```
1
      import numpy as np
2
      def incmatrix(genl1,genl2):
3
      m = len(genl1)
4
      n = len(gen12)
5
      M = None #to become the incidence matrix
6
      VT = np.zeros((n*m,1), int) #dummy variable
       #compute the bitwise xor matrix
10
      M1 = bitxormatrix(genl1)
      M2 = np.triu(bitxormatrix(genl2),1)
11
12
      for i in range(m-1):
13
14
        for j in range(i+1, m):
           [r,c] = np.where(M2 == M1[i,j])
15
           for k in range(len(r)):
16
17
            VT[(i)*n + r[k]] = 1;
             VT[(i)*n + c[k]] = 1;
18
19
            VT[(j)*n + r[k]] = 1;
            VT[(j)*n + c[k]] = 1;
20
21
            if M is None:
22
            M = np.copy(VT)
23
24
             else:
            M = np.concatenate((M, VT), 1)
25
26
             VT = np.zeros((n*m,1), int)
27
28
      return M
29
30
```

```
import numpy as np
2
3
       def incmatrix(genl1,genl2):
      m = len(genl1)
4
      n = len(gen12)
5
      {\tt M} = {\tt None} #to become the incidence matrix
      VT = np.zeros((n*m,1), int) #dummy variable
7
       \# compute the bitwise xor matrix
9
10
       M1 = bitxormatrix(genl1)
       M2 = np.triu(bitxormatrix(genl2),1)
11
12
13
       for i in range(m-1):
       for j in range(i+1, m):
14
       [r,c] = np.where(M2 == M1[i,j])
15
       for k in range(len(r)):
16
       VT[(i)*n + r[k]] = 1;
17
18
       VT[(i)*n + c[k]] = 1;
       VT[(j)*n + r[k]] = 1;
19
       VT[(j)*n + c[k]] = 1;
20
21
       if M is None:
22
      M = np.copy(VT)
23
       else:
24
25
       M = np.concatenate((M, VT), 1)
26
      VT = np.zeros((n*m,1), int)
```

```
28
29
       return M
30
       import numpy as np
1
2
       def incmatrix(genl1,genl2):
3
4
       m = len(genl1)
 5
       n = len(gen12)
       M = None #to become the incidence matrix
6
       VT = np.zeros((n*m,1), int) #dummy variable
       #compute the bitwise xor matrix
9
10
       M1 = bitxormatrix(genl1)
       M2 = np.triu(bitxormatrix(genl2),1)
11
12
       for i in range(m-1):
13
       for j in range(i+1, m):
[r,c] = np.where(M2 == M1[i,j])
14
15
       for k in range(len(r)):
16
17
       VT[(i)*n + r[k]] = 1;
       VT[(i)*n + c[k]] = 1;
18
       VT[(j)*n + r[k]] = 1;
19
       VT[(j)*n + c[k]] = 1;
20
21
22
       if M is None:
       M = np.copy(VT)
23
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

```
function X = BitXorMatrix(A,B)
_{2} % function to comput the sum without charge of two vectors
3
      % convert elements into usigned integers
4
      A = uint8(A);
      B = uint8(B);
6
      m1 = length(A);
8
      m2 = length(B);
9
      X = uint8(zeros(m1, m2));
10
      for n1 = 1:m1
11
          for n2 = 1:m2
12
              X(n1,n2) = bitxor(A(n1), B(n2))
13
14
15 end
```