

## SINGLE PRECISION REPRESENTATION:

### 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);

printfBinary(var.raw.exponent, 8);

printf(" | ");

printfBinary(var.raw.mantissa, 23);

printf("\n");

}
```

```
int main()

{

myfloat var;

var.f = 1259.125;

printf("IEEE 754
representation of %f is : \n",

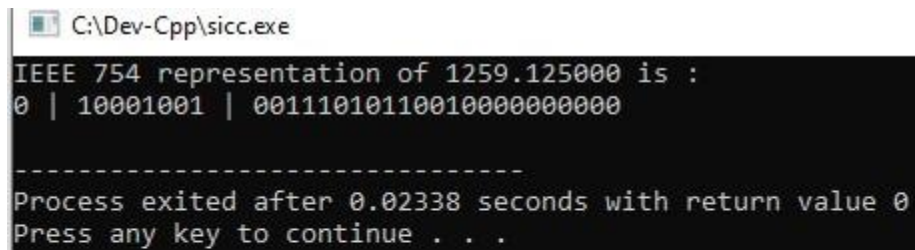
var.f);

printIEEE(var);

return 0;
```

}

INPUT & OUTPUT:



A screenshot of a DevC++ console window. The title bar shows the file path "C:\Dev-Cpp\sicc.exe". The console output displays the IEEE 754 representation of the number 1259.125000. The output is as follows:

```
IEEE 754 representation of 1259.125000 is :  
0 | 10001001 | 001110101100100000000000  
  
-----  
Process exited after 0.02338 seconds with return value 0  
Press any key to continue . . .
```

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