

PHASE 1 DAY4

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)==0)

printf("Number is even,LSB is 0");

else

printf("Number is odd,LSB is 1");

return 0;

}
```

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 an integer: ");

scanf("%d", &num);

printf("Enter the bit position to retrieve: ");

scanf("%d", &n);
```

```

int mask = 1;
for (int i = 0; i < n; i++)
{
    mask = mask*2;
}
if (num & mask)
{
    printf("The %dth bit is: 1\n", n);
}
else
{
    printf("The %dth bit is: 0\n", n);
}
return 0;
}

```

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 an integer: ");

    scanf("%d", &num);

    printf("Enter the bit position to retrieve: ");

    scanf("%d", &n);

    int result=num|(1<<n);

    printf("Result=%d",result);

    return 0;
}

```

```
}
```

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 an integer: ");
    scanf("%d", &num);

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

    int mask=~(1<<n);

    int result=num&mask;

    printf("Result=%d",result);

    return 0;
}
```

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

```
#include<stdio.h>

int main()
{
    int num, n;

    printf("Enter an integer: ");
    scanf("%d", &num);

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

```

Int result=num^(1<<n);

printf("Result=%d",result);

return 0;

}

```

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 an integer: ");

    scanf("%d", &num);

    printf("Enter the value of n: ");

    scanf("%d", &n);

    int result = num << n;

    printf("Result: %d\n",result);

    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()
{
    unsigned int num = 1;

    int shiftCount = 0;

```

```

while ((num << 1) > num)
{
    num <<= 1;
    shiftCount++;
}

printf("You can left shift the number %d times before it overflows.\n", shiftCount);

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,mask=0;
    printf("enter n:");
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        mask=mask|(1<<i);
    }
    printf("Mask=%d",mask);
    return 0;
}

```

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

```

#include<stdio.h>

int main()

```

```

{
Int num,result;
printf("Enter an integer: ");
    scanf("%d", &num);
Int count=8;
for(int i=0;i<count;i++)
{
result=num<<i;
}
printf("Number is =%d",result);
return 0;
}

```

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

```

#include<stdio.h>

int main()
{
int num, n,shift;

    printf("Enter an integer: ");

    scanf("%d", &num);

    printf("Enter the value of n: ");

    scanf("%d", &n);

printf(" number of shift:");

scanf("%d",&shift);

Unsigned int result=(num<<shift)|(num>>(32-shift));

printf("%u",result);

```

```
return 0;
```

```
}
```

11. 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 an integer: ");
```

```
    scanf("%d", &num);
```

```
    printf("Enter the value of n: ");
```

```
    scanf("%d", &n);
```

```
    int result = num >> n;
```

```
    printf("Result: %d\n",result);
```

```
    return 0;
```

```
}
```

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

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

```
int main() {
```

```
    int num;
```

```
    int Count = 0;
```

```
    printf("Enter an integer: ");
```

```
    scanf("%d", &num);
```

```
    while (num > 0)
```

```
{
```

```

        num >>= 1;

        Count++;
    }

    printf("You can right shift the number %d times before it becomes zero.\n",Count);

    return 0;
}

```

13. 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,result ;

    printf("Enter an integer: ");

    scanf("%d", &num);

    printf("Enter the value of n: ");

    scanf("%d", &n);

    result=num &((1<<n)-1);

    printf("Last %d bits: %d\n", n, result);

    return 0;
}

```

14. 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 an integer: ");

scanf("%d", &num);

printf("Enter the value of n: ");

scanf("%d", &n);

int mask =(1<<n)-1;

if((num&mask)==mask)

printf("last %d bits are set",n);

else

printf("last %d bits are not set",n);

}
```