```
#initialize strings u & v
u = '01101110'
v= '10101000'
w = '0110110011101000'

len_u = len(u)
len_v = len(v)
len_w = len(w)

#Below matrix is to save states at each stage of nodes which is initialized to False.
#0th node is  made as (True ) and then based on the conditions this (True) will be traversed
#Main goal is True should be traversed until the complete W is evaluated
#At the end if True is teh value of teh last node, it means our W is teh interleaving of u & v.


Matrix_To_Result = [[False for i in range(len_u+1)] for i in range(len_v+1)]


for x in range(len_u+1):
    for y in range(len_v+1):
        #Belwo condition is for the inital node (0,0) which is just a starting point
        if(x == 0 and y ==0):
            Matrix_To_Result[x][y] = True;

        elif(x == 0):
            if(v[y-1] == w[y-1]):
                Matrix_To_Result[x][y] = Matrix_To_Result[x][y-1];
        elif(y == 0):
            if(u[x-1] == w[x-1]):
                Matrix_To_Result[x][y] = Matrix_To_Result[x-1][y];

        elif(u[x-1] == w[x+y-1] and v[y-1] != w[x+y-1]):
            Matrix_To_Result[x][y] = Matrix_To_Result[x-1][y];
        elif(u[x-1] !=w[x+y-1] and v[y-1] == w[x+y-1]):
            Matrix_To_Result[x][y] = Matrix_To_Result[x][y-1];



        elif(u[x-1]==w[x+y-1] and v[y-1] == w[x+y-1]):
            Matrix_To_Result[x][y] = Matrix_To_Result[x-1][y] or Matrix_To_Result[x][y-1];




if(Matrix_To_Result[x][y] == 1):
    print('Yes')
else:
    print('No')
```