Name - Jaspreet singh
Submitted to - prof.alex yang
Subject - computer Organisation
2023 fall homework #1

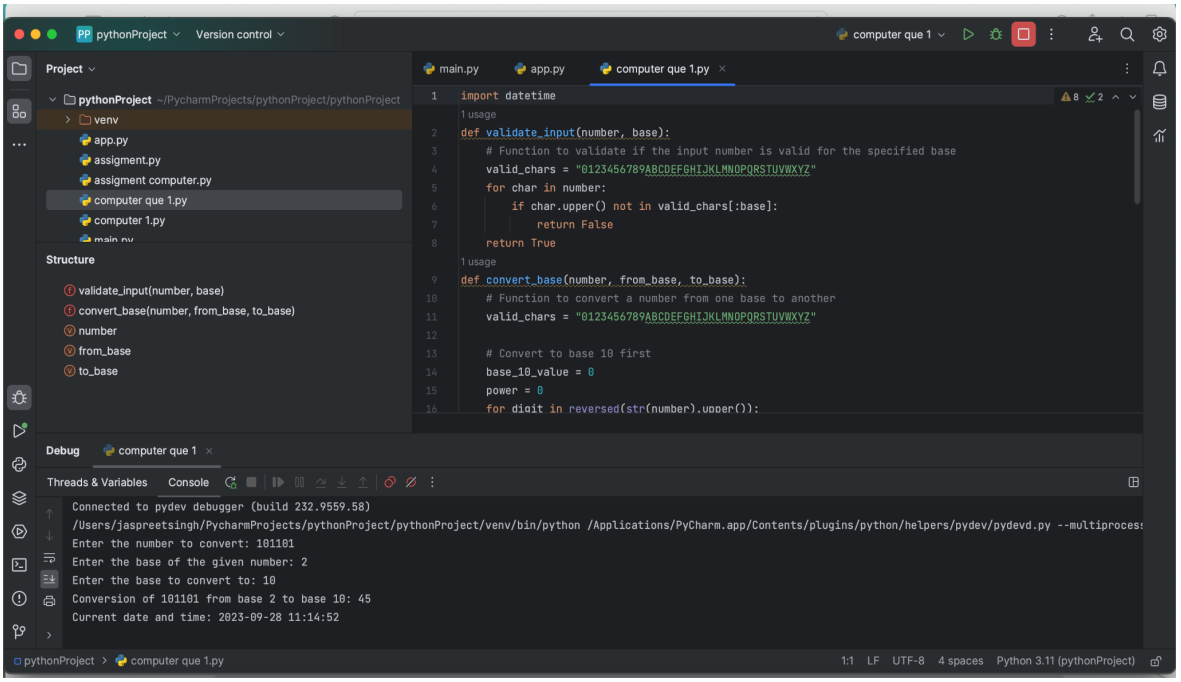
QUE 1. Write the program in any computer language to convert the given number from any base to a different base. The program needs to verify the validity of the given number first. If it is invalid, please prompt error information. Otherwise, print the correct result in the new base. For instance, as follows is the def function "base_conv" in Python.

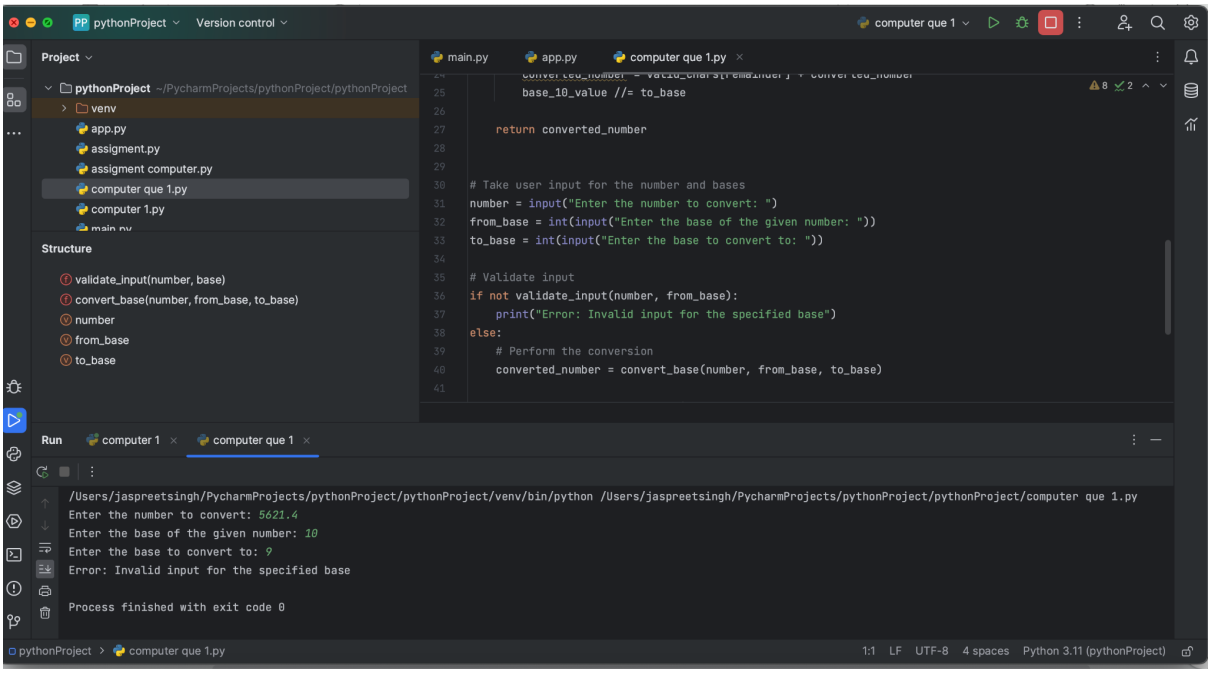
CODE:

