In [11]:
```
#csv is used to import and export the database so that improt the csv modules for
#--analysing the database
import csv
import os#import the os module to open the file and get the path of the file, etc
```

In [12]:
```
def main():
#call put all functions inorder to obtain the users' criteria to fill out the criteria of
#---the patients
    print_the_header() # haeder function
    filename = get_filename() #function for get the name of the file
    #function to select the max/min range of the age as a requirement
    age_min, age_max = get_age_range()
    #function to select the max/min range of the infection length as a requirement
    small_inf, big_inf = get_inf_length()
    pat_sex = get_pet_sex()#function to select the gender as a requirement
    pat_coin = get_pat_coin()#function to select if the pat has coinfection as a requirement
    pat_ther = get_pat_ther()#function to select if the pat has therapy as a requirement

    out_filename = filename + '.valid' #saving the output file as a same filename with .valid
    num_pats = 0

    # Open file and start reader
    with open(filename) as handle:
        reader = csv.DictReader(handle) # use the dictreader to read the handle

        # open file base on the out_filename and allow to write the file
        with open(out_filename, mode='w') as out_handle:
            fields = ['PAT_NUM', 'SEX', 'AGE', 'INFECTION_LENGTH', 'ON_THERAPY', 'COINFECTION']
            writer = csv.DictWriter(out_handle, fields)
            writer.writeheader()

# Since the homework requirement ask to record all patients that match the criteria and
#---save them into a new file, --- I use the for and several if loop to find that the
#---patients from inclusion criteria (more easier to output the list)
```

```
# filter out the if the patients is match with the criteria and save t
hem in a new file
            for row in reader:
                # identify and split each item's meaning
                pat_age = int(row['AGE'])
                pat_sex_col = str(row['SEX'])
                pat_therapy = str(row['ON_THERAPY'])
                pat_coinfection = str(row['COINFECTION'])
                inf_length = int(row['INFECTION_LENGTH'])

                # build up the filter for separating out the long logi
c for clarity
                # age and INFECTION_LENGTH are numbers which also cont
ain within a range
                match_age = (pat_age > age_min) and (pat_age < age_max
)
                match_inf_length = (inf_length > small_inf) and (inf_l
ength < big_inf)

                # if the criteria is match, print them out SAVE in a s
plit file
                # chcking the gender, if it is match with the requirem
ent
                match_sex = True
                if pat_sex != '0':
                    match_sex = (pat_sex_col == pat_sex)

                 # chcking the coninfectuin, if it is match with the r
equirement
                match_coin = True
                if pat_coin!= '0':
                    match_coin = (pat_coinfection == pat_coin)


                 # chcking the therapy, if it is match with the requir
ement
                match_therapy = True
                if pat_ther !='0':
                    match_therapy = (pat_therapy == pat_ther)


                # if all the requirements are match, record into the f
ile
                if match_age and match_sex and match_coin and match_th
erapy and match_inf_length:
                    # count the total number of patients from inclusio
n criteria
                    num_pats += 1
```

```
                    # write the patients that pass filter into the new
file
                    writer.writerow(row)

# print the criteria based on the following format
    print('Based on the following criteria:')
    print(' - Age: [%i, %i]' % (age_min, age_max))
    print(' - Infection Length: [%i, %i]' % (small_inf, big_inf))
    print(' - Gender %s', pat_sex)
    print(' - Coinfection %s', pat_coin)
    print(' - Therapy %s', pat_ther)
    print('There are %i patients from the inclusion criteria' % num_pa
ts)
```

In [13]:
```
def print_the_header(): # print the header of the entire progarm
    print('---------------------------------')
    print('     Process Demographics     ')
    print('---------------------------------')
```

In [14]:
```
# creating a function to obatin the filename
def get_filename():

    # if there is none filename, ask the user to type one file name
    # if the input of the user is not exist, asking he/she to try agai
n
    filename = None
    while filename is None:

        filename = input('What is the /path/to/the/file? ')

        # Check if the filename exists.
        if not os.path.exists(filename):
            print('That file could not be found. Please try again.')
            filename = None

    return filename
```

