In [ ]:
```
import os
import numpy as np
from glob import glob
import re
import time
import shutil
from tqdm import tqdm
```

In [ ]:
```
centre = 'rccc'

number_of_weeks_to_keep = 25
```

In [ ]:
```
directory_map = {
    'rccc': {
        'search_string': r'\\monacoda\FocalData\RCCC\1~Clinical\*~*\demographic.*',
        'holding': r'\\monacoda\FocalData\ToBeArchived',
        'archived': r'\\UTILSVR\PhysBack\MONACO_ARCHIVE_1'
    },
    'nbcc': {
        'search_string': r'\\nbccc-monaco\Users\Public\Documents\CMS\FocalData\NBC
        'holding': r'\\nbccc-monaco\Users\Public\Documents\HoldingDirectory',
        'archived': r'\\nbccc-monaco\Archive\Patients'
    }
}
```

In [ ]:
```
patient_demographic_files = np.array(glob(directory_map[centre]['search_string']))
patient_demographic_files
```

In [ ]:
```
patient_directories = np.array([
    os.path.dirname(item)
    for item in patient_demographic_files])
patient_directories
```

In [ ]:
```
corresponding_number_of_weeks_ago = []

for current_patient_directory in patient_directories
    files_and_folders_one_level = glob(os.path.join(current_patient_directory, '*')
    all_date_modified = np.array([
        os.path.getmtime(item) for item in files_and_folders_one_level])

    number_of_weeks_ago = (time.time() - all_date_modified) / (60 * 60 * 24 * 7)

    minimum_number_of_weeks_ago = np.min(number_of_weeks_ago)

    corresponding_number_of_weeks_ago.append(minimum_number_of_weeks_ago)
```

In [ ]:
```
corresponding_number_of_weeks_ago = np.array(corresponding_number_of_weeks_ago)
corresponding_number_of_weeks_ago
```

In [ ]:
```
archive_reference = corresponding_number_of_weeks_ago > number_of_weeks_to_keep
directories_to_be_archived = patient_directories[archive_reference]
directories_to_be_archived
```

In [ ]:
```
corresponding_number_of_weeks_ago[archive_reference]
```

In [ ]:
```
len(corresponding_number_of_weeks_ago[archive_reference])
```

In [ ]:
```
patient_folder_name = [
    os.path.basename(item) for item in directories_to_be_archived]
patient_folder_name
```

In [ ]:
```
test_archive_directory = directory_map[centre]['archived']
test_location_to_move_to = [
    os.path.join(test_archive_directory, item) for item in patient_folder_name]
test_location_to_move_to
```

In [ ]:
```
for i in tqdm(range(len(directories_to_be_archived))):
    assert os.path.exists(directories_to_be_archived[i]), "Error {} doesnt exist an
    assert not(os.path.exists(test_location_to_move_to[i])), "Error {} exists alrea

    print("{} => {}".format(directories_to_be_archived[i], test_location_to_move_to
```

In [ ]:
```
archive_directory = directory_map[centre]['holding']
location_to_move_to = [
    os.path.join(archive_directory, item) for item in patient_folder_name]
location_to_move_to
```

In [ ]:
```
for i in tqdm(range(len(directories_to_be_archived))):
    assert os.path.exists(directories_to_be_archived[i]), "Error {} doesnt exist an
    assert not(os.path.exists(location_to_move_to[i])), "Error {} exists already!".

    print("{} => {}".format(directories_to_be_archived[i], location_to_move_to[i]))
```

In [ ]:
```

```

In [ ]:
```
location_to_move_to
```

In [ ]:
```
for i in tqdm(range(len(directories_to_be_archived))):
    assert os.path.exists(directories_to_be_archived[i]), "Error {} doesnt exist an
    assert not(os.path.exists(location_to_move_to[i])), "Error {} exists already!".

    print("{} => {}".format(directories_to_be_archived[i], location_to_move_to[i]))
    shutil.move(directories_to_be_archived[i], location_to_move_to[i])

    assert not(os.path.exists(directories_to_be_archived[i])), "The move failed to
    assert os.path.exists(location_to_move_to[i]), "Failed to archive!"

    print("    Done\n")
```

Move the the result across to its final location via system tools

Use system tools to move across the data within `\\monacoda\FocalData\ToBeArchived` to `\\UTILSVR\PhysBack\MONACO_ARCHIVE_1` then verify that the data was successfully moved by running the following.

In [ ]:
```
for i in tqdm(range(len(directories_to_be_archived))):
    assert not(os.path.exists(directories_to_be_archived[i])), "The move failed to
    assert not(os.path.exists(location_to_move_to[i])), "The move failed to delete
    assert os.path.exists(test_location_to_move_to[i]), "File not found within arch

    print("    Done\n")
```