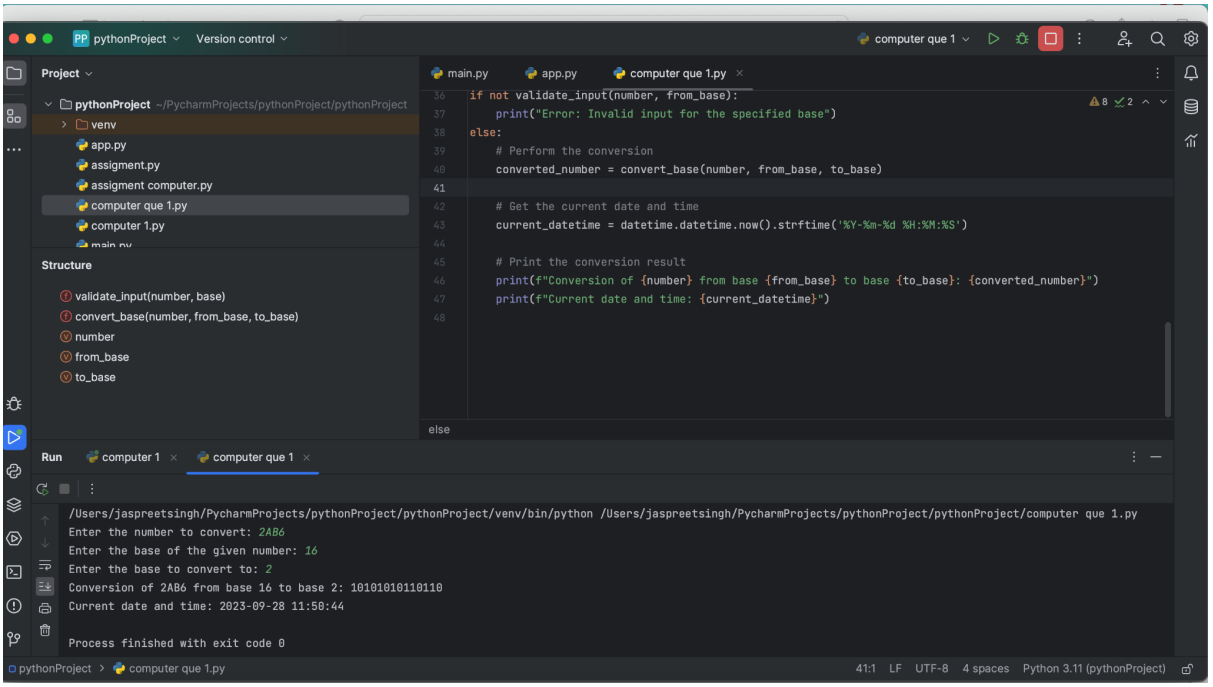
```
import datetime
def validate_input(number, base):
    # Function to validate if the input number is valid for the specified base
    valid_chars = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    for char in number:
        if char.upper() not in valid_chars[:base]:
            return False
    return True
def convert_base(number, from_base, to_base):
    # Function to convert a number from one base to another
    valid_chars = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ"

    # Convert to base 10 first
    base_10_value = 0
    power = 0
    for digit in reversed(str(number).upper()):
        base_10_value += valid_chars.index(digit) * (from_base ** power)
        power += 1

    # Convert to the desired base
    converted_number = ''
    while base_10_value:
        remainder = base_10_value % to_base
        converted_number = valid_chars[remainder] + converted_number
        base_10_value //= to_base

    return converted_number


# Take user input for the number and bases
number = input("Enter the number to convert: ")
from_base = int(input("Enter the base of the given number: "))
to_base = int(input("Enter the base to convert to: "))

# Validate input
if not validate_input(number, from_base):
    print("Error: Invalid input for the specified base")
else:
```

```
    # Perform the conversion
    converted_number = convert_base(number, from_base, to_base)

    # Get the current date and time
    current_datetime = datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S')

    # Print the conversion result
    print(f"Conversion of {number} from base {from_base} to base {to_base}:
{converted_number}")
    print(f"Current date and time: {current_datetime}")
```

