**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
5. r=d%10
6. if r=1 then, s1=s1+2ⁱ
7. i=i+1
8. d=bin, bin=bin-d, i=1
9. Repeat until bin=0
10. Bin=bin\*10, d=bin
11. If d ≠0 then, s2=s2+1/2 ⁱ
12. I=i+1,bin=bin-d
13. Return s1+s2
14. 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";

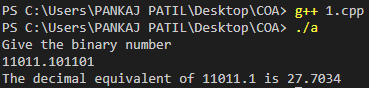
cin>>a;

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

return 0;

}

**Output:**

****

**Conclusion:**

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