**SINGLE PRECISION RESTORING USING C**

**AIM:**

To write a C program to implement CPU performance using C.  
  
  
  
  
  
**ALGORITHM:**  
  
  
1)      Store  
the remainder when the number is divided by 8 in an array.  
  
  
2)      Divide  
the number by 8 now  
  
  
3)      Repeat  
the above two steps until the number is not equal to 0.  
  
  
4)      Print  
the array in reverse order now.

**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;

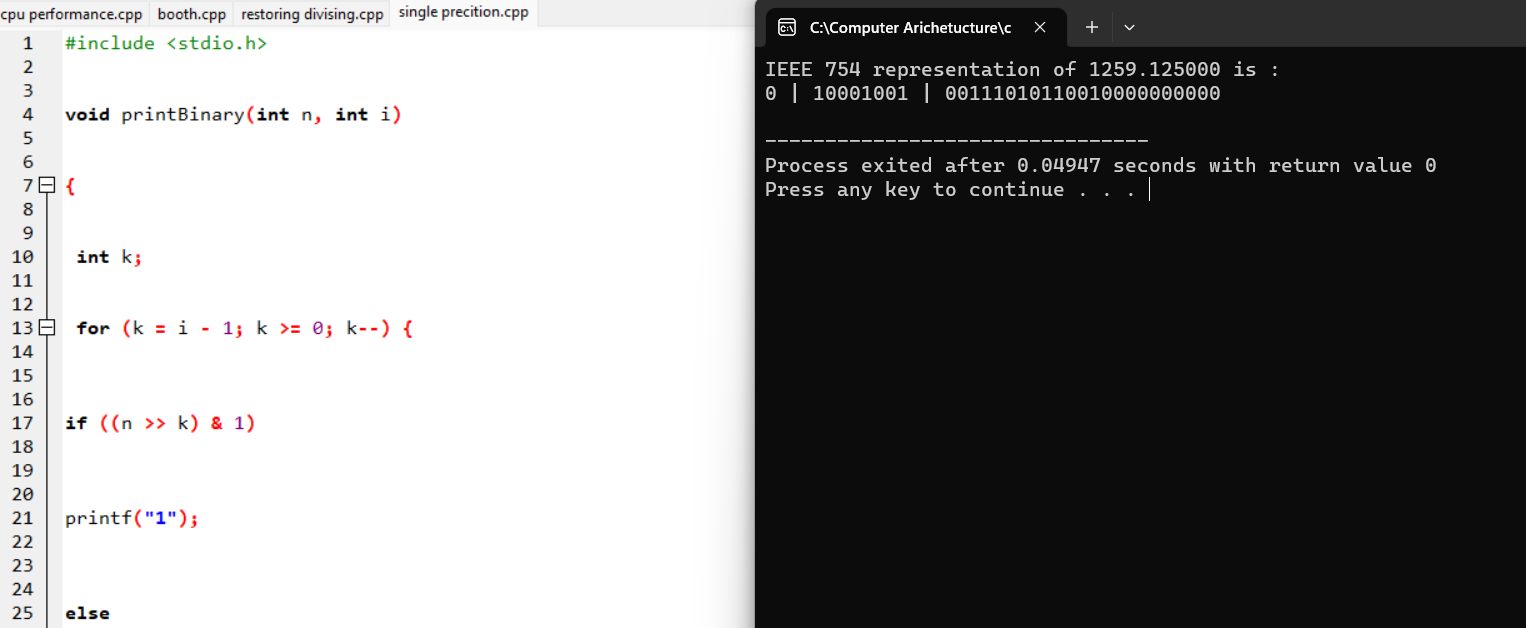
printf("IEEE 754 representation of %f is : \n",var.f);

printIEEE(var);

return 0;

}

**INPUT & OUTPUT:**

****

**RESULT:**

Thus we can write the program for Single Precision Representation in DEV C++.