OUTPUT:







GITHUB REPOSITORY LINK:

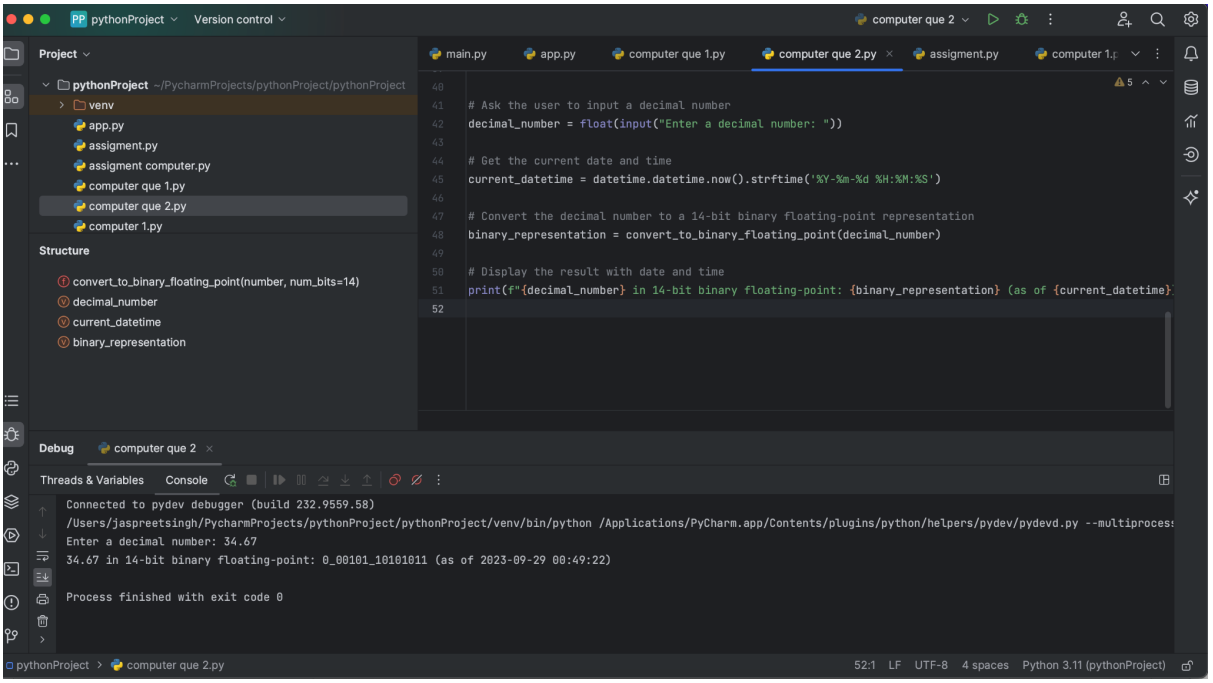
QUE 2. Write the program in any computer language to convert the floating decimal number to 14-bits binary floating-point model as the real digital values in the hardware memory.

