```
import os, re, sys
import numpy as np
import time, math, string
import matplotlib
from matplotlib import pyplot as plt

def index11664_fiducials ():
    road_list = [’Adams’,’Bush’,’Clinton’ ,’Dwight’ ,’Eisenhwr’ , ’Ford’ , ’Grant’ , ’Hoover’ , ’India’ ]
    cross_list = [’1st’ , ’2nd’ , ’3rd’ , ’4th’ , ’5th’ , ’6th’ , ’7th’ , ’8th’ ,’9th’ ]
    block_row_list = [’a’ ,’b’ ,’c’ ,’d’ ,’e’ ,’f’ ,’g’ ,’h’ ,’i’ ,’j’ ,’k’ ,’l’ ]
    block_col_list = [’a’ ,’b’ ,’c’ ,’d’ ,’e’ ,’f’ ,’g’ ,’h’ ,’i’ ,’j’ ,’k’ ,’l’ ]
    corners_list = []
    for road in road_list:
        for cross in cross_list:
            for r2 in block_row_list:
                for c2 in block_col_list:
                    addr = road[0] + cross[:-2] + ’_’ + r2 + c2
                    if r2+c2 in [’aa’ , ’la’ , ’ll’ ]:
                        corners_list.append(addr)
    fid_list = [\
    ’A1_ag’ , ’A2_ag’ , ’A3_ag’ , ’A4_ag’ , ’A5_ag’ , ’A6_ag’ , ’A7_ag’ , ’A8_ag’ ,’A9_ag’ , \
    ’A1_aj’ , ’A2_bj’ , ’A3_cj’ , ’A4_ak’ , ’A5_bk’ , ’A6_ck’ , ’A7_al’ , ’A8_bl’ ,’A9_cl’ , \
    ’B1_bg’ , ’B2_bg’ , ’B3_bg’ , ’B4_bg’ , ’B5_bg’ , ’B6_bg’ , ’B7_bg’ , ’B8_bg’ ,’B9_bg’ , \
    ’B1_aj’ , ’B2_bj’ , ’B3_cj’ , ’B4_ak’ , ’B5_bk’ , ’B6_ck’ , ’B7_al’ , ’B8_bl’ ,’B9_cl’ , \
    ’C1_cg’ , ’C2_cg’ , ’C3_cg’ , ’C4_cg’ , ’C5_cg’ , ’C6_cg’ , ’C7_cg’ , ’C8_cg’ ,’C9_cg’ , \
    ’C1_aj’ , ’C2_bj’ , ’C3_cj’ , ’C4_ak’ , ’C5_bk’ , ’C6_ck’ , ’C7_al’ , ’C8_bl’ ,’C9_cl’ , \
    ’D1_ah’ , ’D2_ah’ , ’D3_ah’ , ’D4_ah’ , ’D5_ah’ , ’D6_ah’ , ’D7_ah’ , ’D8_ah’ ,’D9_ah’ , \
    ’D1_aj’ , ’D2_bj’ , ’D3_cj’ , ’D4_ak’ , ’D5_bk’ , ’D6_ck’ , ’D7_al’ , ’D8_bl’ ,’D9_cl’ , \
    ’E1_bh’ , ’E2_bh’ , ’E3_bh’ , ’E4_bh’ , ’E5_bh’ , ’E6_bh’ , ’E7_bh’ , ’E8_bh’ ,’E9_bh’ , \
    ’E1_aj’ , ’E2_bj’ , ’E3_cj’ , ’E4_ak’ , ’E5_bk’ , ’E6_ck’ , ’E7_al’ , ’E8_bl’ ,’E9_cl’ , \
    ’F1_ch’ , ’F2_ch’ , ’F3_ch’ , ’F4_ch’ , ’F5_ch’ , ’F6_ch’ , ’F7_ch’ , ’F8_ch’ ,’F9_ch’ , \
    ’F1_aj’ , ’F2_bj’ , ’F3_cj’ , ’F4_ak’ , ’F5_bk’ , ’F6_ck’ , ’F7_al’ , ’F8_bl’ ,’F9_cl’ , \
    ’G1_ai’ , ’G2_ai’ , ’G3_ai’ , ’G4_ai’ , ’G5_ai’ , ’G6_ai’ , ’G7_ai’ , ’G8_ai’ ,’G9_ai’ , \
    ’G1_aj’ , ’G2_bj’ , ’G3_cj’ , ’G4_ak’ , ’G5_bk’ , ’G6_ck’ , ’G7_al’ , ’G8_bl’ ,’G9_cl’ , \
    ’H1_bi’ , ’H2_bi’ , ’H3_bi’ , ’H4_bi’ , ’H5_bi’ , ’H6_bi’ , ’H7_bi’ , ’H8_bi’ ,’H9_bi’ , \
    ’H1_aj’ , ’H2_bj’ , ’H3_cj’ , ’H4_ak’ , ’H5_bk’ , ’H6_ck’ , ’H7_al’ , ’H8_bl’ ,’H9_cl’ , \
    ’I1_ci’ , ’I2_ci’ , ’I3_ci’ , ’I4_ci’ , ’I5_ci’ , ’I6_ci’ , ’I7_ci’ , ’I8_ci’ ,’I9_ci’ , \
    ’I1_aj’ , ’I2_bj’ , ’I3_cj’ , ’I4_ak’ , ’I5_bk’ , ’I6_ck’ , ’I7_al’ , ’I8_bl’ ,’I9_cl’ ]
    fid_list = sorted(fid_list)
    corners_list = sorted(corners_list)
    return fid_list, corners_list

def hits_scrape (fid, diamond_dict):
    hits_dict = {}
    for i in range(11664):
        hits_dict[diamond_dict[i]] = 0
    f = open(fid)
    for line in f.readlines()[1:]:
        entry = line.split()
        i = int(entry[0])
        yesno = 1
        #yesno = int(entry[1])
        hits_dict[diamond_dict[i]] = yesno
    return hits_dict

