

## Report

### Decimal to Binary and Binary to Decimal Conversion

This report provides an overview of a Python program that serves as a versatile decimal-to-binary and binary-to-decimal converter. The program allows users to interactively enter decimal numbers to convert them to binary and vice versa. The user can convert back and forth until they choose to quit the program.

- **decimal\_to\_binary:** This function takes a decimal number as input and converts it to its binary representation. It utilizes the `bin()` function to perform the conversion and removes the `'0b'` prefix from the result.
- **binary\_to\_decimal:** This function converts a binary string to its decimal representation. It uses the `int()` function with base 2 to perform the conversion and handles any `ValueError` exceptions that may occur if the input is not a valid binary string.

The core functionality of the program is encapsulated within the `convert_decimal_to_binary` function, which handles the interactive conversion process.

- **User Input:** The program uses an infinite loop to repeatedly prompt the user for input. The user can enter either a decimal number to convert to binary or a binary number to convert to decimal. They can also choose to quit the program by entering 'q.'
- **Decimal to Binary:** If the user enters a decimal number, the program attempts to convert it to an integer. It checks if the input is non-negative and converts it to binary using the `decimal_to_binary` function. The binary representation is displayed.
- **Binary to Decimal:** If the user enters a binary number, the program checks if it consists of only '0' and '1' characters. If it's a valid binary string, the program converts it to a decimal number using the `binary_to_decimal` function. The decimal representation is displayed.
- **Input Validation:** The program provides robust input validation, handling various scenarios, such as invalid inputs or negative decimal numbers.

This program is a valuable tool for various applications, including:

- Educational purposes, teaching the binary numbering system and conversions.
- Quick conversions between decimal and binary for programmers and students.
- Debugging and verification of binary data in computer science and engineering.

In conclusion, the Python program presented in this report offers a user-friendly and versatile decimal-to-binary and binary-to-decimal converter. Its interactive nature allows users to explore conversions conveniently and learn about the relationships between decimal and binary representations. With input validation and a user-friendly interface, it provides a practical tool for a wide range of users interested in binary conversions.

```

Module-2 > Lab-3,4 > Task-4 > binary_conversion.py > print_system_info
1  ## Student Name: Harsh Siddhapura
2  ## Student ID: 1230169813
3  ## Date: 09/01/2023
4
5  import os
6  import datetime
7
8  def print_system_info():
9      # Get user data
10     os.system('clear') # os.system('clear') for Linux
11     username = os.getlogin()
12     # Get computer information
13     computer_info = os.name
14     # Get current date and time
15     current_time = datetime.datetime.now()
16     # Format log message
17     log_message = f"User: {username}\nTime:{current_time}\nComputer Info: {computer_info}"
18     # Print log message
19     print(log_message)
20     # Call the function to print the log
21     print_system_info()
22
23     def decimal_to_binary(decimal_num):
24         # Convert a decimal number to its binary representation
25         binary_repr = bin(decimal_num)[2:] # Remove '0b' prefix
26         return binary_repr
27
28     def convert_decimal_to_binary():
29         # Convert decimal numbers to binary
30         while True:
31             user_input = input("Enter a decimal number (q to quit): ")
32             if user_input.lower() == 'q':
33                 break # Exit the loop if 'q' is entered
34
35             try:
36                 decimal_num = int(user_input) # Attempt to convert input to integer
37                 binary_repr = decimal_to_binary(decimal_num) # Convert to binary
38                 print(f"Binary representation: {binary_repr}")
39             except ValueError:
40                 print("Invalid input. Please enter a valid decimal number or 'q' to quit.")
41
42     convert_decimal_to_binary()

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL COMMENTS

zsh - Task-4

```

User: root
Time:2023-09-01 21:42:34.370533
Computer Info: posix
Enter a decimal number (q to quit): 8
Binary representation: 1000
Enter a decimal number (q to quit): 5
Binary representation: 101
Enter a decimal number (q to quit): 12
Binary representation: 1100
Enter a decimal number (q to quit): 25
Binary representation: 11001
Enter a decimal number (q to quit): 56
Binary representation: 111000
Enter a decimal number (q to quit): q

```

main\* 0 0 0 tabnine starter

Ln 17, Col 35 Spaces: 4 UTF-8 LF Python 3.9.6 ('.venv': venv) Prettier

## Challenge Code (Optional)

The image shows a VS Code editor window with a file explorer on the left and a terminal at the bottom. The file explorer shows a project structure under 'PYTHON LABS' with folders for 'Module-1', 'Module-2', 'Lab-1', 'Lab-2', 'Lab-3,4', and 'Task-1' through 'Task-4'. The 'Task-4' folder is selected, showing files 'binary\_conversion\_s...', 'binary\_conversion.py', and 'challenge\_binary\_co...'. The main editor displays the file 'challenge\_binary\_conversion.py' with the following code:

```

23
24 def decimal_to_binary(decimal_num):
25     # Convert a decimal number to its binary representation
26     binary_repr = bin(decimal_num)[2:] # Convert to binary and remove '0b' prefix
27     return binary_repr
28
29 def binary_to_decimal(binary_str):
30     # Convert a binary string to its decimal representation
31     try:
32         decimal_num = int(binary_str, 2)
33         return decimal_num
34     except ValueError:
35         return None
36
37 def convert_decimal_to_binary():
38     # Convert decimal numbers to binary and vice versa
39     while True:
40         decimal_user_input = input("Enter a decimal number to convert to binary (q to quit): ")
41         if decimal_user_input.lower() == 'q': break
42
43         binary_user_input = input("Enter a binary number to convert to decimal (q to quit): ")
44         if binary_user_input.lower() == 'q': break
45
46         try:
47             decimal_num = int(decimal_user_input) # Attempt to convert input to integer
48             if decimal_num >= 0:
49                 binary_repr = decimal_to_binary(decimal_num) # Convert to binary
50                 print(f"Binary representation: {binary_repr}")
51                 if all(bit in '01' for bit in binary_user_input): # Check if input is a binary number
52                     decimal_num = binary_to_decimal(binary_user_input)
53                     if decimal_num is not None:
54                         print(f"Decimal representation: {decimal_num}")
55                 else:
56                     print("Invalid binary input. Please enter a valid binary number or 'q' to quit.")
57             else:
58                 print("Negative decimal numbers are not supported. Please enter a non-negative decimal number.")
59         except ValueError:
60             print("Invalid input. Please enter a valid decimal, or 'q' to quit.")
61
62 # Start the program by calling convert_decimal_to_binary
63 convert_decimal_to_binary()

```

The terminal output shows the execution of the script:

```

User: harshsiddhapura
Time: 2023-09-01 22:06:52.992530
Computer Info: posix
Enter a decimal number to convert to binary (q to quit): 8
Enter a binary number to convert to decimal (q to quit): 10001
Binary representation: 1000
Decimal representation: 17
Enter a decimal number to convert to binary (q to quit): -8
Enter a binary number to convert to decimal (q to quit): 101
Negative decimal numbers are not supported. Please enter a non-negative decimal number.
Enter a decimal number to convert to binary (q to quit): K
Enter a binary number to convert to decimal (q to quit): 10101
Invalid input. Please enter a valid decimal or binary number, or 'q' to quit.
Enter a decimal number to convert to binary (q to quit): q
o (.venv) harshsiddhapura@Harshs-MacBook-Air Task-4 %

```

The status bar at the bottom indicates the file is 'main\*' in the 'tabnine starter' project, using 'UTF-8' encoding, 'LF' line endings, 'Python 3.9.6' interpreter, and 'Prettier' formatter.