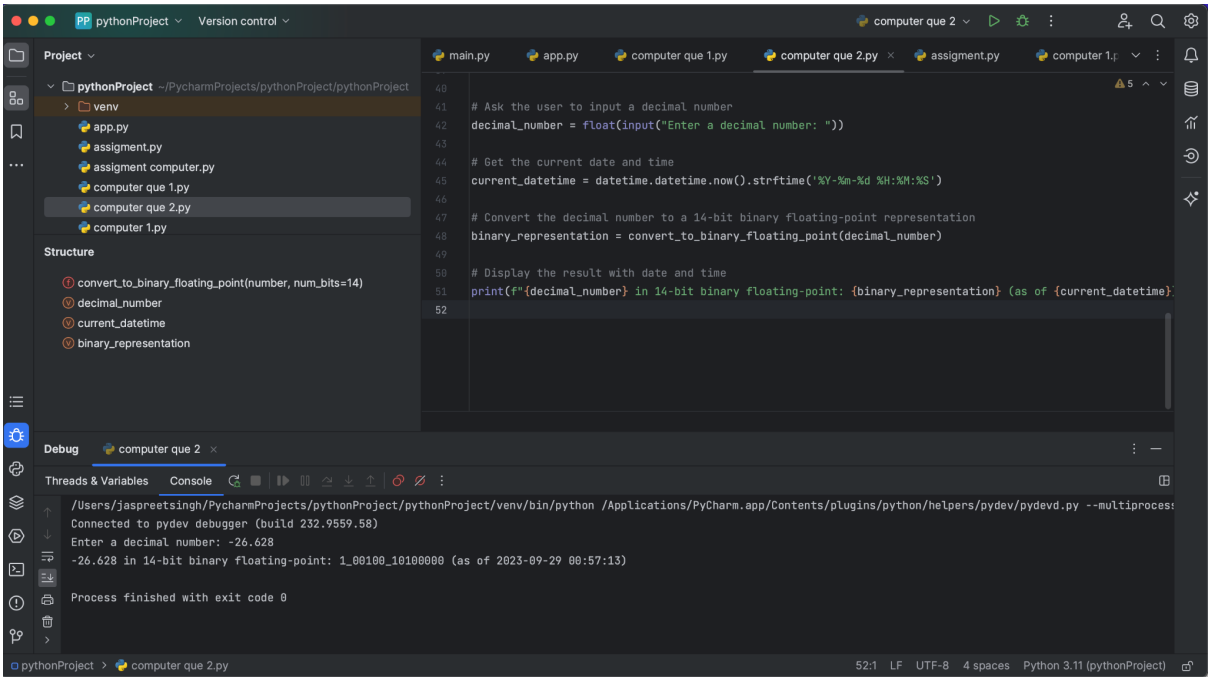
CODE:

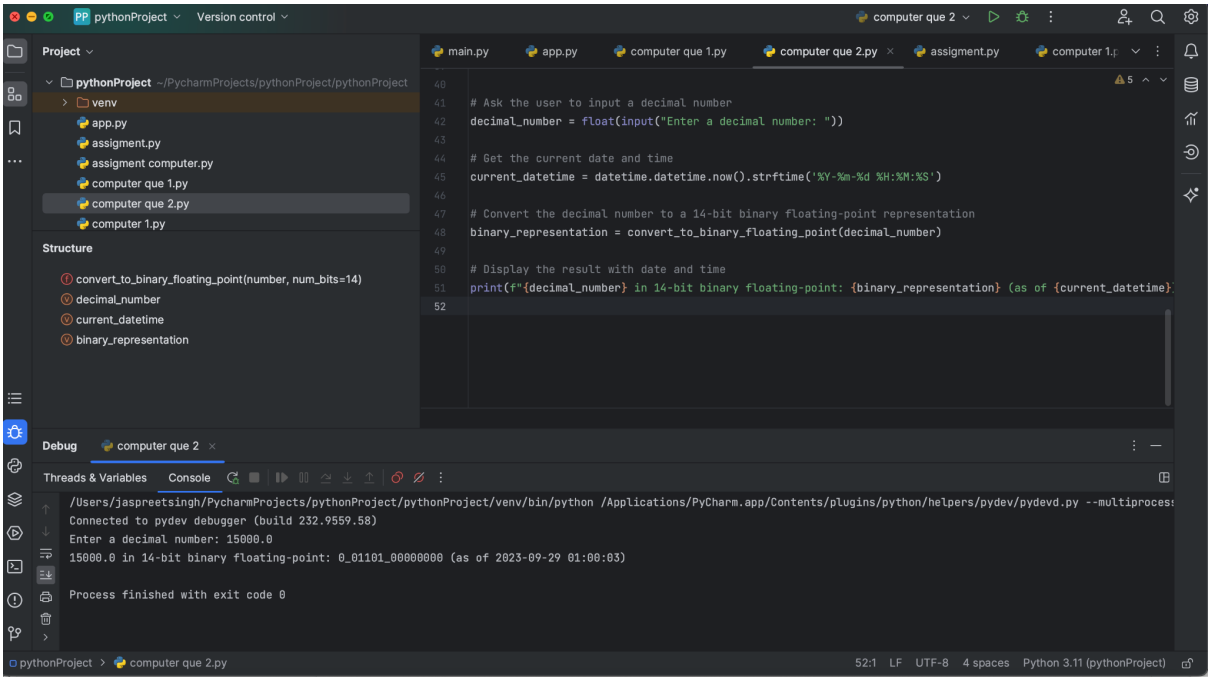
```
import datetime

def convert_to_binary_floating_point(number, num_bits=14):
    # Step 1: Determine the sign bit
    sign_bit = '0' if number >= 0 else '1'
    number = abs(number)

    # Step 2: Convert the integer part and fractional part to binary
    integer_part = bin(int(number))[2:]
```

```
    fractional_part = ''
    fractional_value = number - int(number)

    while len(fractional_part) < 8:  # Limit the mantissa to 8 bits
        fractional_value *= 2
        bit = '1' if fractional_value >= 1.0 else '0'
        fractional_part += bit
        if fractional_value >= 1.0:
            fractional_value -= 1.0

    # Trim trailing zeros from the fractional part
    fractional_part = fractional_part.rstrip('0')

    # Step 3: Adjust the binary point
    binary_value = integer_part + '.' + fractional_part

    # Step 4: Adjust the exponent
    exponent = len(integer_part) - 1  # Adjust for the implicit leading '1'

    # Step 5: Combine the sign bit, exponent, and mantissa
    mantissa = fractional_part[:8]  # Limit the mantissa to 8 bits

    # Ensure the mantissa is of the desired length
    mantissa = mantissa.ljust(8, '0')

    # Format the final binary representation
    binary_representation = f"{sign_bit}_{exponent:05b}_{mantissa}"

    return binary_representation


# Ask the user to input a decimal number
decimal_number = float(input("Enter a decimal number: "))

# Get the current date and time
current_datetime = datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S')

# Convert the decimal number to a 14-bit binary floating-point representation
binary_representation = convert_to_binary_floating_point(decimal_number)

# Display the result with date and time
print(f"{decimal_number} in 14-bit binary floating-point:
{binary_representation} (as of {current_datetime})")
```

