**6.1 Function in C++.                       Date: 26-02-22**

“Function is a piece of code that preforms a specific task”.

**syntax:**

        returnType functionName(Parameter1, Parameter2,....){

            //Function body

        }

Example:

void print(int num){

    cout<<num<<endl;

    return;

}

int add(int num1, int num2){

    print(num1);

    print(num2);

    int sum = num1+num2;

    return sum;

}

int main(){

    int a=2;

    int b=3;

    cout<<add(a,b)<<endl;

    return 0;

}

**6.2 Function in C++.  Examples             Date: 26-02-22**

**Example**: Prime Number between two Numbers.

#include<iostream>

#include<math.h>

using namespace std;

bool isPrime(int num){

    for(int i=2; i<=sqrt(num); i++){

        if(num%i==0){

            return false;

        }

    }

    return true;

}

int main(){

    int a, b;

    cin>>a>>b;

    for(int i=a; i<=b; i++){

        if(isPrime(i)){

            cout<<i<<endl;

        }

    }

    return 0;

}

**Example: Fibenacci Sequence?**

t1=0, t2=1 find t3=?...........tn=?

#include<iostream>

using namespace std;

void fib( int n){

    int t1=0;

    int t2=1;

    int nextTerm;

    for(int i=1; i<=n; i++){

        cout<<t1<<endl;

        nextTerm=t1+t2;

        t1=t2;

        t2=nextTerm;

    }

    return;

}

int main()

{

    int n;

    cin>>n;

    fib(n);

    return 0;

}

**Example: Find factorial n ?**

#include<iostream>

using namespace std;

int fact(int n){

    int factorial=1;

    for(int i=2; i<=n; i++){

        factorial=factorial\*i;

        //factorial\*=i;

    }

    return factorial;

}

int main(){

    int n;

    cin>>n;

    int ans = fact(n);

    cout<<ans<<endl;

    return 0;

}

**Example**: Find factorial n and r.

calculate nCr =  \_\_\_n!\_\_\_\_

                  (n-r)! X r!

#include<iostream>

using namespace std;

int fact(int n){

    int factorial=1;

    for(int i=2; i<=n; i++){

        factorial\*=i;

    }

    return factorial;

}

int main()

{

    int n, r;

    cin>>n>>r;

    int ans = fact(n) / ( fact(r)\*fact(n-r) );

    cout<<ans<<endl;

    return 0;

}

**Example**: **Pascal Triangle(using Factorial nCr)**

n=5        Output:      1

i=row                   1 1

j=coloum                 1 2 1

                        1 3 3 1

                        1 4 6 4 1

#include<iostream>

using namespace std;

int fact(int n){

    int factorial=1;

    for(int i=2; i<=n; i++){

        factorial\*=i;

    }

    return factorial;

}

int main()

{

    int n;

    cin>>n;

    for (int i=0; i<n; i++){

        for (int j=0; j<=i; j++){

            cout<<fact(i) / (fact(j)\*fact(i-j))<<" ";

        }

        cout<<endl;

    }

    cout<<endl;

    return 0;

}

**6.3  Practice Question of Function in C++.        Date: 26-02-22**

1. Sum of first n natural Numbers.

2. check Pythagorial Triplet.

3. Conversions.

    a. Binary to Decimal.

    B. Octal to decimal.

    c. Hexadecimal to decimal

    d. Decimal to Binary.

    e. Decimal to Octal.

    f. Decimal to Hexadecimal.

4. Add two Binary Numbers.

**1. Sum of first n natural Numbers.**

    S = 1+2+3+4+5+6+...n.

