```cpp
/*
A 64-bit integer can be viewed as an array of 64 bits,
with the bit at index 0 corresponding to the least significant bit ,
and the bit at index 63 corresponding to the most significant bit.
Implement code that takes as input a 64-bit integer x and swaps the bits at indices i and j.
*/

/* Size of long long int on my system is 8 Bytes = 8*8 = 64 bits. */

#include<iostream>
#include<string.h>
#include<bitset>
#define br cout << "\n";

using namespace std;

long long int swap_bits(long long int x, const int &i, const int &j)
{
    //check i & j as they cannot be same
    if(((x>>i) & 1) != ((x>>j) & 1))
    {
        x ^= (1L<<i) | (1L<<j);
    }

    return x;
}

int main()
{
    ios_base::sync_with_stdio(false);
    long long int x, oldx, newx;
    int i, j;
    cout << "Enter Number:\t";
    cin >> x;
    cout << "Enter i:\t";
    cin >> i;
    cout << "Enter j:\t";
    cin >> j;

    cout << "Binary before swap:\t";
    oldx = x;
    //decimal to binary
    string binary = bitset<64>(oldx).to_string(); //to binary
    cout << binary;
    br;

    //swap routine
    x = swap_bits(x,i,j);

    cout << "Binary after swap:\t";
    newx = x;
    binary = bitset<64>(x).to_string(); // to binary
    cout << binary;
    br;

    cout << "Final number :\t";
    cout << newx << "\n";
}
```