DAY 44

Elements of Programming Interview :

1. Compute Parity

How would you compute the parity of a very large number of 64 bit words ?

CODE :

PYTHON

```
x=int(input())
def parity_checker(x):
    result=0
    while(x):
        result^=(x&1)
        x>>=1
    return result
if(parity_checker(x)):
    print("Odd Parity")
else:
    print("Even Parity")
```

(OR)

```
x=int(input())
def parity_checker(x):
    result=0
    while(x):
        result^=1
        x&=(x-1)
    return result
if(parity_checker(x)):
    print("Odd Parity")
else:
    print("Even Parity")
```

2. Swap Bits

A 64-bit integer can be viewed as an array of 64 bits,with the bit at index 0 corresponding to least significant bit(LSB) and the bit at index 64 corresponding to most significant bit(MSB).Implement the code that takes as input a 64-bit integer and swaps the bits in that integer at indices i and j.

CODE :

C++

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

```
long swap_it(long x,int i,int j);
int main(){
    long num;
    int i;
    int j;
    cin >>num;
    cin >>i;
    cin >>j;
    cout<<"Result is given by :"<<swap_it(num,i,j);
}
long swap_it(long x,int i,int j){
    if(((x>>i)&1)!=((x>>j)&1)){
        x^=(1L<<i)|(1L<<j);
    }
    return x;
}
```

3. Reverse Bits

Write a function that takes a 64-bit word x and returns a 64-bit word consisting of the bits of x in reverse order.

CODE :

PYTHON

```
num=int(input())
def reverse_bits(x):
    rev_num=0
    while x>0:
        rev_num<<=1
        if x&1==1:
            rev_num^=1
        x>>=1
    return rev_num
print("Reversed Number is :",reverse_bits(num))
```

4. Find the closest element with same weight

Write a function which takes as input an unsigned 64-bit integer x and returns y!=x such that y has the same weight as x and the difference of x and y is as small as possible.you can assume x is not 0 or all 1s i.e 2^64-1

CODE :

C++

```
#include<iostream>
using namespace std;
unsigned long closest_element(unsigned long x);
```

```
int main(){
    unsigned long num;
    cin >>num;
    cout<<"Closest element is given by : "<<closest_element(num);
}
unsigned long closest_element(unsigned long x){
    for(int i=0;i<63;i++){
        if(((x>>i)&1)^((x>>(i+1))&1)){
            x^=(1UL<<i)|(1UL<<(i+1));
            return x;
        }
    }
}
```