A.1)

['2018', '0', '347749', '364155', '711904']

['2017', '1', '355472', '370453', '725925']

['2016', '2', '363162', '378518', '741680']

['2015', '3', '372402', '387906', '760308']

['2014', '4', '387042', '399232', '786274']

['2013', '5', '389920', '407611', '797531']

['2012', '6', '396835', '417471', '814306']

['2011', '7', '403349', '418623', '821972']

['2010', '8', '412555', '429919', '842474']

['2009', '9', '408232', '427917', '836149']

['2008', '10', '410703', '430934', '841637']

A.2)

'\n' permet de faire un retour à la ligne console.

B.1)

Ce programme permet d'afficher le tableau dans la console ligne par ligne avec la bonne séparation des colonnes et des lignes dans la variable 'ligne' en une chaine de caractère.

B.2)

for ligne in data:

    ligne = ligne.strip().split(sep) #on sépare avec le caractère séparateur sep

    year.append(int(ligne[0]))

print (year)

a)) le rôle de 'int' est de convertir 'ligne' qui est de type 'string' en 'int' pour être cohérent est pouvoir plus tard plus facilement exploiter la liste 'year'.

b)) La méthode strip () renvoie une copie de la chaîne en supprimant les caractères de début et de fin (en fonction de l'argument de chaîne passé).

La méthode append () ajoute un élément à la fin de la liste.

B.4)

'year'

[2018, 2017, 2016, 2015, 2014, 2013, 2012, 2011, 2010, 2009, 2008, 2007, 2006, 2005, 2004, 2003, 2002, 2001, 2000, 1999, 1998, 1997, 1996, 1995, 1994, 1993, 1992, 1991, 1990, 1989, 1988, 1987, 1986, 1985, 1984, 1983, 1982, 1981, 1980, 1979, 1978, 1977, 1976, 1975, 1974, 1973, 1972, 1971, 1970, 1969, 1968, 1967, 1966, 1965, 1964, 1963, 1962, 1961, 1960, 1959, 1958, 1957, 1956, 1955, 1954, 1953, 1952, 1951, 1950, 1949, 1948, 1947, 1946, 1945, 1944, 1943, 1942, 1941, 1940, 1939, 1938, 1937, 1936, 1935, 1934, 1933, 1932, 1931, 1930, 1929, 1928, 1927, 1926, 1925, 1924, 1923, 1922, 1921, 1920, 1919, 1918]

'age'

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100]

'popu'

[711904, 725925, 741680, 760308, 786274, 797531, 814306, 821972, 842474, 836149, 841637, 834910, 848353, 829359, 826638, 824659, 827342, 840123, 845937, 800716, 783525, 754642, 756046, 739682, 716463, 714161, 749884, 763138, 781915, 790072, 800741, 806741, 820146, 819576, 814250, 805347, 855167, 864189, 875950, 830022, 813357, 818487, 794791, 818068, 861330, 908597, 929413, 924284, 906593, 894451, 885443, 880170, 899644, 902138, 909113, 898554, 867146, 866057, 861040, 852678, 833677, 826729, 818238, 806489, 803021, 784593, 792788, 772744, 796836, 777985, 770557, 751055, 705646, 529118, 513167, 494088, 453731, 399713, 410469, 421613, 404517, 387057, 375998, 352943, 342852, 312308, 300889, 270976, 248176, 205828, 179704, 150253, 125591, 103764, 81760, 62875, 47983, 34993, 24790, 10331, 15646]

C.1

Le programme renvoie la population totale de chaque année de 2018 à 2014.

C.2/3

print("--> popu d'age entre 10 et 18 <--")

for i in age:

    if age[i] <= 10 or age[i] <= 18:

        print(popu[i])

print("--> popu total <--")

for i in range(len(popu)):

    print(popu[i])

C.4

def popentreage(age1, age2):

    returnVal = []

    for i in age:

        if age[i] <= age1 or age[i] <= age2:

            returnVal.append(popu[i])

    return returnVal

print("--> fx popentreage <--")

print(popentreage(10, 18))

C.5

Le programme revoie dans une liste les 5 premières valeurs (l'age) qu'il trouve dans la liste 'age'.

C.6

print("--> tabR <--")

tabR = []

for i in range(len(popu)):

    if popu[i] >= 830000:

        if age[i] <= 10 or age[i] <= 18:

            tabR.append(age[i])

print(tabR)

C.7

print("--> tabR 2 <--")

tabR2 = []

for i in range(len(popu)):

    if popu[i] >= 700000:

        tabR2.append(age[i])

print(tabR2)

print("--> tabR 3 <--")

tabR3 = []

for i in range(len(popu)):

    if popu[i] >= 900000:

        tabR3.append(year[i])

print(tabR3)

D.1 moyenne des âges qui compte plus de 900000 personnes.

moyenne = 0

for i in range(len(tabR2)):

    moyenne += age[tabR2[i]]

moyenne /= (len(tabR2))

print("moyenn -->", moyenne)

D.2 années de naissance où la population est de moins de 100000 personnes et tranche d'âge

ado = 0

active = 0

nonActive = 0

tabR4 = []

for i in range(len(popu)):

    if popu[i] <= 100000 :

        tabR4.append(year[i])

print(tabR4)

for i in range(len(tabR4)):

    if tabR4[i] > 2004:

        ado += 1

    elif tabR4[i] <= 2004 and tabR4[i] >= 1955:

        active += 1

    elif tabR[4] < 1955:

        nonActive += 1

print("ado =", ado/len(tabR4)\*100, "%", "active =", active/len(tabR4)\*100, "%", "non active", nonActive/len(tabR4)\*100, "%")

D.3

index = 0

popuMax = 0

for i in range(len(year)):

    if age[i] >= 15 and age[i] <= 25:

        if popu[i] > popuMax:

            popuMax = popu[i]

            index = i

print(age[index], "ans, population maximale de cet âge pour la tranche d'âge 15-25 ans ->", popu[index])

D.4

trent = 0

quadra = 0

quinqua = 0

for i in range(len(year)):

    if year[i] < 1989 and year[i] >= 1979:

        trent += popu[i]

    if year[i] < 1979 and year[i] >= 1969:

        quadra += popu[i]

    if year[i] < 1969 and year[i] >= 1959:

        quinqua += popu[i]

print("pop trentenaires-->", trent, "; pop quadragénaires-->", quadra, "; pop quinquagénaires-->", quinqua)

D.5

print("--> occurence <--")

def occurence(population):

    nb = 0

    for i in range(len(popu)):

        if popu[i] >= population:

            nb += 1

    return nb

print("occurence -->", occurence(800000))   #renvoie : 48 années