Single Precision Representation

EXP NO: 35

AIM:

To write a C program to implement IEEE 754 single-precision floating-point representation for a given floating-point number.

ALGORITHM:

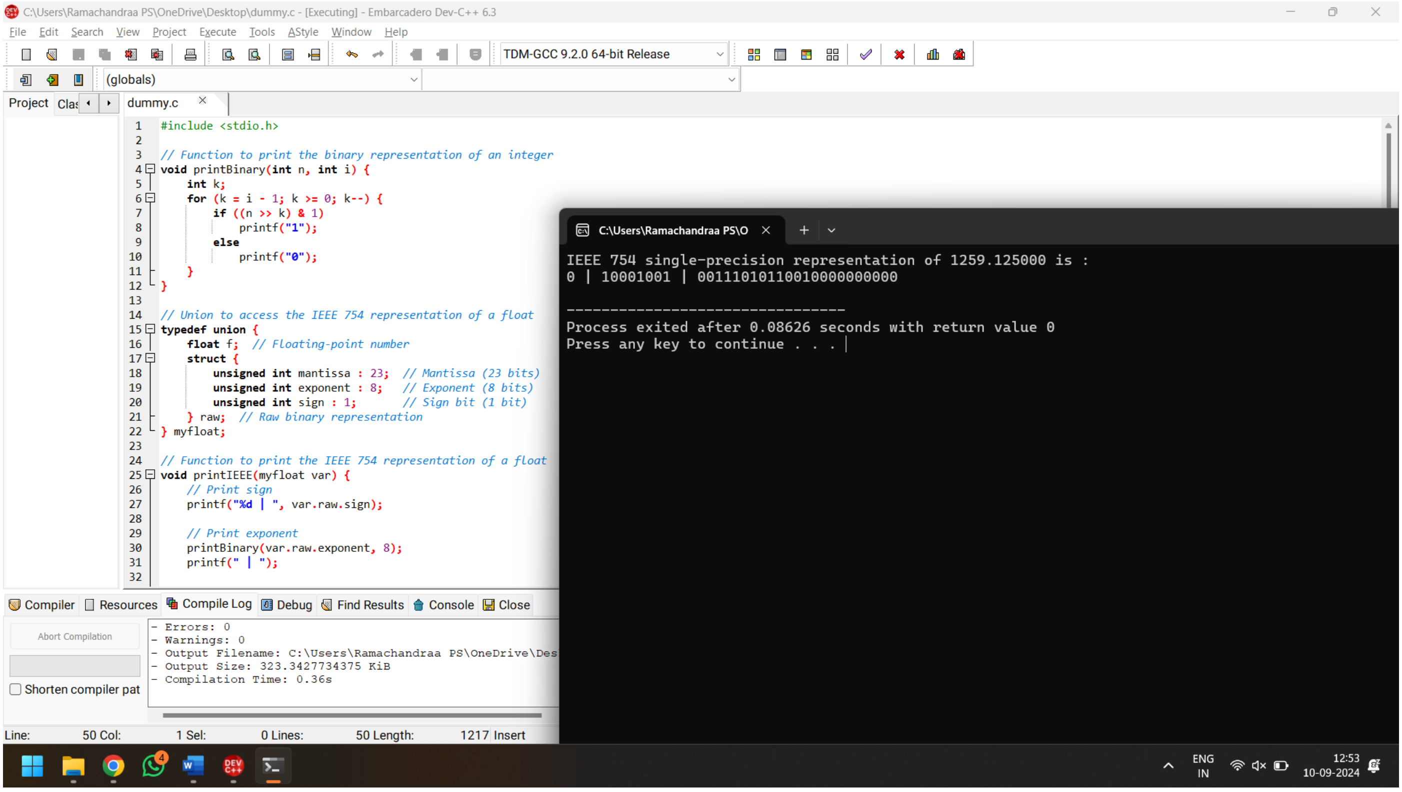
1. **Start**
2. **Define a union** to access the binary representation of a float.
   - The union includes:
     - A float variable f
     - A structure with three fields:
       - mantissa (23 bits)
       - exponent (8 bits)
       - sign (1 bit)
3. **Define a function** printBinary to print the binary representation of an integer.
4. **Define a function** printIEEE to print the IEEE 754 single-precision representation of a float using the union.
   - Print the sign bit.
   - Print the exponent in binary.
   - Print the mantissa in binary.
5. **In the main function**:
   - Initialize the union with a floating-point value.
   - Print the IEEE 754 representation of the float using the printIEEE function.
6. **End**

PROGRAM:

```
#include <stdio.h>


// Function to print the binary representation of an integer
void printBinary(int n, int i) {
  int k;
  for (k = i - 1; k >= 0; k--) {
    if ((n >> k) & 1)
      printf("1");
    else
```

```
            printf("0");
    }
}

// Union to access the IEEE 754 representation of a float
typedef union {
    float f;  // Floating-point number
    struct {
        unsigned int mantissa : 23;  // Mantissa (23 bits)
        unsigned int exponent : 8;   // Exponent (8 bits)
        unsigned int sign : 1;       // Sign bit (1 bit)
    } raw;  // Raw binary representation
} myfloat;

// Function to print the IEEE 754 representation of a float
void printIEEE(myfloat var) {
    // Print sign
    printf("%d | ", var.raw.sign);

    // Print exponent
    printBinary(var.raw.exponent, 8);
    printf(" | ");

    // Print mantissa
    printBinary(var.raw.mantissa, 23);
    printf("\n");
}

int main() {
    myfloat var;

    // Initialize the float variable
    var.f = 1259.125;
    // Print the IEEE 754 representation
    printf("IEEE 754 single-precision representation of %f is : \n", var.f);
    printIEEE(var);


    return 0;
}
```

INPUT & OUTPUT:



C:\Users\Ramachandraa PS\OneDrive\Desktop\dummy.c - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 64-bit Release
(globals)
Project Clas
dummy.c

```c
#include <stdio.h>

// Function to print the binary representation of an integer
void printBinary(int n, int i) {
    int k;
    for (k = i - 1; k >= 0; k--) {
        if ((n >> k) & 1)
            printf("1");
        else
            printf("0");
    }
}

// Union to access the IEEE 754 representation of a float
typedef union {
    float f;  // Floating-point number
    struct {
        unsigned int mantissa : 23;  // Mantissa (23 bits)
        unsigned int exponent : 8;   // Exponent (8 bits)
        unsigned int sign : 1;       // Sign bit (1 bit)
    } raw;  // Raw binary representation
} myfloat;

// Function to print the IEEE 754 representation of a float
void printIEEE(myfloat var) {
    // Print sign
    printf("%d | ", var.raw.sign);

    // Print exponent
    printBinary(var.raw.exponent, 8);
    printf(" | ");

```

```
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Ramachandraa PS\OneDrive\Des
- Output Size: 323.3427734375 KiB
- Compilation Time: 0.36s
```

☐ Shorten compiler pat

Line: 50 Col: 1 Sel: 0 Lines: 50 Length: 1217 Insert

```
IEEE 754 single-precision representation of 1259.125000 is :
0 | 10001001 | 00111010110010000000000

--------------------------------
Process exited after 0.08626 seconds with return value 0
Press any key to continue . . .
```

RESULT: Thus, the program was executed successfully using DevC++