 | ED
-20 | 11101100 | EC
-14 | 11110010 | F2
20 | 11110100 | F4
30 | 11111110 | FE
12 | 1100 | C
-36 | 11011100 | DC
-47 | 11010001 | D1
-27 | 11100101 | E5
-3 | 1101 | D
6 | 1110 | E
36 | 11100100 | E4
9 | 1001 | 9
-16 | 11110000 | F0
36 | 11100100 | E4
-27 | 11100101 | E5
7 | 1111 | F
-6 | 1010 | A
-5 | 1011 | B
5 | 1101 | D
47 | 11101111 | EF
23 | 11110111 | F7
-48 | 11010000 | D0
-24 | 11101000 | E8
40 | 11101000 | E8
-4 | 1100 | C
-18 | 11101110 | EE
-19 | 11101101 | ED
14 | 1110 | E
15 | 1111 | F
1 | 1111 | F
24 | 11111000 | F8
-9 | 11110111 | F7
-50 | 11001110 | CE
18 | 11110010 | F2
15 | 1111 | F
30 | 11111110 | FE
44 | 11101100 | EC
9 | 1001 | 9
19 | 11110011 | F3
33 | 11100001 | E1
41 | 11101001 | E9
```

-24 | 11101000 | E8
43 | 11101011 | EB
48 | 11110000 | F0
-23 | 11101001 | E9
7 | 1111 | F
28 | 11111100 | FC
8 | 1000 | 8
-2 | 1110 | E
37 | 11100101 | E5
42 | 11101010 | EA
-17 | 11101111 | EF
-6 | 1010 | A
36 | 11100100 | E4
22 | 11110110 | F6
27 | 11111011 | FB
31 | 11111111 | FF
-47 | 11010001 | D1
-43 | 11010101 | D5
0 | 0 | 0
27 | 11111011 | FB
30 | 11111110 | FE
12 | 1100 | C
-15 | 11110001 | F1
-19 | 11101101 | ED
-42 | 11010110 | D6
-37 | 11011011 | DB
31 | 11111111 | FF
-10 | 11110110 | F6
18 | 11110010 | F2
33 | 11100001 | E1
-15 | 11110001 | F1
17 | 11110001 | F1
25 | 11111001 | F9
-25 | 11100111 | E7
-5 | 1011 | B
33 | 11100001 | E1
-13 | 11110011 | F3
33 | 11100001 | E1
-10 | 11110110 | F6
47 | 11101111 | EF
47 | 11101111 | EF
-13 | 11110011 | F3
-32 | 11100000 | E0
1 | 1111 | F
1 | 1111 | F
-25 | 11100111 | E7
-33 | 11011111 | DF
16 | 11110000 | F0
17 | 11110001 | F1

```
4 | 1100 | C
Execution Time (seconds): 0.003457742000023245
```

#Análisis del codigo mejorado

```
!pylint convertNumbers.py

************* Module convertNumbers
convertNumbers.py:1:0: C0103: Module name "convertNumbers" doesn't
conform to snake_case naming style (invalid-name)

-----------------------------------------------------------------
Your code has been rated at 9.90/10 (previous run: 9.90/10, +0.00)
```

#Conclusiones

El código desarrollado para la conversión de números decimales a sus respectivas representaciones en codigo binario y hexadecimal es todo un reto sobre todo por el manejo del complemento a 2, que nos permite la representación de números con signo.

Para la implementación solamente se utilizaron algoritmos básicos como; divisiones sucesivas, inversión de bits y suma con acarreo, el código evita funciones de alto nnivel como lo mencionan lo requisitos, lo que nos permite comprender y evidencias las diferentes operaciones que se involucran en el proceso de conversión.

El manejo de los números negativos que se encuentran en los archivos de datos se llevo a cabo mediante el complemento a 1 y el complemento a 2, tratando de garanizar la coherencia de los códigos binarios y hexadecimal. La extensión del bit de signo permite asegurar el valor numérico que representa.

Desde el punto de vista del diseño de software, el código muestra una estructura en módulos, nos enfocamos en que sea legible y documentado, con el objetivo de tratar de cumplir con la norma PEP8, con el objetivo de facilitar su mantenimiento, evaluación y reutilización.

El flujo en el proceso de conversión de decimal a binario y a hexadecimal refleja los procesos internos.

Los requisitos que se mencionan en el ejercicio son los siguientes:

1. El programa será invocado desde la línea de comandos. El programa recibirá un archivo como parámetro. El archivo contendrá una lista de elementos (presumiblemente números). **Se cumple con el requisito**
2. El programa convertirá los números a binario y hexadecimal. Los resultados se mostrarán en pantalla y en un archivo llamado ConvertionResults. **Se cumple con el requisito**
3. El programa incluira un mecanismo para gestionar datos no válidos en el archivo. Los errores se mostrarán en consola y la ejecución continuará. **Se cumple con el requisito**
4. El nombre del programa será convertNumbers.py. **Se cumple con el requisito**

5. El formato mínimo para invocar el programa será el suguiente python convertNumbers.py fileWithdata.txt. **Se cumple con el requsito**
6. El programa gestionará archivos con un número de elementos que puede variar desde cientos hasta miles. **Se cumple con el requisito**
7. El programa debe incluir al final de la ejecución el tiempo transcurrido para la ejecución y el cálculo de los datos. **Se cumple con el requisito**
8. Cumplir con la norma PEP8. **Se cumple con el requisito**

Nota, correct pylint en el código sin mejoras nos envia el siguiente mensaje **"Module name "convertNumbers" doesn't conform to snake_case naming style (invalid-name)"**, al correr el código mejorado, el mensaje se mantiene, sin embargo decidimos dejar el nombre del programa como se menciona en el requisito número 4.

Los tiempos de ejecucion de los diferentes archivos los mostramos a continuación:

- TC1.txt ----->> 0.005078 segundos
- TC2.txt ----->> 0.005159 segundos
- TC3.txt ----->> 0.002285 segundos
- TC4.txt ----->> 0.000850 segundos

La calificación el primer código fue de 9.83/10, mientras que el código mejorado presenta una calificación de 9.90/10. La mejorá para está ocasión fue mínima pero se dejaron de utilizar variables locales.