BIT MAGIC

1. CHECK KTH BIT IS SET OR NOT

```
// CPP program to check if k-th bit
// of a given number is set or not
#include <iostream>
using namespace std;

void isKthBitSet(int n, int k)
{
        if (n & (1 << (k - 1)))
                cout << "SET";
        else
                cout << "NOT SET";
}

// Driver code
int main()
{
        int n = 5, k = 1;
        isKthBitSet(n, k);
        return 0;
}
```

2. COUNT SET BITS

```
// C++ program to Count set
// bits in an integer
#include <bits/stdc++.h>
using namespace std;
```

```
unsigned int countSetBits(int n)
{
      unsigned int count = 0;
      while (n) {
         n &= (n - 1);
         count++;
      }
     return count;
}

/* Program to test function countSetBits */
int main()
{
        int i = 9;
        cout << countSetBits(i);
        return 0;
}
```

3. POWER OF TWO

```
#include <iostream>
using namespace std;

bool isPow2(int n)
{
    if(n == 0)
        return true;

    return ((n & (n - 1)) == 0);
}

int main() {

        int n = 4;
        printf("%s", isPow2(n)? "true": "false");
}
```

4. ONE ODD OCCURRING

```
#include <iostream>
using namespace std;

int findOdd(int arr[], int n)
{
    int res = 0;

    for(int i = 0; i < n; i++)
    {
        res = res ^ arr[i];
    }

    return res;
}

int main() {

         int arr[]= {4, 3, 4, 4, 4, 5, 5, 3, 3}, n = 9;

    cout<<findOdd(arr, n);

}
```

5. TWO ODD OCCURRING

```
#include <iostream>
using namespace std;
```

```
void oddAppearing(int arr[], int n)
{

        int xors = 0, res1 = 0, res2 = 0;

        for (int i = 0; i < n; i++)
        xors = xors ^ arr[i];


        int sn = xors & (~(xors - 1));


        for (int i = 0; i < n; i++)
        {
            if ((arr[i] & sn) != 0)
                res1 = res1 ^ arr[i];
            else
                res2 = res2 ^ arr[i];
        }


        cout <<  res1 << " " << res2;
}



int main() {

         int arr[]= {3, 4, 3, 4, 5, 4, 4, 6, 7, 7}, n = 10;

    oddAppearing(arr, n);

}
```

6. POWER SET

```
#include <iostream>
#include <cmath>
using namespace std;

void printPowerSet(string str)
{
        int n = str.length();

        int powSize = pow(2, n);

        for(int counter = 0; counter < powSize; counter++)
        {
                for(int j = 0; j < n; j++)
                {
                        if((counter & (1 << j)) != 0)
              cout<<str[j];
                }

                cout<<endl;
        }
}



int main() {

        string s = "abc";

    printPowerSet(s);

}
```