Project 3
CS 4471

Create a Gitlab repository labeled Project3 inside the group I invited you to. Work on your project in your Gitlab repository. When finished, tag your latest commit with "Finished" and I will know to grade it. The timestamp of the "Finished" commit will be used to determine if you completed the project on time.

Notes:

Problem 1. You will implement first come first served (FCFS), shortest seek first (SSF) and the elevator algorithm. Start from the given starter code in `scheduler/`.

- The first command-line argument is the starting position of the read head.
- For the elevator algorithm, there is no initial direction. The first cylinder read should just be the closest one.
- Test using the three included data files. Your program will be tested with these and with different files.

Following are four sample outputs. Your output should match these exactly.

```
> ./scheduler 32 test1.dat
FCFS ordering: 32 29 36 20 50
FCFS distance: 56

SSF ordering: 32 29 36 50 20
SSF distance: 54

Elevator ordering: 32 29 20 36 50
Elevator distance: 42
```

```
> ./scheduler 32 test2.dat
FCFS ordering: 32 24 26 27 16 21 27 5 45 59 35
FCFS distance: 133

SSF ordering: 32 35 27 27 26 24 21 16 5 45 59
SSF distance: 87

Elevator ordering: 32 35 45 59 27 27 26 24 21 16 5
Elevator distance: 81
```

```
> ./scheduler 1 test2.dat
FCFS ordering: 1 24 26 27 16 21 27 5 45 59 35
FCFS distance: 148

SSF ordering: 1 5 16 21 24 26 27 27 35 45 59
SSF distance: 58

Elevator ordering: 1 5 16 21 24 26 27 27 35 45 59
Elevator distance: 58
```

```
> ./scheduler 32 test3.dat
FCFS ordering: 32 44 53 12 39 20 38 54 53 60 61 9 50 61 53 50
FCFS distance: 266

SSF ordering: 32 38 39 44 50 50 53 53 53 54 60 61 61 20 12 9
SSF distance: 81

Elevator ordering: 32 38 39 44 50 50 53 53 53 54 60 61 61 20 12 9
Elevator distance: 81
```

Problem 2. Use the code starter given in `deadlock/` to write a program to implement the deadlock detection algorithm with multiple resources of each type. See the TODO comment for where you'll put your changes. Your implementation should match the sample output below.

```
> ./deadlock test1.dat
E = 10

A = 10

C =
    0: 0
    1: 0
    2: 0
    3: 0

R =
    0: 6
    1: 5
    2: 4
    3: 7


Run process 0:
A = 10
C =
    0: 0
    1: 0
    2: 0
    3: 0
R =
    0: 0
    1: 5
    2: 4
    3: 7

Run process 1:
A = 10
C =
    0: 0
    1: 0
    2: 0
    3: 0
R =
    0: 0
    1: 0
    2: 4
    3: 7

Run process 2:
A = 10
C =
    0: 0
    1: 0
    2: 0
    3: 0
R =
    0: 0
    1: 0
    2: 0
    3: 7

Run process 3:
A = 10
C =
    0: 0
    1: 0
    2: 0
    3: 0
R =
    0: 0
    1: 0
    2: 0
    3: 0

The original state is safe.
```

```
> ./deadlock test2.dat
E = 10

A = 2

C =
    0: 1
    1: 1
    2: 2
    3: 4

R =
    0: 4
    1: 3
    2: 2
    3: 3


Run process 2:
A = 4
C =
    0: 1
    1: 1
    2: 0
    3: 4
R =
    0: 4
    1: 3
    2: 0
    3: 3

Run process 0:
A = 5
C =
    0: 0
    1: 1
    2: 0
    3: 4
R =
    0: 0
    1: 3
    2: 0
    3: 3

Run process 1:
A = 6
C =
    0: 0
    1: 0
    2: 0
    3: 4
R =
    0: 0
    1: 0
    2: 0
    3: 3

Run process 3:
A = 10
C =
    0: 0
    1: 0
    2: 0
    3: 0
R =
    0: 0
    1: 0
    2: 0
    3: 0

The original state is safe.
```

```
> ./deadlock test3.dat
E = 10

A = 1

C =
    0: 1
    1: 2
    2: 2
    3: 4

R =
    0: 5
    1: 3
    2: 2
    3: 3


The original state is unsafe.
```

```
> ./deadlock test4.dat
E = 6 3 4 2

A = 1 0 2 0

C =
      0: 3 0 1 1
      1: 0 1 0 0
      2: 1 1 1 0
      3: 1 1 0 1
      4: 0 0 0 0

R =
      0: 1 1 0 0
      1: 0 1 1 2
      2: 3 1 0 0
      3: 0 0 1 0
      4: 2 1 1 0


Run process 3:
A = 2 1 2 1
C =
      0: 3 0 1 1
      1: 0 1 0 0
      2: 1 1 1 0
      3: 0 0 0 0
      4: 0 0 0 0
R =
      0: 1 1 0 0
      1: 0 1 1 2
      2: 3 1 0 0
      3: 0 0 0 0
      4: 2 1 1 0

Run process 0:
A = 5 1 3 2
C =
      0: 0 0 0 0
      1: 0 1 0 0
      2: 1 1 1 0
      3: 0 0 0 0
      4: 0 0 0 0
R =
      0: 0 0 0 0
      1: 0 1 1 2
      2: 3 1 0 0
      3: 0 0 0 0
      4: 2 1 1 0

Run process 1:
A = 5 2 3 2
C =
      0: 0 0 0 0
      1: 0 0 0 0
      2: 1 1 1 0
      3: 0 0 0 0
      4: 0 0 0 0
R =
      0: 0 0 0 0
      1: 0 0 0 0
      2: 3 1 0 0
      3: 0 0 0 0
      4: 2 1 1 0

Run process 2:
A = 6 3 4 2
C =
      0: 0 0 0 0
      1: 0 0 0 0
      2: 0 0 0 0
      3: 0 0 0 0
      4: 0 0 0 0
R =
      0: 0 0 0 0
      1: 0 0 0 0
      2: 0 0 0 0
      3: 0 0 0 0
      4: 2 1 1 0

Run process 4:
A = 6 3 4 2
C =
      0: 0 0 0 0
      1: 0 0 0 0
      2: 0 0 0 0
      3: 0 0 0 0
      4: 0 0 0 0
R =
      0: 0 0 0 0
      1: 0 0 0 0
      2: 0 0 0 0
      3: 0 0 0 0
      4: 0 0 0 0

The original state is safe.
```

```
> ./deadlock test5.dat
E = 6 3 4 2

A = 1 0 0 0

C =
      0: 3 0 1 1
      1: 0 1 1 0
      2: 1 1 1 0
      3: 1 1 0 1
      4: 0 0 1 0

R =
      0: 1 1 0 0
      1: 0 1 0 2
      2: 3 1 0 0
      3: 0 0 1 0
      4: 2 1 0 0


The original state is unsafe.
```