

```

1  import numpy as np
2
3  def incmatrix(genl1,genl2):
4      m = len(genl1)
5      n = len(genl2)
6      M = None #to become the incidence matrix
7      VT = np.zeros((n*m,1), int) #dummy variable
8
9      #compute the bitwise xor matrix
10     M1 = bitxormatrix(genl1)
11     M2 = np.triu(bitxormatrix(genl2),1)
12
13     for i in range(m-1):
14         for j in range(i+1, m):
15             [r,c] = np.where(M2 == M1[i,j])
16             for k in range(len(r)):
17                 VT[(i)*n + r[k]] = 1;
18                 VT[(i)*n + c[k]] = 1;
19                 VT[(j)*n + r[k]] = 1;
20                 VT[(j)*n + c[k]] = 1;
21
22             if M is None:
23                 M = np.copy(VT)
24             else:
25                 M = np.concatenate((M, VT), 1)
26
27             VT = np.zeros((n*m,1), int)
28
29     return M
30

```

Listing 1: Python example

```

1  package assignment_1;
2  import java.util.Scanner;
3
4  public class Employee {
5
6      // create an object of Scanner
7      Scanner input = new Scanner(System.in);
8
9      int emp_id;
10     int age, basic_sal, da, ta;
11     String address_city, position, name;
12     static int ssn;
13
14     Employee()
15     {
16         System.out.println("Default Constructor was called");
17     }
18
19     // Parameterized Constructor
20     Employee(int e, int a, int b, int d, int t, String add, String
pos, String nam)
21     {
22         System.out.println("Parameterized constructor was called");
23         emp_id = e;
24         age = a;

```

```

25         basic_sal = b;
26         da = d;
27         ta = t;
28         address_city = add;
29         position = pos;
30         name = nam;
31     }
32
33     // Copy Constructor
34     Employee(Employee E)
35     {
36         System.out.println("Copy constructor was called");
37         emp_id = E.emp_id;
38         age = E.age;
39         basic_sal = E.basic_sal;
40         da = E.da;
41         ta = E.ta;
42         address_city = E.address_city;
43         position = E.position;
44         name = E.name;
45     }
46
47     double calc_gross_sal()
48     {
49         return basic_sal + da + ta - (0.15 * basic_sal);
50     }
51
52     void display()
53     {
54         ssn = ssn + 1;
55         System.out.println("Employee ssn is: " + ssn);
56         System.out.println("Employee ID is : " + emp_id);
57         System.out.println("Employee Name: " + name);
58         System.out.println("Employee Age: " + age);
59         System.out.println("Employee Position: " + position);
60         System.out.println("Employee basic Salary: " + basic_sal);
61         System.out.println("Employee DA: " + da);
62         System.out.println("Employee TA: " + ta);
63         System.out.println("Employee Gross Salary: " +
64         calc_gross_sal() );
65         System.out.println("Employee Address City: " + address_city
66     );
67         System.out.println("\n");
68     }
69
70     void accept()
71     {
72         System.out.println("Enter the age :");
73         age = input.nextInt();
74         System.out.println("Employee ID is: ");
75         emp_id = input.nextInt();
76         System.out.println("Employee Name: ");
77         name = input.next();
78         System.out.println("Employee Age: ");
79         age = input.nextInt();
80         System.out.println("Employee Position: ");
81         position = input.next();

```

```

80     System.out.println("Employee basic Salary: ");
81     basic_sal = input.nextInt();
82     System.out.println("Employee DA: ");
83     da = input.nextInt();
84     System.out.println("Employee TA: ");
85     ta = input.nextInt();
86     System.out.println("Employee Address City: ");
87     address_city = input.next();
88 }
89
90 // @Override
91 // protected void finalize() throws Throwable
92 // {
93 //     try
94 //     {
95 //         input.close();
96 //     }
97 //     catch(Throwable t)
98 //     {
99 //         throw t;
100 //     }
101 //     finally
102 //     {
103 //         super.finalize();
104 //     }
105 // }
106
107 }

```