In [15]:
```
# asking the user if the patient's age is relative to the study, if ye
s asking the user to input patients' age for filtering
def get_age_range():
    while True:
        age_input = input("If the patient's age range relative to the
study: [Y]es or [N]o:")
        age_input = age_input.lower().strip()  # changing the uppercas
e of the input as the lowercase for decrease the error

        # if the patient's age is relative to the study
        if age_input == "y": # 'y' means yes which is the age is relat
```

```
ive to the study
            age_min, age_max = None, None

            while age_min is None: # if there is no min age, asking th
e user to type it
                age_inp = input('What is the youngest age for the stud
y? ')
                try:
                    age_min = int(age_inp)

                # if the user not input a number, asked he/she to re-t
ype the age
                except ValueError:
                    print(age_inp + ' is not a number. Please try agai
n')
                    continue

                if age_min < 18: # check if the min age is smaller tha
n 18, if so asking the user to re-select one
                    print('Ethics boards require special permission fo
r youth cohort. Please pick an older age')
                    age_min = None

            while age_max is None: # if there is no max age, asking th
e user to type it
                age_inp = input('What is the oldest age for the study?
')
                try:
                    age_max = int(age_inp)

                # if the user not input a number, asked he/she to re-t
ype the age
                except ValueError:
                    print(age_inp + ' is not a number. Please try agai
n')
                    continue
            return age_min, age_max
        # if the patient's age is not relative to the study, then brea
k.
        elif age_input == "n": # 'n' means No which is the age is NOT
relative to the study
            age_min = 0 # just randomly guess the min age
            age_max = 100 # just randomly guess the max age
            return age_min, age_max
        else:
            print('Incorrect input. Please try again')
            continue
```

In [16]:

```
# asking the user if the patient's infection length is relative to the
```

```
study,
#---if yes asking the user to input patient's infection length (range)
for filtering
def get_inf_length():
    while True:
        inf_input = input("If the patient's infection length relative
to the study: [Y]es or [N]o: ")
        inf_input = inf_input.lower().strip()  # changing the uppercas
e of the input as the lowercase for decrease the error

        # if the patient's infection length is relative to the study
        if inf_input == "y":# 'y' means yes which is the infection len
gth is relative to the study
            small_inf, big_inf = None, None

            while small_inf is None:
                small_inf_inp = input('The smallest or minimum infecti
on length for the patient is: ')
                try:
                    small_inf = int(small_inf_inp)

                # if the user not input a number, asked he/she to re-t
ype the infection length
                except ValueError:
                    print(small_inf_inp + ' is not a number. Please tr
y again.')
                    continue

            while big_inf is None:
                big_inf_inp = input('The biggest or maximum infection
length for the patient is')
                try:
                    big_inf = int(big_inf_inp)

                # if the user not input a number, asked he/she to re-t
ype the infection length
                except ValueError:
                    print(big_inf_inp + ' is not a number. Please try
again.')
                    continue

            return small_inf,big_inf

        # if the patient's infection length is not relative to the stu
dy, then break.
        elif inf_input == "n":# 'n' means No which is the infection le
ngth is NOT relative to the study
            small_inf = 0 # just randomly guess the infection length
            big_inf = 100 # just randomly guess the infection length
            return small_inf, big_inf
```

```
        else:
            print('Incorrect input. Please try again')
            continue
```

In [17]:

```
# asking the user if the patient's gender is relative to the study, if yes asking the user to input patients' gender for filtering
def get_pet_sex():
    while True:
        gender_input = input("If the patient's gender relative to the study: [Y]es or [N]o: ")
        gender_input = gender_input.lower().strip()  # changing the up percase of the input as the lowercase for decrease the error

        # if the patient's gender is relative to the study
        if gender_input == "y":# 'y' means yes which is the gender is relative to the study
            pat_sex = None

            while pat_sex is None:
                gender= input('What is the gender for the study? [M]al e or [F]emale:')
                gender = gender.lower().strip()  # changing the upperc ase of the gender as the lowercase for decrease the error

                try:
                    gender=str(gender)
                # if the user not input a correct word, asked he / she to re-input the gender
                except ValueError:
                    print(gender + ' cannot be found in the system. Pl ease try again')
                    continue

                if gender=='m':
                    pat_sex='Male'
                if gender=='f':
                    pat_sex='Female'

            return pat_sex
        # if the patient's gender is not relative to the study, then b reak.
        elif gender_input == "n": # 'n' means No which is the gender i s NOT relative to the study
            break
        else:
            print('Incorrect input. Please try again')
            continue
```

