```
    import numpy as np

    def incmatrix(genl1,genl2):
        m = len(genl1)
        n = len(genl2)
        M = None #to become the incidence matrix
        VT = np.zeros((n*m,1), int)  #dummy variable

        #compute the bitwise xor matrix
        M1 = bitxormatrix(genl1)
        M2 = np.triu(bitxormatrix(genl2),1)

        for i in range(m-1):
            for j in range(i+1, m):
                [r,c] = np.where(M2 == M1[i,j])
                for k in range(len(r)):
                    VT[(i)*n + r[k]] = 1;
                    VT[(i)*n + c[k]] = 1;
                    VT[(j)*n + r[k]] = 1;
                    VT[(j)*n + c[k]] = 1;

                    if M is None:
                        M = np.copy(VT)
                    else:
                        M = np.concatenate((M, VT), 1)

                    VT = np.zeros((n*m,1), int)

        return M

```

Listing 1: Python example

```
package assignment_1;
import java.util.Scanner;

public class Employee {

    // create an object of Scanner
    Scanner input = new Scanner(System.in);

    int emp_id;
    int age, basic_sal, da, ta;
    String address_city, position, name;
    static int ssn;

    Employee()
    {
        System.out.println("Default Constructor was called");
    }

    // Parameterized Constructor
    Employee(int e, int a, int b, int d, int t, String add, String
    pos, String nam)
    {
        System.out.println("Parameterized constructor was called");
        emp_id = e;
        age = a;
```

```
        basic_sal = b;
        da = d;
        ta = t;
        address_city = add;
        position = pos;
        name = nam;
    }

    // Copy Constructor
    Employee(Employee E)
    {
        System.out.println("Copy constructor was called");
        emp_id = E.emp_id;
        age = E.age;
        basic_sal = E.basic_sal;
        da = E.da;
        ta = E.ta;
        address_city = E.address_city;
        position = E.position;
        name = E.name;
    }

    double calc_gross_sal()
    {
        return basic_sal + da + ta - (0.15 * basic_sal);
    }

    void display()
    {
        ssn = ssn + 1;
        System.out.println("Employee ssn is: " + ssn);
        System.out.println("Employee ID is : " + emp_id);
        System.out.println("Employee Name: " + name);
        System.out.println("Employee Age: " + age);
        System.out.println("Employee Position: " + position);
        System.out.println("Employee basic Salary: " + basic_sal);
        System.out.println("Employee DA: " + da);
        System.out.println("Employee TA: " + ta);
        System.out.println("Employee Gross Salary: " +
    calc_gross_sal() );
        System.out.println("Employee Address City: " + address_city
    );
        System.out.println("\n");
    }

    void accept()
    {
        System.out.println("Enter the age :");
        age = input.nextInt();
        System.out.println("Employee ID is: ");
        emp_id = input.nextInt();
        System.out.println("Employee Name: " );
        name = input.next();
        System.out.println("Employee Age: " );
        age = input.nextInt();
        System.out.println("Employee Position: " );
        position = input.next();
```

```
        System.out.println("Employee basic Salary: ");
        basic_sal = input.nextInt();
        System.out.println("Employee DA: ");
        da = input.nextInt();
        System.out.println("Employee TA: ");
        ta = input.nextInt();
        System.out.println("Employee Address City: ");
        address_city = input.next();
    }

    // @Override
    // protected void finalize() throws Throwable
    // {
    //     try
    //     {
    //         input.close();
    //     }
    //     catch(Throwable t)
    //     {
    //         throw t;
    //     }
    //     finally
    //     {
    //         super.finalize();
    //     }
    // }

}
```