

DECIMAL TO BINARY CONVERSION

EXP NO: 21

AIM: To write a C program to implement decimal to binary conversion.

ALGORITHM:

- 1) Check if your number is odd or even.
- 2) If it's even, write 0 (proceeding backwards, adding binary digits to the left of the result).
- 3) Otherwise, if it's odd, write 1 (in the same way).
- 4) Divide your number by 2 (dropping any fraction) and go back to step 1. Repeat until your original number is 0.

PROGRAM:

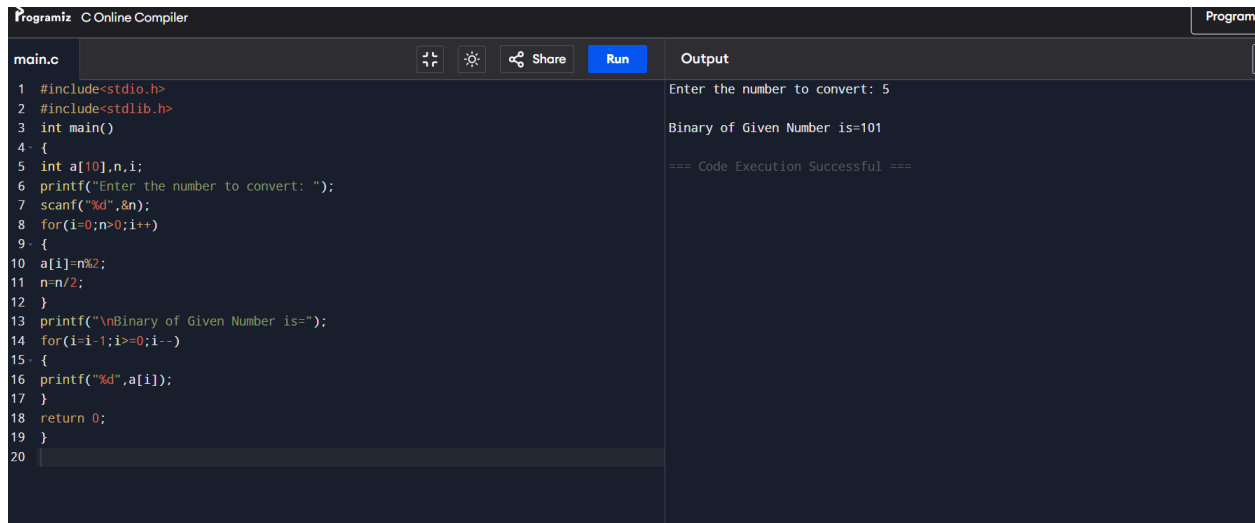
```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int a[10],n,i;
    printf("Enter the number to convert: ");
    scanf("%d",&n);
    for(i=0;n>0;i++)
    {
        a[i]=n%2;
        n=n/2;
    }
```

```
printf("\nBinary of Given Number is=");  
for(i=i-1;i>=0;i--)  
{  
printf("%d",a[i]);  
}  
return 0;  
}
```

INPUT:

5

OUTPUT:



The screenshot shows a web-based C compiler interface. The left pane displays the source code for a program named 'main.c'. The code includes standard headers, a main function that prompts the user for a number, reads the input '5', and then prints the binary representation '101'. The right pane shows the output of the program, which matches the expected result. The status bar at the bottom indicates that the code execution was successful.

```
1 #include<stdio.h>  
2 #include<stdlib.h>  
3 int main()  
4 {  
5     int a[10],n,i;  
6     printf("Enter the number to convert: ");  
7     scanf("%d",&n);  
8     for(i=0;n>0;i++)  
9     {  
10        a[i]=n%2;  
11        n=n/2;  
12    }  
13    printf("\nBinary of Given Number is=");  
14    for(i=i-1;i>=0;i--)  
15    {  
16        printf("%d",a[i]);  
17    }  
18    return 0;  
19 }  
20
```

Enter the number to convert: 5
Binary of Given Number is=101
=== Code Execution Successful ===

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

