

## CS 2213 Advanced Programming Recitation - Exercise

---

### Files-loops-if-else-command line arguments

**Background information about Critical Path Analysis.** A critical path analysis is a technique used to determine the time schedule for a project. This information is important in the planning stages before a project is begun, and it is also useful to evaluate the progress of a project that is partially completed. One method for this analysis starts by dividing a project into sequential events and then dividing each event into various tasks. Although one event must be completed before the next one is started, various tasks within an event can occur simultaneously. The time it takes to complete an event, therefore, depends on the number of days required to finish its longest task. Similarly, the total time it takes to finish a project is the sum of time it takes to finish each event.

Assume that the critical path information for a major construction project has been stored in a data file. Each line of the data file contains an event number, a task number, and the number of days required to complete the task. The data have been stored such that all the task data for the first event are followed by all the task data for the second event, and so on. A typical set of data is shown in the following table.

Event	Task	Number of Days
1	15	3
1	27	6
1	36	4
2	15	5
3	18	4
3	26	1
4	15	2
4	26	7
4	27	7
5	16	4

You can simply download `critical1.txt` from the web page or simply open a file `critical1.txt` and copy/paste just the above numbers into that file.

### Programming problem:

Write a program that gets two file names as command line arguments after `-input` and `-output` options, for example

```
main212> myprog -input critical1.txt -output output1.txt
or
main212> myprog -output output1.txt -input critical1.txt
```

If user enters less arguments or invalid options, your program should print the above usage information and quit.

Then your program reads the critical path information from `critical1.txt` file and prints a project completion timetable into `output1.txt`. Timetable lists each event number, number of tasks under this event, the maximum number of days for this event, and the total number of days for the project completion.

For example, your `output1.txt` will look like as follows based on the above input file:

Project completion timetable

Event	Num of tasks	Max num.of days
1	3	6
2	1	5
3	2	4
4	3	7
5	1	4

Total number of days to finish the project: 26 days

Note that we might give different input file so your program should be general enough to produce correct result for any input file!

---