      =  \_\_\_n(n+1)\_\_

            2

#include<bits/stdc++.h>

using namespace std;

int sum(int n){

    int ans = 0;

    for ( int i=1; i<=n; i++)

        ans += i;

    return ans;

}

int main(){

    int n;

    cin >> n;

    cout << sum(n) <<endl;

}

**2. Check Pythagorial Triplet ?**

#include<bits/stdc++.h>

using namespace std;

bool check(int x, int y, int z){

    int a = max(x, max(y,z));

    int b, c;

    if(a==x){

        b=y;

        c=z;

    }

    else if(a==y){

        b=x;

        c=z;

    }

    else{

        b=x;

        c=y;

    }

    if(a\*a ==b\*b + c\*c)

        return true;

    else

        return false;

}

int main(){

    int x, y, z;

    cin >> x >> y >> z;

    if (check(x,y,z)){

        cout << "Pythagorean Triplet";

    }

    else{

        cout << "Not a Pythagorean Triplet";

    }

    return 0;

}

**3. Conversions.**

**A. Binary to Decimal.**

Eg.(101101)2 = (1X25) + (0) + (1x23) + (1X22) + (0) + (1X20)

              = 32+8+4+1

              = (45)10

**Example:**

#include<bits/stdc++.h>

using namespace std;

int binaryToDecimal(int n){

    int ans = 0;

    int x = 1;

    while(n>0){

        int y = n%10;

        ans += x\*y;

        x \*=2;

        n /= 10;

    }

    return ans;

}

int main(){

    int n;

    cout <<"Enter the Binary To Decimal\n";

    cin>>n;

    cout << binaryToDecimal(n) << endl;

    return 0;

}

**b. Octal to decimal.**

{0,1,2,3,4,5,6,7}

Representation:  (X)8

**Octal converting to Decimal**

Eg. (137)8

(137)8  = 7x80 + 3X81 + 1X82

        =  7 + 24 + 64

        =  (95)10

Algorithm: Traverse over the Digits and make the Decimal Number.

**Example:**

#include<bits/stdc++.h>

using namespace std;

int octalToDecimal(int n){

    int ans = 0;

    int x = 1;

    while(n>0){

        int y = n%10;

        ans += x\*y;

        x \*=8;

        n /= 10;

    }

    return ans;

}

int main(){

    int n;

    cout <<"Enter the Octal To Decimal\n";

    cin>>n;

    cout << octalToDecimal(n) << endl;

    return 0;

}

**C. Hexadecimal to decimal.**

 {0,1,2,3 ....9,A,B,C,D,E,F}

 A=10, B=11  ... E=14, F=15

**Hexadecimal converting to Decimal**

Eg. (1CF)16

(1CF)16 = 15X160 + 12X161 + 1X162

        = 15 + 192 + 256

        = (463)10

**Algorithm**: Traverse over the Digits and make the Decimal Number.

#include<bits/stdc++.h>

using namespace std;

int hexadecimalToDecimal(string n){

    int ans = 0;

    int x = 1;

    int s = n.size();

    for(int i=s-1; i>=0; i--){

        if(n[i] >= '0' && n[i] <='9'){

            ans += x\*(n[i]-'0');

        }

        else if(n[i] >='A' && n[i] <='F'){

            ans += x\*(n[i]-'A' + 10);

        }

        x \*= 16;

    }

    return ans;

}

int main(){

    string n;

    cout <<"Enter the Hexadecimal To Decimal\n";

    cin>>n;

    cout << hexadecimalToDecimal(n) << endl;

    return 0;

}

**D. Decimal to Binary.**

**Eg. (45)10**

Division by2 : 45/2 22/2 11/2 5/2 2/2 1/2

Quotient : 22 11 5 2 1 0

Remainder : 1 0 1 1 0 1

Bit# : 0 1 2 3 4 5

(45)10 = (101101)2

#include<bits/stdc++.h>

using namespace std;

int decimalToBinary(int n){

    int x = 1;

    int ans = 0;

    while(x<=n)

        x\*=2;

    x/=2;

    while(x>0){

        int lastDigit = n/x;

        n -= lastDigit\*x;

        x/=2;

        ans = ans\*10 + lastDigit;

    }

    return ans;

}

int main(){

    int n;

    cout <<"Enter the Decimal to Binary\n";

    cin>>n;

    cout << decimalToBinary(n) << endl;

    return 0;

}

**E. Decimal to Octal.**

Eg. (100)10 = (x)8

(100)10 =  1X82 + 4X81 + 4X80

        =  (144)8

Algorithm: Find the Highest Power of 8, from which over number is divisible,

then reduce the number. do this unit over number becomes 0.

