In [19]:

```
# function to genreate all leap year in a given time period
# 2000 - 2020 -> 2000 2004 2008 2012 2016 2020

def leapyear(a): # to check if a given year is a leap year
    if a%4 == 0 or (a%400 == 0 and a%100!=0):
        return True
    return False

#leapyear(2000,2020)
def generateLeapYear(lb,ub):
    for a in range(lb,ub+1):
        if leapyear(a):
            print(a, end=" ")
    return

generateLeapYear(1919,2019)

```

```
1920 1924 1928 1932 1936 1940 1944 1948 1952 1956 1960 1964 1968 1972 1976 1980
1984 1988 1992 1996 2000 2004 2008 2012 2016
```

In [26]:

```
# calculate days in given time range using leap year
# for every year in the given time period, if the year is not a leap yearb
def days(lb,ub):
    sum = 0
    for a in range(lb,ub):
        if leapyear(a):
            sum = sum + 366
        else:
            sum =  sum + 365
    return sum

days(2000,2020)
```

Out[26]:

```
7305
```

In [13]:

```
# function to calculate number of hours for given period
# (11,1975,3,1999)
# [all days from feb 2016 to dec 2016,
#   all days for years between 2016 and 2019,
#    all day from jan to june 2019]
# no of hours = 24 * no of days
# 3 steps
# step 1 : start month year to endof year -  calculate no of days
# step 2: calculate days for all years between start year ans end year exclu
                # 2017, 20108 - 365n * no of years
# step 3: calculate the days from jan to end of the month year
# first six month - 1,3,4,6,7
                    # all odd month have 31 days
                    # all even month have 30 days
# last six months - 8,9,10,11,12
                    # all odd month have 31 days
                    # all even month have 30 days
    #31 days -(month <= 7 and month % 2 != 0  and month != 2)|| (month)
    # return 31
    # else
# return 30

def numberofdaymonth(month,year):
    if month == 2:
        if leapyear(a):
            return 29
        return 28
    elif (month <= 7 and month % 2 != 0) or (month >= 8 and month % 2 == 0)
            return 31
    else:
        return 30

#numberofdaysmonth(4,2019)

def daysinstartyear(startmonth,startyear):
    days = 0
    for month in range(startmonth, 13):
        days += numberofdaysmonth(month,startyear)
    return

# daysinstartyear(6,2019)
def daysinendyear(endmonth, endyear):
    days = 0
    for month in range(1,endmonth+1):
        day += numberofdaysmonth(month, endyear)
    return days
def numberofhours(startmonth,  startyear, endmonth, endyear):
    days = 0
    if startyear != endyear:
        days += daysinstartyear(startmonth, startyear)
        days += daysinendyear(endmonth,endyear)
        if endyear - startyear == 2:
            days += numberofdays(startyear + 1, startyear + 1)
        elif endyear - startyear > 2:
            days += numberifdays(startyear + 1, endyear - 1)
        else:
            for month in range(startmonth,endmonth)
            days += numberofdaysmonth(month,startyear)
        return 24 * days

numberofhours(11,1975,3,1999)
```

```
  File "<ipython-input-13-7275aa7672a7>", line 28
    elif (month <= 7 and month % 2 != 0) or (month >= 8 and month % 2 == 0)
                                                                           ^
SyntaxError: invalid syntax
```

In [ ]:

In [ ]: