EXPT NO: **DATE:**

BINARY TO DECIMAL CONVERSION

Theory: The idea is to extract the digits of given binary number starting from right most digit and keep a variable dec_value. At the time of extracting digits from the binary number, multiply the digit with the proper base (Power of 2) and add it to the variable dec_value. At the end, the variable dec_value will store the required decimal number.

For Example:
If the binary number is 111.
dec_value = 1*(2^2) + 1*(2^1) + 1*(2^0) = 7

Algorithm:

1. Start
2. Declare float bin and initialize it to binary number, s2=0
3. Declare d of int datatype and initialise it to bin, r, s1=0, i=0
4. Repeat until d=0
    a. r=d%10
    b. if r=1 then, $s1=s1+2^i$
    c. i=i+1
5. d=bin, bin=bin-d, i=1
6. Repeat until bin=0
    a. Bin=bin*10, d=bin
    b. If d ≠0 then, $s2=s2+1/2^i$
    c. I=i+1,bin=bin-d
7. Return s1+s2
8. Stop

Programme:

```
#include<iostream>
#include<cmath>
#include<iomanip>

using namespace std;

double b2d(double bin)
{
   int d=bin,r=0,i=0;
   float s1=0,s2=0;

   while(d!=0)
   {
      r=d%10;
      d=d/10;
      s1=s1+r*pow(2,i);
      i++;
   }

   d=bin;
   bin=bin-d;
   i=1;
   while(bin!=0)
   {
      bin=bin*10;
      d=bin;
      if(d!=0)
      s2=s2+1/pow(2,i);
      i++;
      bin=bin-d;
   }

   return (s1+s2);
}

int main()
{
   double a;
   cout<<"Give the binary number\n";
   cin>>a;

   cout<<"The decimal equivalent of "<<a<<" is
"<<b2d(a);

   return 0;
}
```

Output:

```
PS C:\Users\PANKAJ PATIL\Desktop\COA> g++ 1.cpp
PS C:\Users\PANKAJ PATIL\Desktop\COA> ./a
Give the binary number
11011.101101
The decimal equivalent of 11011.1 is 27.7034
```

Conclusion:

The Programme for Binary to Decimal Conversion was successfully written, debugged ,compiled and executed.