In [18]:
```
# asking the user if the patient has coinfection will relative to the
study, if yes asking the user to input patients' coinfection  for filt
ering
def get_pat_coin():
    while True:
        coin_input = input("If the patient has coinfection will relati
ve to the study: [Y]es or [N]o: ")
        coin_input = coin_input.lower().strip()  # changing the upperc
ase of the input as the lowercase for decrease the error

        # if the patient has coinfection is relative to the study
        if coin_input == "y":# 'y' means YES which is the coinfection
is relative to the study
            pat_coin = None

            while pat_coin is None:
                coinfection = input('Does the patient has coinfection
in this study: [Y]es or [N]o :')
                coinfection = coinfection.lower().strip()  # changing
the uppercase of the gender as the lowercase for decrease the error

                try:
                    coinfection = str(coinfection)
                # if the user input a incorrect choice, ask he / she t
o re-input the coinfection
                except ValueError:
                    print(coinfection + ' cannot identified. Please tr
y again')
                    continue

                if coinfection == 'n':
                    pat_coin = 'No'
                if coinfection == 'y':
                    pat_coin = 'Yes'

            return pat_coin
        # if the patient has coinfection is not relative to the study,
then break.
        elif coin_input == "n":# 'n' means No which is the coinfection
is NOT relative to the study
            break
        else:
            print('Incorrect input. Please try again')
            continue
```

In [19]:
```
# asking the user if the patient on therapy that relative to the study
, if yes asking the user to input if the patient is on therapy for fil
tering
def get_pat_ther():
    while True:
        ther_input = input("If the patient on therapy will relative to
the study: [Y]es or [N]o: ")
        ther_input = ther_input.lower().strip()  # changing the upperc
ase of the input as the lowercase for decrease the error

        # if the patient on therapy is relative to the study
        if ther_input == "y": # 'y' means YES which is the therapy is
relative to the study
            pat_ther = None

            while pat_ther is None:
                therapy= input('Does the patient has the therapy in th
is study: [Y]es or [N]o: ')
                therapy = therapy.lower().strip()  # changing the uppe
rcase of the gender as the lowercase for decrease the error

                try:
                    therapy=str(therapy)
                # if the user input a incorrect choice, ask he / she t
o re-input
                except ValueError:
                    print(therapy+' is not one of the choices. Please
type in a credible answer')
                    continue

                if therapy=='n':
                    pat_ther='No'
                if therapy=='y':
                    pat_ther='Yes'
            return pat_ther
        # if the patient on therapy is not relative to the study, then
break.
        elif ther_input == "n": # 'n' means No which is the therapy is
NOT relative to the study
            break
        else:
            print('Incorrect input. Please try again')
            continue


if __name__ == '__main__':
    main()
```

```
---------------------------------
     Process Demographics
---------------------------------
What is the /path/to/the/file? pat_data.csv
If the patient's age range relative to the study: [Y]es or [N]o:Y
What is the youngest age for the study? 18
What is the oldest age for the study? 60
If the patient's infection length relative to the study: [Y]es or [N
]o: n
If the patient's gender relative to the study: [Y]es or [N]o: y
What is the gender for the study? [M]ale or [F]emale:f
If the patient has coinfection will relative to the study: [Y]es or
[N]o: y
Does the patient has coinfection in this study: [Y]es or [N]o :y
If the patient on therapy will relative to the study: [Y]es or [N]o:
y
Does the patient has the therapy in this study: [Y]es or [N]o: y
Based on the following criteria:
 - Age: [18, 60]
 - Infection Length: [0, 100]
 - Gender %s Female
 - Coinfection %s Yes
 - Therapy %s Yes
There are 1564 patients from the inclusion criteria
```

In [ ]: