Programming Assignment 3

- **Implement the Banker's deadlock avoidance algorithm**
  - Input: 4 text files, each of which represents **Available** vector, **Max** matrix, **Allocation** matrix, and a list of **requests**, respectively.
  - Output: 1 text file (**output.txt** → please use this precise name)
    - avail.txt

```
3 3 2
```

    - alloc.txt

```
0 1 0\n
2 0 0\n
3 0 2\n
2 1 1\n
0 0 2\n
```

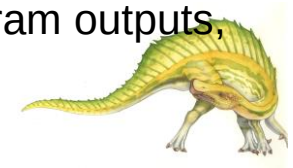
    - max.txt

```
7 5 3\n
3 2 2\n
9 0 2\n
2 2 2\n
4 3 3\n
```

    - req.txt

```
1 1 0 2\n
4 3 3 0\n
```

  - Based on the list of request vectors from the 4th text file, your program outputs, as another file, whether each request can be granted or not.



Programming Assignment 3 (Cont.)

- **Interface**

```
> banker.exe avail.txt max.txt alloc.txt req.txt
```

  - req.txt

```
1 1 0 2\n
4 3 3 0\n
```

    - For each input request, the first number indicates a process id, which is followed by a request vector.
    - If the request can be accepted, your program outputs '**accept**'. Otherwise, output '**reject**'.
  - output.txt

```
accept\n
reject\n
```

  - During the execution, resource allocation state keeps updated according to all the requests that have been successfully granted.

Programming Assignment 3 (Cont.)

- **Submission**
  - Deadline: May 19, 2019, 23:59
  - Upload your source file and 0.5 page description to I-Class
  - Your files should have the following name without any compression:
    - Source file name: **bank.cpp** or **bank.c**
    - Description file name: **readme.txt**