# valid for data collection in June 2016
def collect_dicts ():
    road_list = [’A’ ,’B’ ,’C’ ,’D’ ,’E’ ,’F’ ,’G’ ,’H’ ,’I’ ]
    daor_list = [’I’ ,’H’ ,’G’ ,’F’ ,’E’ ,’D’ ,’C’ ,’B’ ,’A’ ]
    cros_list = [’1’ ,’2’ ,’3’ ,’4’ ,’5’ ,’6’ ,’7’ ,’8’ ,’9’ ]
    sorc_list = [’9’ ,’8’ ,’7’ ,’6’ ,’5’ ,’4’ ,’3’ ,’2’ ,’1’ ]
    wind_list = [’a’ ,’b’ ,’c’ ,’d’ ,’e’ ,’f’ ,’g’ ,’h’ ,’i’ ,’j’ ,’k’ ,’l’ ]
    dniw_list = [’l’ ,’k’ ,’j’ ,’i’ ,’h’ ,’g’ ,’f’ ,’e’ ,’d’ ,’c’ ,’b’ ,’a’ ]
    ordr_list = []
    addr_dict = {}
    ordr_dict = {}

    i = 0
    for c in range(9):
        for r in range(9):
            for wc in range(12):
                #print
                for wr in range(12):
                    if (c % 2 == 0):
                        if (wc % 2 == 0):
                            #addr = daor_list[r] + sorc_list[c] + ’_’ + dniw_list[wc] + dniw_list[wr]
                            addr = daor_list[r] + sorc_list[c] + ’_’ + dniw_list[wc] + dniw_list[wr]
                            ordr_list.append(addr)
                            #print addr,’1’,
                        else :
                            #addr = daor_list[r] + sorc_list[c] + ’_’ + dniw_list[wc] + wind_list[wr]
                            addr = daor_list[r] + sorc_list[c] + ’_’  + dniw_list[wc] + wind_list[wr]
                            ordr_list.append(addr)
                            #print addr,’2’,
                    else :
                        if  (wc % 2 == 0):
                            #addr = daor_list[r] + cros_list[c] + ’_’ + wind_list[wc] + dniw_list[wr]
                            addr = road_list[r] + sorc_list[c] + ’_’  + wind_list[wc] + dniw_list[wr]
                            ordr_list.append(addr)
                            #print addr,’3’,
                        else :
                            #addr = daor_list[r] + cros_list[c] + ’_’ + wind_list[wc] + wind_list[wr]
                            addr = road_list[r] + sorc_list[c] + ’_’  + wind_list[wc] + wind_list[wr]
                            ordr_list.append(addr)
                            #print addr,’4’,
                    addr_dict[addr] = i
                    ordr_dict[i] = addr
                    #print i,
                    i += 1
    return  addr_dict, ordr_dict

def normal_dicts ():
    road_list = [’A’ ,’B’ ,’C’ ,’D’ ,’E’ ,’F’ ,’G’ ,’H’ ,’I’ ]
    daor_list = [’I’ ,’H’ ,’G’ ,’F’ ,’E’ ,’D’ ,’C’ ,’B’ ,’A’ ]
    cros_list = [’1’ ,’2’ ,’3’ ,’4’ ,’5’ ,’6’ ,’7’ ,’8’ ,’9’ ]
    sorc_list = [’9’ ,’8’ ,’7’ ,’6’ ,’5’ ,’4’ ,’3’ ,’2’ ,’1’ ]
    wind_list = [’a’ ,’b’ ,’c’ ,’d’ ,’e’ ,’f’ ,’g’ ,’h’ ,’i’ ,’j’ ,’k’ ,’l’ ]
    dniw_list = [’l’ ,’k’ ,’j’ ,’i’ ,’h’ ,’g’ ,’f’ ,’e’ ,’d’ ,’c’ ,’b’ ,’a’ ]
    ordr_list = []
    ordr_dict = {}
    addr_dict = {}
    i = 0
    for  c  in  range(9):
        #print
        for  r  in  range(9):
            #print
            for  wc  in  range(12):
                #print
                for  wr  in  range(12):
                    if  (r % 2 == 0):
                        if  (wr % 2 == 0):
