Batch: B2 Roll No.:16010124107

Experiment / assignment / tutorial No.5

TITLE: Implementation of IEEE-754 floating point representation

AIM: To demonstrate the single and double precision formats to represent floating point numbers.

Expected OUTCOME of Experiment: (Mention CO attained here)

Books/ Journals/ Websites referred:

1. Carl Hamacher, Zvonko Vranesic and Safwat Zaky, “Computer Organization”, Fifth Edition, TataMcGraw-Hill.

2. William Stallings, “Computer Organization and Architecture: Designing for Performance”, Eighth Edition, Pearson.

Pre Lab/ Prior Concepts:

The IEEE Standard for Floating-Point Arithmetic (IEEE 754) is a technical standard for floating-point computation established in 1985 by the Institute of Electrical and Electronics Engineers (IEEE). The standard addressed many problems found in the diverse floating point implementations that made them difficult to use reliably and portably. Many hardware floating point units now use the IEEE 754 standard.

The standard defines:

- *arithmetic formats:* sets of binary and decimal floating-point data, which consist of finite numbers (including signed zeros and subnormal numbers), infinities, and special "not a number" values (NaNs)

- *interchange formats:* encodings (bit strings) that may be used to exchange floating-point data in an efficient and compact form

- *rounding rules:* properties to be satisfied when rounding numbers during arithmetic and conversions

- *operations:* arithmetic and other operations (such as trigonometric functions) on arithmetic formats

- *exception handling:* indications of exceptional conditions (such as division by zero, overflow, *etc*

Algorithm Steps:

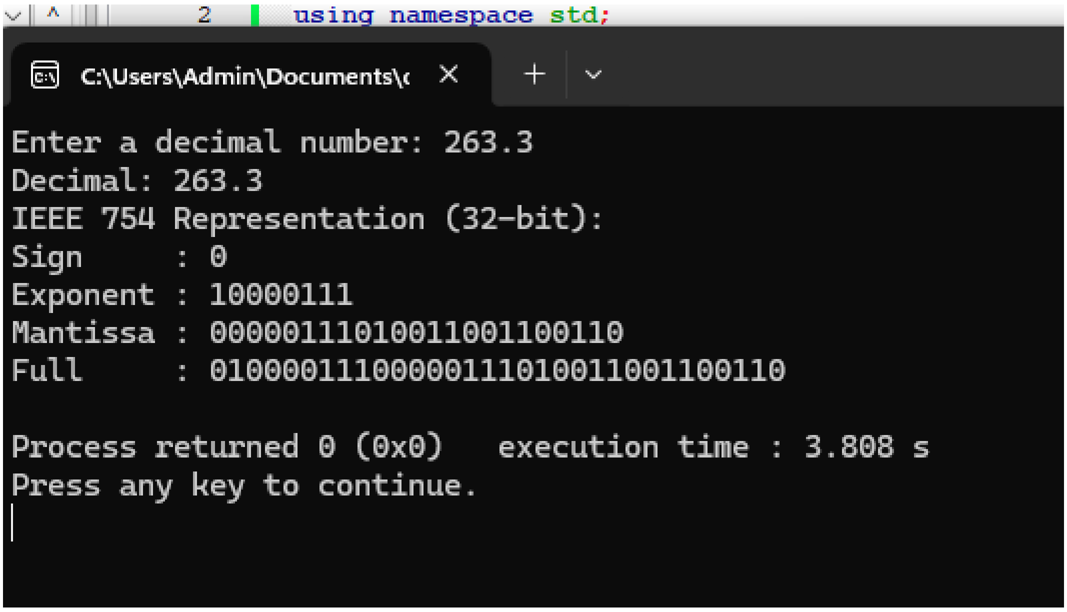
1. start
2. input a number
3. create a union with two data types: float and uint32_t
4. convert the uint32_t part to binary using bitset<32>
5. separate the binary part to signed, mantissa, and exponent part using substr of strings
6. Display as required
7. End

CODE:

```
#include <bits/stdc++.h>
using namespace std;

void doubleToIEEE754(double num) {
    union {
        double input;
        uint64_t bits;
    } data;

    data.input = num;

    bitset<64> binary(data.bits);
    string sign     = binary.to_string().substr(0, 1);
    string exponent = binary.to_string().substr(1, 11);
    string mantissa = binary.to_string().substr(12);

    cout << "Decimal: " << num << endl;
    cout << "IEEE 754 Representation (64-bit):" << endl;
    cout << "Sign     : " << sign << endl;
    cout << "Exponent : " << exponent << endl;
    cout << "Mantissa : " << mantissa << endl;
    cout << "Full     : " << binary << endl;
}

int main() {
    double num;

    cout << "Enter a decimal number: ";
    cin >> num;

    doubleToIEEE754(num);

    return 0;
}
```

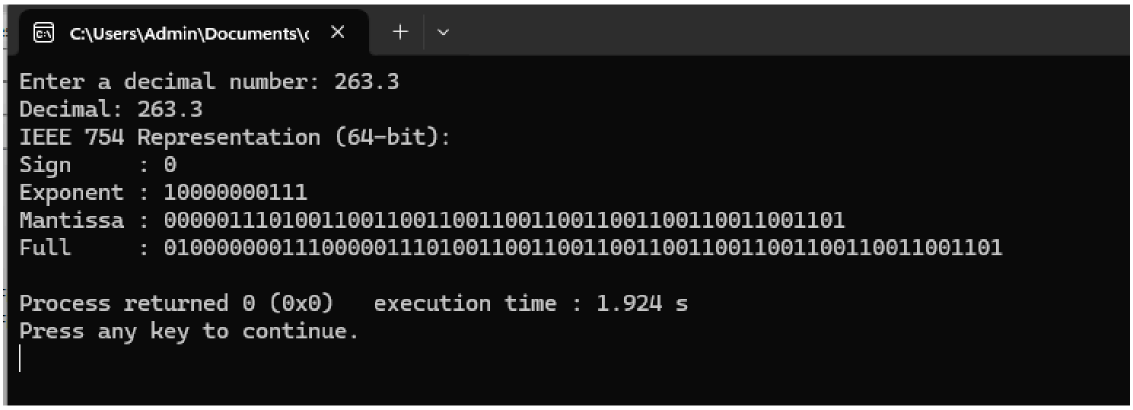
OUTPUT:

FOR 32 BIT REPRESENTATION:-



```
Enter a decimal number: 263.3
Decimal: 263.3
IEEE 754 Representation (32-bit):
Sign     : 0
Exponent : 10000111
Mantissa : 00000111010011001100110
Full     : 01000011100000111010011001100110

Process returned 0 (0x0)   execution time : 3.808 s
Press any key to continue.
```

FOR 64 BIT REPRESENTATION:-



```
Enter a decimal number: 263.3
Decimal: 263.3
IEEE 754 Representation (64-bit):
Sign     : 0
Exponent : 10000000111
Mantissa : 0000011101001100110011001100110011001100110011001101
Full     : 0100000001110000011101001100110011001100110011001100110011001101

Process returned 0 (0x0)   execution time : 1.924 s
Press any key to continue.
```

Post Lab Descriptive Questions

Q. Give the importance of IEEE-754 representation for floating point numbers?
The IEEE 754 standard for floating-point numbers is crucial because it provides a universal, consistent, and portable way to represent and compute real numbers on computers, solving the problem of incompatible floating-point implementations that existed previously.

Q. Draw the IEEE 754 Floating Point representation format.

Single:

Sign(1 bit)	**Exponent(8 bits)**	**Mantissa(23 bits)**
0	**00000000**	**0000000000000**

Double:

Sign (1 bit)	**Exponent(11 bits)**	**Mantissa (52 bits)**
0	**00000000000**	**000000000000000000000000000000000000000000000000000**

Conclusion:-

Implementing the IEEE-754 floating-point standard involves creating hardware or software that correctly represents numbers using binary formats like single-precision (32-bit) or double-precision (64-bit), with a sign bit, a biased exponent, and a significand (mantissa).

Date:
22/08/2025