CS 214: Introduction to computing systems
Winter Term, 2026

Lab 7

Allocation safety

In this lab, you'll be implementing the key step of the Banker's algorithm to avoid deadlock. Specifically, you'll read in a scenario showing the number of resources used and required by a set of jobs as well as the number of each kind of resource still available. Then, they will see which jobs the system can guaranttee to finish. (This is using the algorithm we discussed in class on 2/16.)

The first input should be r, the number of kinds of resources. Then it should read the contents of A, which is r integers representing the number of each kind of resource currently available. Next comes n, the number of currently running jobs. Following this is the information for the jobs 1 at a time. Each comes as 2 integers for each resource, first the number of that resource still needed (i.e. a cell of R) and then the number of that resource currently used (i.e. a cell of C). For example, the situation shown on slide 6 from Deadlock.pdf would be represented with the following:

```
3
2 1 1
4
1 1 0 1 2 0
2 0 2 1 1 1
0 2 1 0 1 0
3 1 1 0 0 0
```

The program's output should identify any jobs that can finish (by their number, starting with 0), followed by the message either "All jobs finished" or "No other jobs can finish". For example, the output generated by the input above could be the following:

```
Job 2 completes
Job 3 completes
No other jobs can finish
```

You should dynamically allocate (i.e. using `malloc`) the vector and matrices needed for this problem. Here is code to do this and the corresponding code to free it at the end of the program:

```
//allocate
int** R = (int**) malloc(n * sizeof(int*));
for(int i=0; i<n; i++)
  R[i] = (int*) malloc(r * sizeof(int));

//deallocate
for(int i=0; i<n; i++)
  free(R[i]);
free(R);
```

Use `scanf("%d", ...)` to read the input one integer at a time. (You'll need several loops...) In order to "cross out" jobs that have been completed, I suggest managing R as an array-based list of rows (so you keep track of the number of rows and remove an entry as the corresponding job is finished).