                            addr = road_list[r] + cros_list[c] + ’_’  + wind_list[wc] + wind_list[wr]
                            ordr_list.append(addr)
                            #print addr,
                        else :
                            addr = road_list[r] + cros_list[c] + ’_’  + wind_list[wc] + wind_list[wr]
                            ordr_list.append(addr)
                            #print addr,
                    else :
                        if  (wr % 2 == 0):
                            addr = road_list[r] + cros_list[c] + ’_’  + wind_list[wc] + wind_list[wr]
                            ordr_list.append(addr)
                            #print addr,
                        else :
                            addr = road_list[r] + cros_list[c] + ’_’  + wind_list[wc] + wind_list[wr]
                            ordr_list.append(addr)
                            #print addr,
                    ordr_dict[i] = addr
                    addr_dict[addr] = i
                    #print i,
                    i += 1
    return  addr_dict, ordr_dict

def get_xy (xtal_name):
    w2w = 0.125
    b2b_horz = 0.825
    b2b_vert = 1.125
    #b2b_horz = 0
    #b2b_vert = 0
    cell_format = [9, 9, 12, 12]
    entry = xtal_name.split(’_’ )[-2:]
    R, C = entry[0][0], entry[0][1]
    r2, c2 = entry[1][0], entry[1][1]
    blockR = int(string.uppercase.index(R))
    blockC = int(C) - 1
    windowR = string.lowercase.index(r2)
    windowC = string.lowercase.index(c2)
    x = (blockC * b2b_horz) + (blockC * (11) * w2w) + (windowC * w2w)
    y = (blockR * b2b_vert) + (blockR * (11) * w2w) + (windowR * w2w)
    return x, y

def main():
    x_list, y_list, z_list = [], [], []
    # [addr] = i, [i] = addr
    normal_addr_dict, normal_ordr_dict = normal_dicts()
    fid_list, corners_list = index11664_fiducials()
    for i in sorted(normal_ordr_dict.keys()):
        addr = normal_ordr_dict[i]
        x, y = get_xy(addr)
        if addr in corners_list:
            z = 2
        elif addr in fid_list:
            z = 7
        else:
            z = 4
        x_list.append(float(x))
        y_list.append(float(y))
        z_list.append(float(z))

    X = np.array(x_list)
    Y = np.array(y_list)
    Z = np.array(z_list)
    xr = X.ravel()
    yr = Y.ravel()
    zr = Z.ravel()

    fig = plt.figure(num=None, figsize=(9,9), facecolor='0.6', edgecolor='k')
    fig.subplots_adjust(left=0.03,bottom=0.03,right=0.97,top=0.97,wspace=0,hspace=0)
    ax1 = fig.add_subplot(111, aspect='equal', axisbg='0.7')
    ax1.scatter(xr, yr, c=zr, s=14, alpha=1, marker='s', linewidth=0.1, cmap='PuOr')
    ax1.set_xticks([2.2*x for x in range(11)])
    ax1.set_yticks([2.5*x for x in range(11)])
    ax1.set_xlim(xr.min()-0.2, xr.max()+0.2)
    ax1.set_ylim(yr.min()-0.2, yr.max()+0.2)
    ax1.invert_yaxis()
    plt.savefig('chip_image.png', dpi=600, bbox_inches='tight', pad_inches=0.05)

    return X, Y, Z

if __name__ == '__main__':
    main()
    plt.show()
plt.close()
```