**Problem No: 01**

**Topic: The Foundations: Logic and Proof**

**Problem Title:**

Given two bit strings of length n, find the bitwise AND, bitwise OR and bitwise XOR of these strings.

**Objectives:**

To learn about bitwise AND, bitwise OR and bitwise XOR of two string.

**Theory:**

Bitwise AND Takes two numbers as operands and does AND on every bit of two numbers. The result of AND is 1 only if both bits are 1.

Bitwise OR Takes two numbers as operands and does OR on every bit of two numbers. The result of OR is 1 any of the two bits is 1.

Bitwise XOR ( ^ ) like the other operators (except ~) also take two equal-length bit patterns. If both bits in the compared position of the bit patterns are 0 or 1, the bit in the resulting bit pattern is 0, otherwise 1.

**Source Code:**

#include<bits/stdc++.h>

using namespace std;

int main()

{

int n, i, j, p, q;

cin >> n;

int a[n], b[n], A[n], O[n], X[n];

string s1, s2;

cin >> s1 >> s2;

for(i = 0; i < n; i++){

a[i] = s1[i] - 48;

b[i]= s2[i] - 48;

}

for(i = 0; i < n; i++){

if(a[i] && b[i])

A[i] = 1;

else

A[i] = 0;

if(a[i] == 0 && b[i] == 0)

O[i] = 0;

else

O[i] = 1;

if(a[i] == b[i])

X[i] = 0;

else

X[i] = 1;

}

cout << "AND operation: ";

for(i = 0; i < n; i++)

cout << A[i];

cout << endl << "OR operation: ";

for(i= 0; i < n; i++)

cout << O[i];

cout << endl << "XOR operation: ";

for(i = 0; i < n; i++)

cout << X[i];

cout << endl;

return 0;

}

**Output:**

4

1010

0101

AND operation: 0000

OR operation: 1111

XOR operation: 1111

Process returned 0 (0x0) execution time : 5.684 s

Press any key to continue

**Problem No: 02**

**Topic: The Foundations: Logic and Proof**

**Problem Title:**

Given two set A and B, a proposition p(x,y) where x ε A and y ε B, Find the propositional value of

∀ ∀ *p* ( *x* , *y* ) *x y,* ∀∃ *p* ( *x* , *y* ) *x y,* ∃ ∀ *p* ( *x* , *y* ) *x y,* ∃ ∃ *p ( x , y ) x y.*

**Objectives:**

To learn about universal and existential quantifiers.

**Source Code:**

#include<iostream>

using namespace std;

int Universal\_exeistential()

{

int A[3] = {1,2,3}, B[3] = {1,4,9};

int i,j,count=0;

for(i=0;i<3;i++){

for(j=0;j<3;j++){

if((A[i])\*(A[i])==B[j]){

count++;

}

}

}

if(count==3){

return 1;

}

else{

return 0;

}

}

int Exeistential\_exeistential()

{

int A[3] = {1,2,3}, B[3] = {1,4,9};

int i,j,count=0;

for(i=0;i<3;i++){

for(j=0;j<3;j++){

if((A[i])\*(A[i])==B[j]){

count++;

}

}

}

for(i=0;i<3;i++){

for(j=0;j<3;j++){

if(B[i]==(A[j])\*(A[j])){

count++;

}

}

}

if(count==6){

return 1;

}

else{

return 0;

}

}

int Exeistential\_universal()

{

int A[3] = {1,2,3}, B[3] = {1,4,9};

int i, j, count=0;

for(i=0;i<3;i++){

for(j=0;j<3;j++){

if((A[j])\*(A[j])==B[i]){

count++;

}

}

if(count==3)

{

return 1;

}

else

{

count = 0;

}

}

return 0;

}

int Universal\_universal()

{

int A[3] = {1,2,3}, B[3] = {1,4,9};

int i, j, count=0, count1 = 0;

for(i=0;i<3;i++){

for(j=0;j<3;j++){

if((A[j])\*(A[j])==B[i]){

count++;

}

}

if(count==3){

count1 ++;

}

else{

count = 0;

}

}

for(i=0;i<3;i++){

for(j=0;j<3;j++){

if((A[i])\*(A[i])==B[j]){

count++;

}

}

if(count==3){

count1 ++;

}

else{

count = 0;

}

}

if (count1 == 2){

return 1;

}

else{

return 0;

}

}

int main()

{

int A[3] = {1,2,3}, B[3] = {1,4,9};

int x,y,i,j;

int U\_E = Universal\_exeistential();

cout << "Universal\_exeistential: " << U\_E << endl;

int E\_E =Exeistential\_exeistential();

cout << "Exeistential\_exeistential: " << E\_E << endl;

int E\_U =Exeistential\_universal();

cout << "Exeistential\_universal: " << E\_U << endl;

int U\_U =Universal\_universal();

cout << "Universal\_universal: " << U\_U << endl;

}

**Output:**

Universal\_exeistential: 1

Exeistential\_exeistential: 1

Exeistential\_universal: 0

Universal\_universal: 0

Process returned 0 (0x0) execution time : 0.028 s

Press any key to continue.