**Example:**

#include<bits/stdc++.h>

using namespace std;

int decimalToOctal(int n){

    int x = 1;

    int ans = 0;

    while(x<=n)

        x\*=8;

    x/=8;

    while(x>0){

        int lastDigit = n/x;

        n -= lastDigit\*x;

        x/=8;

        ans = ans\*10 + lastDigit;

    }

    return ans;

}

int main(){

    int n;

    cout <<"Enter the Decimal to Octal\n";

    cin>>n;

    cout << decimalToOctal(n) << endl;

    return 0;

}

**F. Decimal to hexadecimal**

**Ex.** (100)10 = ( x )16

= 6X161 + 4X160

= (64)16

**Example:**

#include<bits/stdc++.h>

using namespace std;

string decimalToHexadecimal(int n){

    int x = 1;

    string ans = "";

    while(x<=n)

        x\*=16;

    x/=16;

    while(x>0){

        int lastDigit = n/x;

        n -= lastDigit\*x;

        x/=16;

        if(lastDigit <= 9)

            ans = ans + to\_string(lastDigit);

        else{

            char c = 'A' + lastDigit - 10;

            ans.push\_back(c);

        }

    }

    return ans;

}

int main(){

    int n;

    cout <<"Enter the Decimal to Hexadecimal\n";

    cin>>n;

    cout << decimalToHexadecimal(n) << endl;

    return 0;

}

**4. Add two Binary Numbers.**

a = 10101   1 1 1 1 1

b = 11011     1 0 1 0 1

            + 1 1 0 1 1

            1 1 0 0 0 0

#include<bits/stdc++.h>

using namespace std;

int reverse(int n){   //2nd function create

    int ans = 0;

    while(n>0){

        int lastDigit = n%10;

        ans = ans\*10 + lastDigit;

        n/=10;

    }

    return ans;

}

int addbinary(int a, int b){   //1st function create

    int ans = 0;

    int prevCarry = 0;

    while(a>0 && b>0){

        if(a%2 == 0 && b%2 == 0){

            ans = ans\*10 + prevCarry;

            prevCarry = 0;

        }

        else if((a%2==0 && b%2==1) || (a%2==1 && b%2==0)){

            if(prevCarry == 1){

                ans = ans\*10 + 0;

                prevCarry = 1;

            }

            else{

                ans = ans\*10 + 1;

                prevCarry = 0;

            }

        }

        else{

            ans =ans\*10 + prevCarry;

            prevCarry = 1;

        }

        a /=10;

        b /=10;

    }

    while(a>0){

        if(prevCarry == 1){

            ans == ans\*10 + 0;

            prevCarry = 1;

        }

        else{

            ans = ans\*10 + 1;

            prevCarry = 0;

        }

    }

    while(a>0){

        if(prevCarry == 1){

            if(a%2 == 1){

                ans = ans\*10 + 1;

                prevCarry = 1;

            }

            else{

                ans = ans\*10 + 1;

                prevCarry = 0;

            }

        }

        else{

            ans = ans\*10 + (a%2);

        }

        a/=10;

    }

    while(a>0){

        if(prevCarry == 1){

            ans == ans\*10 + 0;

            prevCarry = 1;

        }

        else{

            ans = ans\*10 + 1;

            prevCarry = 0;

        }

    }

    while(b>0){

        if(prevCarry == 1){

            if(b%2 == 1){

                ans = ans\*10 + 1;

                prevCarry = 1;

            }

            else{

                ans = ans\*10 + 1;

                prevCarry = 0;

            }

        }

        else{

            ans = ans\*10 + (b%2);

        }

        b/=10;

    }

    if(prevCarry == 1){

        ans = ans\*10 + 1;

    }

    ans = reverse(ans); //2nd reverse function used

    return ans;

}

int main() {

    int a, b;

    cin >> a >> b;

    cout<<addbinary(a,b)<<endl;  //1st function used

    return 0;

}