VIT Vidyalankar Institute of Technology Accredited A+ by NAAC	**DEPARTMENT OF COMPUTER ENGINEERING**

VIT Vidyalankar Institute of Technology Accredited A+ by NAAC

Experiment No. 3A

Semester	T.E. Semester VI
Subject	ARTIFICIAL INTELLIGENCE (CSL 604)
Subject Professor In-charge	Prof. Avinash Shrivas
Assisting Teachers	Prof. Avinash Shrivas

Student Name	Deep Salunkhe
Roll Number	21102A0014
Lab Number	310A

Title:

Mathematical solution to Water jug problem

Theory:

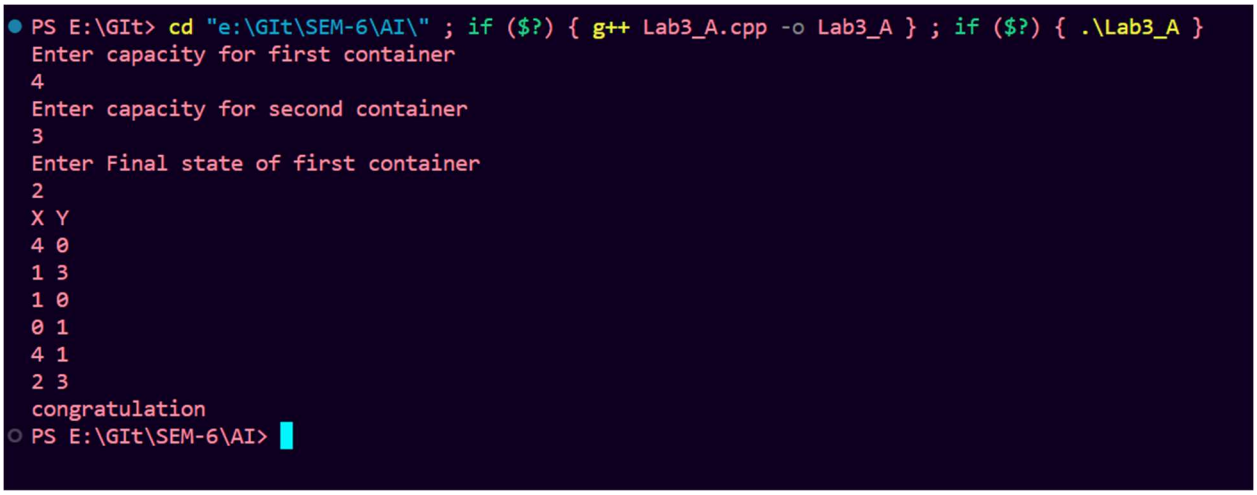
1. The **gcd** function calculates the greatest common divisor of two numbers, which is used to check if it's possible to measure out the desired quantity of water using the given jug capacities.
2. The **ispossible** function checks if it's possible to measure out the desired quantity of water using the given jug capacities and the **finalx** quantity.
3. The **fillx** function fills the first jug to its maximum capacity.
4. The **emptyy** function empties the second jug.
5. The **transfer_x_to_y** function transfers water from the first jug to the second jug until the second jug is full or the first jug is empty.
6. In the **main** function, the user inputs the capacities of the two jugs (**xcapacity** and **ycapacity**) and the desired quantity of water (**finalx**). It then checks if it's possible to achieve the desired quantity of water with the given capacities using the **ispossible** function.
7. Finally, when the desired quantity is achieved, the program prints "congratulation" and terminates.

Program Code:

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

int gcd(int x,int y){

    if(y==0)
    return x;
    else
    return gcd(y,x%y);
}

bool ispossible(int x,int y,int d){
    int valgcd=gcd(x,y);
    if(d%valgcd==0)
    return true;
    return false;
}

void fillx(int &xState,int &xcapacity){
    xState=xcapacity;
}

void emptyy(int &yState,int ycapacity){
    yState=0;
}

void transfer_x_to_y(int &xState,int &yState,int &xcapacity,int &ycapacity){

    int y_can_take=(ycapacity-yState);
    if(y_can_take>=xState){
        yState=yState+xState;
        xState=0;
    }else{
        xState=xState-y_can_take;
        yState=ycapacity;

    }
}

int main(){

    int xcapacity,ycapacity,finalx;
    int xState=0,yState=0;
```

```
    bool possible=true;
    cout<<"Enter capacity for first container"<<endl;
    cin>>xcapacity;

    cout<<"Enter capacity for second container"<<endl;
    cin>>ycapacity;

    cout<<"Enter Final state of first container"<<endl;
    cin>>finalx;

    possible=ispossible(xcapacity,ycapacity,finalx);
    if(!possible){
        cout<<"Not possible"<<endl;
        return 0;
    }

    cout<<"X"<<" "<<"Y"<<endl;

    while(xState!=finalx){
        if(xState==0){
            fillx(xState,xcapacity);
        }else if(yState==ycapacity){
            emptyy(yState,ycapacity);
        }else{
            transfer_x_to_y(xState,yState,xcapacity,ycapacity);
        }

        cout<<xState<<" "<<yState<<endl;
        if(xState==finalx ||  yState==finalx ){
            cout<<"congratulation"<<endl;
            return 0;
        }

    }




    return 0;
}
```

Output:



Conclusion:

The provided code effectively tackles the Water Jug Problem, employing fundamental programming constructs like conditionals, loops, and functions. It illustrates the application of mathematical concepts, such as the greatest common divisor, in solving real-world problems through programming. By simulating the pouring of water between jugs until the desired quantity is achieved, the code demonstrates problem-solving techniques and reinforces programming proficiency. This lab exercise serves as a practical example for understanding problem-solving methodologies