```
Input data from the user:

Name1 = name of the 1st person
Brithdate1 = date of birth of person 1 (written in 8 digits YYYYMMDD)

Name2 = name of the 2nd person
Brithdate2 = date of birth of person 2 (written in 8 digits YYYYMMDD)

Separate birthdate data by year, month and day

Get the last two digits of birthdate1 using the modulus of 100

day1 = birthdate1 % 100

Get two middle digits by first removing the last 2 digits using
integer division and the first 4 digits using the modulus operator

month1 = birthdate // 100 % 100

Get the year by using integer division to get rid of the month and
day digits

year1 = birthdate1 // 1000

Repeat process for birthdate 2

day2 = birthdate2 % 100
month2 = birthdate2 // 100 % 100
year2 = birthdate2 // 1000

Print name1 and name2's date of birth

Check who is older

If name1 and name2 were born in the same year

     If name1 and name 2 were born in the same month

          If name1 and name2 were born on the same day
               Print that they are the same age

          If name1 was born a day before
               Print that name1 is older

          Else name2 is older
```

```
      Else if name1 was born a month before name2
            Print that name1 is older

      Else name2 is older than name1

Else if name1 was born a year before year2
      Print that name1 is older

Else name2 is older
```

Pseudocode · Scratch work that got me to the pseudocode:

Input data from the user

name 1
birthdate 1 (has to be an int)

name 2
birthdate 2 (must be changed to int)

separate birthdate data by year, month & day

Ex. 1964|04|05 ⟶ Similar to class exercise of inverting 4 digit numbers
need to separate first 4 numbers

day = 19640405 % 100 = 5 ⟶ gets me the last 2 numbers (day: 05) which is the remainder of dividing date by 100

Thus day = birthdate# % 100 (where birthday# is any variable with birthdate)

year = 19640405 // 10000 = 1964 ⟶ gets the year since // gives an integer
4

Thus year = birthdate # // 1000

month = 19640405 // 100 % 100 = 4 ⟶ get the month

↳ 1964|04|05 probably similar to getting middle numbers in reverse.py exercise
↳ need to get this ↑

↳ removed last two digits

lets say 1964 0405 // 100 = 196404 % 100 = 04 ⟶ removed first 4 digits
remove remove
↳ could be 2 digits so it can't be % 10
ex december would return as 2 instead of 12
gives me the middle 2 numbers

Thus month = birthdate // 100 % 100

```
if ( year1 < year2 )
        name1 is older than name2
else if (year2 < year 1)
        name 2 is older than name1
else if (name1 == name 2)
        if (month 1 < month 2)
                name1 is older than name2
        else if (month 2 < month 1)
                name 2 is older than name 1
        else if (month 1 == month 2)
                if (day 1 < day 2)
                        name 1 is older than name 2
                else if (day 2 < day 1)
                        name 2 is older than name 1
                else if (day 1 == day 2)
                        name 1 & name 2 are the same age
```

Flip the logic so the
- first if checks if year, date and month are the same,
- second if checks if name1 was born before else name2 would be older