EXPERIMENT-11

1.AIM: Write a program to apply bitwise OR, AND and NOT operators on bit level.

CODE:

```
#include <stdio.h>
int main()
{
int a,b;
printf("enter two numbers: ");
if(scanf("%d %d", &a, &b)!=2)
{
printf("invalid input:");
return 0;
}
printf("a = %d, b = %d\n", a, b);
printf("a & b = %d\n", a & b); // AND
printf("a | b = %d\n", a | b); // OR
printf("~a = %d\n", ~a); // NOT → flips all bits

return 0;
}
```

OUTPUT:

```
1.f
invalid input:
```

2.

```
5 gh
invalid input:
```

3. enter two numbers: 5 6

a = 5, b = 6

a & b = 4

a | b = 7

~a = -6

2.AIM: Write a program to apply left shift and right shift operator.

CODE:

```
#include <stdio.h>

int main()
{
int num, shift;//declaration

printf("Enter a number: ");//number to be shifted
if(scanf("%d", &num)!=1)//condition to accept only numbers
{
printf("invalid input");
return 0;
}

printf("Enter the number of positions to shift: ");
if(scanf("%d", &shift)!=1)//condition to accept only numbers
{
printf("invalid input");
return 0;
}
printf("Original number: %d\n", num);
printf("Left shift by %d: %d\n", shift, num << shift);
printf("Right shift by %d: %d\n", shift, num >> shift);

return 0;
}
```

OUTPUT:

```
1. Enter a number: 5

Enter the number of positions to shift: R

invalid input%

2. Enter a number: 5

Enter the number of positions to shift: 2

Original number: 5

Left shift by 2: 20

Right shift by 2: 1
```