*def* load\_profile\_n\_household():

        pv\_params = {"directory\_data": "brussels", "weather\_file\_name":"brussels\_50.8444\_4.35609\_msg-iodc\_60\_", "directory\_output" :  "pv\_example\_2", "n\_years" : 3, "begin\_year" : 2017, "end\_year" : 2019,

                "n\_households" : 8, "key" : "hybrid", "PV\_inclination": [20], "PV\_orientation" : [200], "PV\_area" : [1], "PV\_efficiency" : 0.182, "PV\_module\_size": [1.99, 0.991, 0.0075],

                "PV\_NOCT" : 43.6, "PV\_betacoeff": 0.0034, "PV\_Tref" : 25, "sharing\_price" : 0.2, "grid\_injection\_price" : 0.04, "investment\_cost" : 126792, "estimated\_lifetime" : 25, "interest\_rate" : 0.03,

                "battery": True, "battery\_capacity": 10000, "battery\_efficiency": 0.9, "battery\_charging\_power": 5000, "battery\_discharging\_power": 5000

                }

        #"battery": True, "battery\_capacity": 10000, "battery\_efficiency": 0.9, "battery\_charging\_power": 5000, "battery\_discharging\_power": 5000,

        #"EV\_charger" : True, "EV\_price" : 0.45

        #"EV\_charger" : True, "EV\_price" : 0.3, 'EV\_file' : 'C:/Users/simva/OneDrive/Documents/1 Master 2/Mémoire/code/memoire\_partage\_elec/ev\_charging\_profile/data\_2023\_selected.csv'

        enercom = EnergyCommunity(pv\_params)

        enercom.get\_weather\_data()

        enercom.func\_compute\_total\_production()

        file\_save = "C:/Users/simva/OneDrive/Documents/1 Master 2/Mémoire/code/memoire\_partage\_elec/new/production\_profile"

        np.savetxt(file\_save, enercom.total\_production)

        print("Production profile saved to", file\_save)

        for i in range(1,50):

                n\_households = i

                input\_directory = "pv\_example\_2"

                output\_directory = "example\_2"

                cooking\_params = ["low", "medium", "medium", "high", "high", "high", "low", "low", "high", "medium", "medium", "medium", "high", "low", "low", "medium", "low", "medium", "high", "high", "high", "low", "medium", "medium" ]

                wh\_capacity\_params = ["low", "medium", "medium", "high", "high", "medium", "low", "low", "medium", "low", "medium", "medium", "high", "low", "medium", "low", "medium", "medium", "medium", "medium", "high", "low", "low", "medium"]

                n\_cold\_source\_params = [1, 1, 1, 2, 2, 2, 1, 1, 2, 1, 1, 2, 2, 1, 1, 1, 2,2,2,2,2,1,1,2]

                wm\_frequency\_params = ["medium", "medium", "medium", "medium", "high", "medium", "low", "low", "medium", "low", "medium", "medium", "high", "low", "low", "low", "medium", "high", "medium", "high", "high", "low", "low", "medium"]

                have\_dryer\_params = [False, False, True, False, True, False, False, False, False, False, False, False, True, False, False, False, False, False, False, True, True, False, False, False]

                dryer\_type\_params = [None, None, "condensation", None, "evacuation", None, None, None, None, None, None, None, "heat\_pump", None, None, None, None, None, None, "evacuation", "condensation", None, None, None]

                dryer\_frequency\_params = [None, None, "medium", None, "medium", None, None, None, None, None, None, None, "low", None, None, None, None, None, None, "medium", "low", None, None, None]

                have\_dw\_params = [False, True, True, True, True, True, False, False, True, False, True, True, True, False, False, False, False, True, True, True, True, False, False, False]

                dw\_frequency\_params = [None, "low", "medium", "medium", "high", "high", None, None, "medium", None, "low", "high", "medium", None, None, None, None, "medium", "low", "medium", "low", None, None, None]

