

DAY 4

1. Write a C program to determine if the least significant bit of a given integer is set (i.e., check if the number is odd).

```
#include <stdio.h>

int main()
{
    int num;
    printf("Enter the number:");
    scanf("%d", &num);
    if(num&1)
        printf("Odd number\n");
    else
        printf("Even number\n");
}
```

2. Create a C program that retrieves the value of the nth bit from a given integer.

```
#include <stdio.h>

int main()
{
    int num, n;
    printf("Enter the number: ");
    scanf("%d",&num);
    printf("Enter the bit to be retrieved:");
    scanf("%d",&n);

    (num&(1<<n)) == 0 ? printf("0") : printf("1");
}
```

3. Develop a C program that sets the nth bit of a given integer to 1.

```
#include <stdio.h>
```

```
int main()
{
    int num, n;
    printf("Enter the number: ");
    scanf("%d",&num);
    printf("Enter the bit to be set:");
    scanf("%d",&n);

    printf("The number after setting %d th bit is %d\n",n,(num | (1<<n)));
}
```

4. Write a C program that clears (sets to 0) the nth bit of a given integer.

```
#include <stdio.h>
```

```
int main()
{
    int num, n;
    printf("Enter the number: ");
    scanf("%d",&num);
    printf("Enter the bit to be cleared:");
    scanf("%d",&n);

    printf("The number after setting %d th bit is %d\n",n,(num & (~ (1<<n))));
}
```

5. Create a C program that toggles the nth bit of a given integer.

```
#include <stdio.h>
```

```
int main()
{
    int num, n;
    printf("Enter the number: ");
    scanf("%d",&num);
    printf("Enter the bit to be toggled:");
    scanf("%d",&n);

    printf("The number after setting %d th bit is %d\n",n,num^(1<<n));
}
```

6. Write a C program that takes an integer input and multiplies it by 2^n using the left shift operator.

```
#include <stdio.h>
int main()
{
    int num, n;
    printf("Enter a number: ");
    scanf("%d", &num);
    printf("Enter the value of n: ");
    scanf("%d", &n);

    printf("The product of %d x  $2^n$  is %d", num, n, num<<n);
    return 0;
}
```

7. Create a C program that counts how many times you can left shift a number before it overflows (exceeds the maximum value for an integer).

```
#include <stdio.h>
```

```
int main()
{
    int num, count = 0;

    printf("Enter a number: ");
    scanf("%d", &num);

    while (num > 0)
    {
```

```

        num <= 1;
        count++;
    }

    printf("The number of left shifts before overflow: %d\n", count);
    return 0;
}

```

8. Write a C program that creates a bitmask with the first n bits set to 1 using the left shift operator.

```
#include <stdio.h>
```

```

int main()
{
    int n;
    printf("Enter the value of n:");
    scanf("%d",&n);

    int mask=(1<<n)-1;
    printf("Bitmask is %d",mask);
}

```

9. Develop a C program that reverses the bits of an integer using left shift and right shift operations.

```
#include <stdio.h>
```

```

int main()
{
    unsigned int num, reversed_num;
    printf("Enter a number: ");
    scanf("%d", &num);

    for (int i = 0; i < 32; i++)
    {
        if (num & 1)
        {
            reversed_num |= (1 << (32 - 1 - i));
        }
        num >>= 1;
    }

    printf("After reversing the number is: %u", reversed_num);
}

```

10. Create a C program that performs a circular left shift on an integer.

```
#include <stdio.h>

int main()
{
    int num, n, ret;
    printf("Enter a number: ");
    scanf("%d", &num);
    printf("The number of left shifts: ");
    scanf("%d", &n);

    ret=( ((1<n)-1)<<(31-n)) & num )>>31-n | (num<<n);

    printf("The value after %d circular left shifts on %d is: %d", n, num, ret);
    return 0;
}
```

11. Write a program to extract bits from 14th to 9th bits of a number.

```
#include <stdio.h>
void printbinary(int num, int n)
{
    for(int i=n ;i>=0; i--)
    {
        if(num&(1<<i))
            printf("1");
        else
            printf("0");
    }
}
int main()
{
    unsigned int num = 0x1234;
    printf("The number in binary is:");
    printbinary(num, 16);    //Print 16 bits

    int res=(num>>9) & 0x3F;
    printf("\nThe number in binary is:");
    printbinary(res, 6);    //Print 6 bits
}
```

12. Write a C program that takes an integer input and divides it by 2^n using the right shift operator.

```
#include <stdio.h>
```

```
int main()
{
    int num, n;
    printf("Enter the number: ");
    scanf("%d", &num);
    printf("Enter the power of 2: ");
    scanf("%d", &n);

    printf("The result of %d / 2^%d is %d", num, n, (num>>n));
    return 0;
}
```

13. Create a C program that counts how many times you can right shift a number before it becomes zero.

```
#include <stdio.h>
```

```
int main()
{
    unsigned int num, count=0;
    printf("Enter the number: ");
    scanf("%d", &num);
    int backup=num; // for printing
    while(num > 0)
    {
        num>>=1;
        ++count;
    }

    printf("%d can be right shifted %d times before turning 0", backup, count);
    return 0;
}
```

14. Write a C program that extracts the last n bits from a given integer using the right shift operator.

```
#include <stdio.h>
```

```
int main()
{
```

```

int num, n, res = 0;
printf("Enter a number: ");
scanf("%d", &num);
printf("The number of bits to be extracted: ");
scanf("%d", &n);

for (int i = 0; i < n; i++)
{
    res |= ((num >> i) & 1) << i;
}

for (int i = n - 1; i >= 0; i--)
{
    if (res & (1 << i))
        printf("1");
    else
        printf("0");
}

return 0;
}

```

15. Develop a C program that uses the right shift operator to create a bitmask that checks if specific bits are set in an integer.

```
#include <stdio.h>
```

```

int main()
{
    int num, n;
    printf("Enter the number: ");
    scanf("%d", &num);
    printf("Enter the bit to be checked: ");
    scanf("%d", &n);

    if ((num >> n) & 1)
        printf("%d th bit of %d is set\n", n, num);
    else
        printf("%d th bit of %d is not set\n", n, num);

    return 0;
}

```