Single Precision Representation :

**EXP NO:35**

# AIM: To write a C program to implement Single Precision Representation

**APPARATUS:** DEV C++

**ALGORITHM:**

he specifier for printing and scanning float values is the same **%f**.  
We can specify the precision of the output. To do this, we write:

*[integer part][.][fraction part]*

* The integer part is optional. Usually it will be ignored.
* The decimal delimiter ‘.’ must be used if we intend to print from the fraction.
* It specifies how many of digits from the fraction should be printed. When the actual value has longer precision, it will be rounded for the printing purposes.

In the next example (**%.2f**) we will print the value with precision two digits after the decimal mark.

**PROGRAM:**

#include <stdio.h>  
void printBinary(int n, int i)  
{  
 int k;  
 for (k = i - 1; k >= 0; k--) {   
if ((n >> k) & 1)    
printf("1");  
else    
printf("0");  
 }  
  
  
}  
  
  
typedef union {  
  
  
 float f;  
  
  
 struct  
  
  
 {  
  
  
   
unsigned int mantissa : 23;  
  
  
   
unsigned int exponent : 8;  
  
  
   
unsigned int sign : 1;  
  
  
 } raw;  
  
  
} myfloat;  
  
  
void printIEEE(myfloat var)  
  
  
{  
  
  
 printf("%d | ", var.raw.sign);  
  
  
 printBinary(var.raw.exponent, 8);  
  
  
 printf(" | ");  
  
  
 printBinary(var.raw.mantissa, 23);  
  
  
 printf("\n");  
  
  
}  
  
  
int main()  
  
  
{  
  
  
myfloat var;  
  
  
var.f = 1259.125;  
  
  
printf("IEEE 754  
representation of %f is : \n",  
  
  
   
var.f);  
  
  
 printIEEE(var);  
  
  
 return 0;  
  
  
}

**INPUT:**

**OUTPUT:**

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