                grid\_price\_day\_params=[0.36,0.39,0.39,0.39,0.36,0.36,0.36,0.36,0.36,0.39,0.39,0.39,0.36,0.36,0.36,0.36,0.36,0.39,0.39,0.39,0.36,0.36,0.36,0.36]

                grid\_price\_night\_params=[0.29,0.30,0.30,0.30,0.33,0.33,0.29,0.29,0.29,0.30,0.30,0.30,0.33,0.33,0.29,0.29,0.29,0.30,0.30,0.30,0.33,0.33,0.29,0.29,0.29]

                heating\_is\_elec\_params = [True]\*n\_households

                T\_ext\_th\_params = [12,9,14,11,13,13,11,13,12,10,11,9,13,15,12,12,12,11,13,14,9,9,14,12]

                T\_ext\_th\_night\_params = [7,5,9,9,5,7,6,8,7,8,7,6,5,9,10,7,7,6,8,8,4,5,10,7]

                annual\_heating\_value\_params = [95] \* n\_households

                #PEB\_params = ["A","A", "A", "B", "A", "B", "A", "B", "B","B", "A", "B", "A", "A", "B", "B", "A","B", "A", "A", "A","B", "A","B"]

                heating\_eff\_params = [3]\*n\_households

                flat\_area\_params = [60,100,120,150,250,120,60,80,100,80,120,150,250,66,100,75,110,120,120,150,250,66,80,90]

                #wh\_night = [False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False]

                wh\_night = [False]\*n\_households

                wh\_intelligence\_params = True

                #wh\_hour\_mode = "perfect\_knowledge"

                wh\_hour\_mode = "fixed"

                if wh\_hour\_mode == "perfect\_knowledge":

                        wh\_multiyears\_params = [True]\*n\_households

                else:

                        wh\_multiyears\_params = [False]\*n\_households

                params = {

                        "input\_directory": input\_directory,

                        "output\_directory": output\_directory,

                        "n\_households": n\_households,

                        "cooking\_params": cooking\_params,

                        "wh\_usage\_params": wh\_capacity\_params,

                        "n\_cold\_source\_params": n\_cold\_source\_params,

                        "wm\_frequency\_params": wm\_frequency\_params,

                        "have\_dryer\_params": have\_dryer\_params,

                        "dryer\_type\_params": dryer\_type\_params,

                        "dryer\_frequency\_params": dryer\_frequency\_params,

                        "have\_dw\_params": have\_dw\_params,

                        "dw\_frequency\_params": dw\_frequency\_params,

                        "grid\_price\_day\_params": grid\_price\_day\_params,

                        "grid\_price\_night\_params": grid\_price\_night\_params,

                        "wh\_intelligence\_params": wh\_intelligence\_params,

                        "heating\_is\_elec\_params": heating\_is\_elec\_params,

                        "T\_ext\_th\_params": T\_ext\_th\_params,

                        "T\_ext\_th\_night\_params": T\_ext\_th\_night\_params,

                        "annual\_heating\_value\_params": annual\_heating\_value\_params,

                        #"PEB\_params": PEB\_params,

                        "heating\_eff\_params": heating\_eff\_params,

                        "flat\_area\_params": flat\_area\_params,

                        "wh\_night\_params": wh\_night,

                        "wh\_hour\_mode": wh\_hour\_mode,

                        "wh\_multiyears\_params": wh\_multiyears\_params,

                        "wh\_intelligence\_params" : wh\_intelligence\_params,

                }

                enercom = EnergyCommunity(pv\_params)

                #enercom.get\_weather\_data()

                #enercom.func\_compute\_total\_production()

                #enercom.save\_production()

                multi = MultiHousehold(params, enercom)

                multi.run()

                total\_consumption = multi.total\_electric\_consumption

                file = *f*"C:/Users/simva/OneDrive/Documents/1 Master 2/Mémoire/code/memoire\_partage\_elec/new/load\_profile\_simu/{n\_households}\_households.csv"

                np.savetxt(file, total\_consumption, *delimiter*=",", *fmt*='%.2f')

                print(*f*"Load profile for {n\_households} households saved to {file}")

load\_profile\_n\_household()