OUTPUT


pythonProject
Version control
computer que 2
Project
pythonProject ~/PycharmProjects/pythonProject/pythonProject
venv
app.py
assigment.py
assigment computer.py
computer que 1.py
computer que 2.py
computer 1.py
Structure
convert_to_binary_floating_point(number, num_bits=14)
decimal_number
current_datetime
binary_representation
main.py
app.py
computer que 1.py
computer que 2.py
assigment.py
computer 1.
# Ask the user to input a decimal number
decimal_number = float(input("Enter a decimal number: "))
# Get the current date and time
current_datetime = datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S')
# Convert the decimal number to a 14-bit binary floating-point representation
binary_representation = convert_to_binary_floating_point(decimal_number)
# Display the result with date and time
print(f"{decimal_number} in 14-bit binary floating-point: {binary_representation} (as of {current_datetime}
Debug
computer que 2
Threads & Variables
Console
Connected to pydev debugger (build 232.9559.58)
/Users/jaspreetsingh/PycharmProjects/pythonProject/pythonProject/venv/bin/python /Applications/PyCharm.app/Contents/plugins/python/helpers/pydev/pydevd.py --multiproces
Enter a decimal number: 34.67
34.67 in 14-bit binary floating-point: 0_00101_10101011 (as of 2023-09-29 00:49:22)
Process finished with exit code 0
pythonProject
computer que 2.py
52:1
LF
UTF-8
4 spaces
Python 3.11 (pythonProject)


pythonProject
Version control
computer que 2
Project
pythonProject ~/PycharmProjects/pythonProject/pythonProject
venv
app.py
assigment.py
assigment computer.py
computer que 1.py
computer que 2.py
computer 1.py
Structure
convert_to_binary_floating_point(number, num_bits=14)
decimal_number
current_datetime
binary_representation
main.py
app.py
computer que 1.py
computer que 2.py
assigment.py
computer 1.
# Ask the user to input a decimal number
decimal_number = float(input("Enter a decimal number: "))
# Get the current date and time
current_datetime = datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S')
# Convert the decimal number to a 14-bit binary floating-point representation
binary_representation = convert_to_binary_floating_point(decimal_number)
# Display the result with date and time
print(f"{decimal_number} in 14-bit binary floating-point: {binary_representation} (as of {current_datetime}
Debug
computer que 2
Threads & Variables
Console
/Users/jaspreetsingh/PycharmProjects/pythonProject/pythonProject/venv/bin/python /Applications/PyCharm.app/Contents/plugins/python/helpers/pydev/pydevd.py --multiproces
Connected to pydev debugger (build 232.9559.58)
Enter a decimal number: -26.628
-26.628 in 14-bit binary floating-point: 1_00100_10100000 (as of 2023-09-29 00:57:13)
Process finished with exit code 0
pythonProject
computer que 2.py
52:1
LF
UTF-8
4 spaces
Python 3.11 (pythonProject)



GITHUB REPOSITORY LINK: