

# Sabancı University

Faculty of Engineering and Natural Sciences  
CS204 Advanced Programming  
Spring 2022

## Homework 7 – Simulation of Printer Queue using Threads

Due: 10/06/2022, Friday, 21:00

### PLEASE NOTE:

**Your program should be a robust one such that you have to consider all relevant programmer mistakes and extreme cases; you are expected to take actions accordingly!**

**You can NOT collaborate with your friends and discuss solutions. You have to write down the code on your own. Plagiarism will not be tolerated!**

### Introduction

In this homework, you are asked to write a **multithreaded** C++ program that simulates a printer queue where print jobs are being sent to the queue by multiple users. There are **3 users** sharing the queue of one printer, as depicted in Figure 1. In this homework, you will simulate the process of users **generating print jobs, sending it to the print queue and the printer processing those print jobs** by getting from the queue. The printer queue is a dynamic queue data structure (provided in the homework package), and you will use multithreading techniques.

The waiting time after sending a print job by a user to the queue is probabilistic. **Thus, the minimum and maximum range values for this duration will be input.** Moreover, number of pages in each print job is also probabilistic, **and minimum and maximum range values of it will also be input taken at the beginning** (See Section "Details of Simulation" for details).

In the scope of this homework, **simulation means** to employ a printer queue where print jobs with random amount of pages are being sent to the rear of the queue at random intervals and the printer processes those print jobs by getting from the front of the queue one by one. Simulation starts after taking the inputs from the keyboard and continues until all of the print jobs are processed (printed). During the simulation, you are going to display some verbose output about the actions of users and the printer (see Section "Details of Simulation" for details and Sample Runs for the output format).

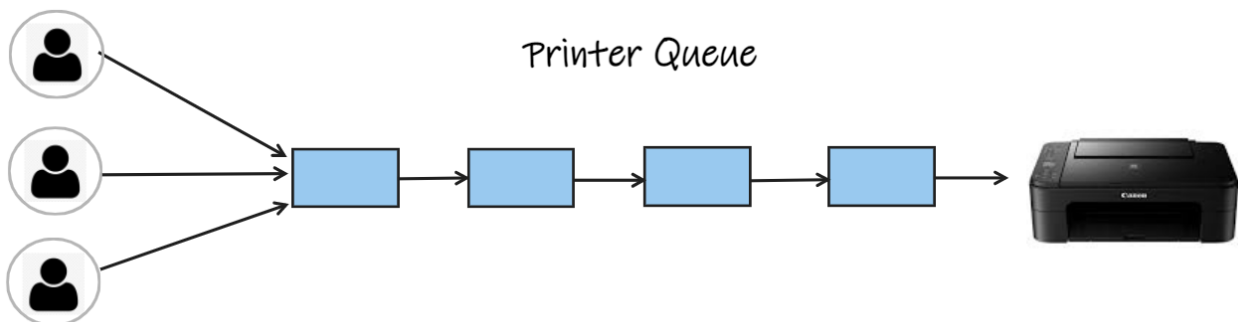


Figure 1: Three users, one printer queue and a printer

## Using Threads

There will be four threads (other than the main thread) in your program. Three threads are for the users, one for each; and one thread for the printer. In the user threads, print jobs with a random number of pages will be generated by the users and then the users will enqueue the print jobs to the rear of the queue. The printer will then take the print jobs from the front of the queue (dequeue) and process it. There is only one queue shared by all threads. Since there are three users, print jobs by different users will be enqueued to the same queue during the simulation. However, there is only one printer dequeuing and printing the jobs from that queue.

Please use the `HW7PrintQueue` class given in this homework package as the queue data structure. Moreover, do not forget that enqueueing (sending) print jobs to the queue will be done in a three users' threads to a single `HW7PrintQueue` queue object. At the same time, the same queue object is shared with the printer and its thread for dequeuing. Such a structure may cause some special cases to deal with in a multithreaded application. For example, there might be a synchronization conflict during enqueueing and dequeuing print jobs to/from the queue. Some other special cases and conflicts may occur; therefore, you have to take some precautions in the threads to avoid such situations. Please refer to lecture and lab materials which include methods for dealing with these cases using *mutex*. At the end, your threads must be joined properly, and your program must terminate without any complications/crashes.

## Details of Simulation

Before the simulation begins, some inputs (total 5 of them) are entered via keyboard. First, *maxPrintJobs* is entered. *maxPrintJobs* is the maximum number of print jobs to be generated in the program (not per user, total for all users). In other words, the simulation will finish when *maxPrintJobs* have been printed by the printer. Moreover, you should not allow more than *maxPrintJobs* to be generated in total. Thus, at the end of simulation, all print jobs should be processed by the printer and the queue must be empty.

Second and third input values are minimum and maximum values for the random time duration (in seconds minutes) that a user should wait after creating a print job. Those values will determine the range of the random number that will be generated to allow a user thread to sleep. Moreover, each user thread should sleep during a random duration before creating the first print job.

Fourth and fifth inputs values are minimum and maximum values for the range of the random number of pages that will be generated for each print job. Assume that one page takes one second to print and you have to sleep the printer thread accordingly to simulate the printing operation.

To generate the random numbers above for time duration and for the number of pages, **do not use the `RandGen` class from CS201** since it is not thread-safe in multithreaded applications. Instead, **you must use the following thread-safe function, `random_range`, for generating random waiting times inside the threads**. This function returns a random number between (and including) `min` and `max` parameters. In the threads, you can call it by passing minimum and maximum values of the corresponding thread as arguments.

```
#include <random>
#include <time.h>

int random_range(const int & min, const int & max) {
    static mt19937 generator(time(0));
    uniform_int_distribution<int> distribution(min, max);
    return distribution(generator);
}
```

Simulation starts when the user enters all the inputs through keyboard. At the beginning of the program (after getting the inputs and just before the beginning of the simulation), you have to display a message saying that the simulation is starting and the current time.

During the simulation, users will generate print jobs in corresponding threads as explained before. Each new print job must be associated with a consecutive ID number starting with 1. You may simulate the time interval after each print job (and before creating the first print job) by sleeping the threads using the `this_thread::sleep_for(chrono::seconds(time_in_seconds))` command of `thread` (for `this_thread::sleep_for`) and `chrono` (for `chrono::seconds`) libraries.

During the simulation, each user thread should display the details of the print job that is sent to the queue. Basically, it will display a message that includes the user ID, print job ID, the number of pages of that print job, the size of the queue after enqueueing, and the time of sending the print job (please refer to the sample runs for more details).

Similarly, the printer thread should display the details of the processed print job. Basically, it will display a message that includes the print job ID, its number of pages and the queue size after the job has been started and the time of start of printing. You have to display another message at the end of printing as well using the same data except the queue size. **Please remark that one page of printing takes one second.**

Please do not forget to sleep each of the user threads before creating the first print job.

At the end of the simulation, a message saying that the simulation is ended must be displayed together with the current time. Please remark that since you sleep the user threads after creating a print job, there could be some time gaps between the end of processing of the last print job and the end of simulation; this basically depends on the min and max parameters and the random values generated. Please see the sample run section for some examples.

### **Two important rules about the simulation details and use of threads:**


- Here please remark that a single line of output from a particular thread may be interleaved by the output of another thread if you do not take appropriate precautions. If this happens, the outputs mix up and become messy. You have to code appropriately, with the help of a dedicated mutex, in order not to end up with such an undesirable situation. Here by "dedicated" we mean this; the mutex that you use for tidy output should not be the one that you use for accessing the shared queue object. These two critical sections are different concepts and must be handled via different mutexes.
- Do not sleep a thread while a mutex is locked. We use sleeping to simulate the waiting duration and printing duration (as number of pages). We do not access shared resources while doing so. Thus, sleeping a thread while a mutex is locked is totally nonsense.
- Any violation to these rules will be penalized severely (50 points or more).

### **HW7PrintQueue class and use of global variables**

We provide a header file (HW7PrintQueue.h) and its implementation (HW7PrintQueue.cpp) together with this homework. **You have to use this queue class without any change.** In order to use it, include both the header and the .cpp to your project. Of course, you have to copy these files to appropriate folders in your project and rename according to our naming convention.

Finally, good news; you may use global variables in this homework. Actually, it would be miserable not to use them in a program that has several threads. However, we kindly request you not to exaggerate the global usage since after a certain point you may lose control over your program (as the famous Turkish proverb says "azı karar, çoğu zarar").

Some versions of C++ compilers impose a limitation for the number of parameters for the thread constructor. It is 5 (except the entry point function address) for VC++ 2012 and if you exceed this, you receive compilation errors. If this amount of parameters is not sufficient for you, you may make some of them global, or pack them into structs/arrays. There is also a way to increase this limit using the steps explained below (taken from [stackoverflow.com](https://stackoverflow.com)); however, since you cannot know whether the grader has the same setup or not, it is better to stay within the standard limit.

Right click on your project in Solution Explorer. Select Properties... and navigate to C/C++ -> Preprocessor and set add \_VARIADIC\_MAX=10 to "Preprocessor Definitions". Make sure you do that for all the configurations of the project (debug, release, etc.) – JP Flouret Nov 13 '14 at 22:44 

---

## Sample Runs

Some sample runs are given below, but these are not comprehensive, therefore you have to consider all cases, to get full mark.

Due to the probabilistic nature of the homework and due to scheduling of threads, same inputs may yield different outputs for your code. However, the order of the events must be consistent with the homework requirements and the given inputs. Nevertheless, occasional (i.e., rare) inconsistencies in the display order of the events occurred at the same time are acceptable.

The inputs from the keyboard are written in ***boldface and italic***.

### Sample Run 1:

```
Please enter the max number of print jobs: 1
Please enter the min and max values for the waiting time period (in seconds) after creating a print job:
Min: 1
Max: 1
Please enter the min and max values for the number of pages in a print job:
Min number of pages: 1
Max number of pages: 1
Simulation starts 23:38:23
User 1 sent new print job with ID 1 sent to the printer queue, number of pages: 1 (print queue size: 1) 23:38:25
The printer started to print the job with ID: 1, number of pages: 1 (queue size is: 0) 23:38:25
The printer finished printing the job with ID: 1, number of pages: 1 23:38:26
End of the simulation ends 23:38:26
```

### Sample Run 2:

```
Please enter the max number of print jobs: 1
Please enter the min and max values for the waiting time period (in seconds) after creating a print job:
Min: 1
Max: 5
Please enter the min and max values for the number of pages in a print job:
Min number of pages: 1
Max number of pages: 1
Simulation starts 23:38:57
User 3 sent new print job with ID 1 sent to the printer queue, number of pages: 1 (print queue size: 1) 23:39:01
The printer started to print the job with ID: 1, number of pages: 1 (queue size is: 0) 23:39:01
The printer finished printing the job with ID: 1, number of pages: 1 23:39:02
End of the simulation ends 23:39:05
```

### Sample Run 3:

```
Please enter the max number of print jobs: 1
Please enter the min and max values for the waiting time period (in seconds) after creating a print job:
Min: 1
Max: 1
Please enter the min and max values for the number of pages in a print job:
Min number of pages: 1
Max number of pages: 4
Simulation starts 23:40:02
User 2 sent new print job with ID 1 sent to the printer queue, number of pages: 1 (print queue size: 1) 23:40:03
The printer started to print the job with ID: 1, number of pages: 1 (queue size is: 0) 23:40:03
The printer finished printing the job with ID: 1, number of pages: 1 23:40:04
End of the simulation ends 23:40:04
```

## Sample Run 4:

```
Please enter the max number of print jobs: 10
Please enter the min and max values for the waiting time period (in seconds) after creating a print job:
Min: 1
Max: 1
Please enter the min and max values for the number of pages in a print job:
Min number of pages: 3
Max number of pages: 7
Simulation starts 23:40:41
User 2 sent new print job with ID 1 sent to the printer queue, number of pages: 7 (print queue size: 1) 23:40:42
The printer started to print the job with ID: 1, number of pages: 7 (queue size is: 0) 23:40:42
User 1 sent new print job with ID 2 sent to the printer queue, number of pages: 6 (print queue size: 1) 23:40:42
User 3 sent new print job with ID 3 sent to the printer queue, number of pages: 7 (print queue size: 2) 23:40:42
User 2 sent new print job with ID 4 sent to the printer queue, number of pages: 5 (print queue size: 3) 23:40:43
User 1 sent new print job with ID 5 sent to the printer queue, number of pages: 3 (print queue size: 4) 23:40:43
User 3 sent new print job with ID 6 sent to the printer queue, number of pages: 7 (print queue size: 5) 23:40:43
User 2 sent new print job with ID 7 sent to the printer queue, number of pages: 5 (print queue size: 6) 23:40:44
User 1 sent new print job with ID 8 sent to the printer queue, number of pages: 4 (print queue size: 7) 23:40:44
User 3 sent new print job with ID 9 sent to the printer queue, number of pages: 4 (print queue size: 8) 23:40:44
User 2 sent new print job with ID 10 sent to the printer queue, number of pages: 7 (print queue size: 9) 23:40:45
The printer finished printing the job with ID: 1, number of pages: 7 23:40:49
The printer started to print the job with ID: 2, number of pages: 6 (queue size is: 8) 23:40:49
The printer finished printing the job with ID: 2, number of pages: 6 23:40:55
The printer started to print the job with ID: 3, number of pages: 7 (queue size is: 7) 23:40:55
The printer finished printing the job with ID: 3, number of pages: 7 23:41:02
The printer started to print the job with ID: 4, number of pages: 5 (queue size is: 6) 23:41:02
The printer finished printing the job with ID: 4, number of pages: 5 23:41:07
The printer started to print the job with ID: 5, number of pages: 3 (queue size is: 5) 23:41:07
The printer finished printing the job with ID: 5, number of pages: 3 23:41:10
The printer started to print the job with ID: 6, number of pages: 7 (queue size is: 4) 23:41:10
The printer finished printing the job with ID: 6, number of pages: 7 23:41:17
The printer started to print the job with ID: 7, number of pages: 5 (queue size is: 3) 23:41:17
The printer finished printing the job with ID: 7, number of pages: 5 23:41:22
The printer started to print the job with ID: 8, number of pages: 4 (queue size is: 2) 23:41:22
The printer finished printing the job with ID: 8, number of pages: 4 23:41:26
The printer started to print the job with ID: 9, number of pages: 4 (queue size is: 1) 23:41:26
The printer finished printing the job with ID: 9, number of pages: 4 23:41:30
The printer started to print the job with ID: 10, number of pages: 7 (queue size is: 0) 23:41:30
The printer finished printing the job with ID: 10, number of pages: 7 23:41:37
End of the simulation ends 23:41:37
```

## Sample Run 5:

```
Please enter the max number of print jobs: 10
Please enter the min and max values for the waiting time period (in seconds) after creating a print job:
Min: 2
Max: 5
Please enter the min and max values for the number of pages in a print job:
Min number of pages: 1
Max number of pages: 1
Simulation starts 23:43:30
User 2 sent new print job with ID 1 sent to the printer queue, number of pages: 1 (print queue size: 1) 23:43:32
The printer started to print the job with ID: 1, number of pages: 1 (queue size is: 0) 23:43:32
User 1 sent new print job with ID 2 sent to the printer queue, number of pages: 1 (print queue size: 1) 23:43:32
The printer finished printing the job with ID: 1, number of pages: 1 23:43:33
The printer started to print the job with ID: 2, number of pages: 1 (queue size is: 0) 23:43:33
User 2 sent new print job with ID 3 sent to the printer queue, number of pages: 1 (print queue size: 1) 23:43:34
The printer finished printing the job with ID: 2, number of pages: 1 23:43:34
User 1 sent new print job with ID 4 sent to the printer queue, number of pages: 1 (print queue size: 2) 23:43:34
The printer started to print the job with ID: 3, number of pages: 1 (queue size is: 1) 23:43:34
User 3 sent new print job with ID 5 sent to the printer queue, number of pages: 1 (print queue size: 2) 23:43:35
The printer finished printing the job with ID: 3, number of pages: 1 23:43:35
The printer started to print the job with ID: 4, number of pages: 1 (queue size is: 1) 23:43:35
The printer finished printing the job with ID: 4, number of pages: 1 23:43:36
The printer started to print the job with ID: 5, number of pages: 1 (queue size is: 0) 23:43:36
The printer finished printing the job with ID: 5, number of pages: 1 23:43:37
User 2 sent new print job with ID 6 sent to the printer queue, number of pages: 1 (print queue size: 1) 23:43:38
The printer started to print the job with ID: 6, number of pages: 1 (queue size is: 0) 23:43:38
User 1 sent new print job with ID 7 sent to the printer queue, number of pages: 1 (print queue size: 1) 23:43:38
User 3 sent new print job with ID 8 sent to the printer queue, number of pages: 1 (print queue size: 2) 23:43:39
The printer finished printing the job with ID: 6, number of pages: 1 23:43:39
The printer started to print the job with ID: 7, number of pages: 1 (queue size is: 1) 23:43:39
The printer finished printing the job with ID: 7, number of pages: 1 23:43:40
The printer started to print the job with ID: 8, number of pages: 1 (queue size is: 0) 23:43:40
User 1 sent new print job with ID 9 sent to the printer queue, number of pages: 1 (print queue size: 1) 23:43:40
User 3 sent new print job with ID 10 sent to the printer queue, number of pages: 1 (print queue size: 2) 23:43:41
The printer finished printing the job with ID: 8, number of pages: 1 23:43:41
The printer started to print the job with ID: 9, number of pages: 1 (queue size is: 1) 23:43:41
The printer finished printing the job with ID: 9, number of pages: 1 23:43:42
The printer started to print the job with ID: 10, number of pages: 1 (queue size is: 0) 23:43:42
The printer finished printing the job with ID: 10, number of pages: 1 23:43:43
End of the simulation ends 23:43:45
```

## Sample Run 6:

```
Please enter the max number of print jobs: 10
Please enter the min and max values for the waiting time period (in seconds) after creating a print job:
Min: 2
Max: 9
Please enter the min and max values for the number of pages in a print job:
Min number of pages: 1
Max number of pages: 8
Simulation starts 23:45:31
User 2 sent new print job with ID 1 sent to the printer queue, number of pages: 1 (print queue size: 1) 23:45:36
The printer started to print the job with ID: 1, number of pages: 1 (queue size is: 0) 23:45:36
The printer finished printing the job with ID: 1, number of pages: 1 23:45:38
User 1 sent new print job with ID 2 sent to the printer queue, number of pages: 4 (print queue size: 1) 23:45:38
The printer started to print the job with ID: 2, number of pages: 4 (queue size is: 0) 23:45:38
User 3 sent new print job with ID 3 sent to the printer queue, number of pages: 4 (print queue size: 1) 23:45:39
The printer finished printing the job with ID: 2, number of pages: 4 23:45:42
The printer started to print the job with ID: 3, number of pages: 4 (queue size is: 0) 23:45:43
User 2 sent new print job with ID 4 sent to the printer queue, number of pages: 5 (print queue size: 1) 23:45:43
User 3 sent new print job with ID 5 sent to the printer queue, number of pages: 7 (print queue size: 2) 23:45:45
User 1 sent new print job with ID 6 sent to the printer queue, number of pages: 6 (print queue size: 3) 23:45:46
The printer finished printing the job with ID: 3, number of pages: 4 23:45:47
The printer started to print the job with ID: 4, number of pages: 5 (queue size is: 2) 23:45:47
User 3 sent new print job with ID 7 sent to the printer queue, number of pages: 3 (print queue size: 3) 23:45:48
User 3 sent new print job with ID 8 sent to the printer queue, number of pages: 6 (print queue size: 4) 23:45:51
The printer finished printing the job with ID: 4, number of pages: 5 23:45:52
The printer started to print the job with ID: 5, number of pages: 7 (queue size is: 3) 23:45:52
User 2 sent new print job with ID 9 sent to the printer queue, number of pages: 7 (print queue size: 4) 23:45:53
User 1 sent new print job with ID 10 sent to the printer queue, number of pages: 5 (print queue size: 5) 23:45:53
The printer finished printing the job with ID: 5, number of pages: 7 23:45:59
The printer started to print the job with ID: 6, number of pages: 6 (queue size is: 4) 23:45:59
The printer finished printing the job with ID: 6, number of pages: 6 23:46:05
The printer started to print the job with ID: 7, number of pages: 3 (queue size is: 3) 23:46:05
The printer finished printing the job with ID: 7, number of pages: 3 23:46:08
The printer started to print the job with ID: 8, number of pages: 6 (queue size is: 2) 23:46:08
The printer finished printing the job with ID: 8, number of pages: 6 23:46:14
The printer started to print the job with ID: 9, number of pages: 7 (queue size is: 1) 23:46:14
The printer finished printing the job with ID: 9, number of pages: 7 23:46:21
The printer started to print the job with ID: 10, number of pages: 5 (queue size is: 0) 23:46:21
The printer finished printing the job with ID: 10, number of pages: 5 23:46:26
End of the simulation ends 23:46:26
```

## Sample Run 7:

```
Please enter the max number of print jobs: 13
Please enter the min and max values for the waiting time period (in seconds) after creating a print job:
Min: 1
Max: 6
Please enter the min and max values for the number of pages in a print job:
Min number of pages: 4
Max number of pages: 13
Simulation starts 23:53:33
User 1 sent new print job with ID 1 sent to the printer queue, number of pages: 6 (print queue size: 1) 23:53:34
The printer started to print the job with ID: 1, number of pages: 6 (queue size is: 0) 23:53:34
User 3 sent new print job with ID 2 sent to the printer queue, number of pages: 11 (print queue size: 1) 23:53:34
User 2 sent new print job with ID 3 sent to the printer queue, number of pages: 13 (print queue size: 2) 23:53:37
User 3 sent new print job with ID 4 sent to the printer queue, number of pages: 5 (print queue size: 3) 23:53:37
User 1 sent new print job with ID 5 sent to the printer queue, number of pages: 11 (print queue size: 4) 23:53:39
The printer finished printing the job with ID: 1, number of pages: 6 23:53:40
The printer started to print the job with ID: 2, number of pages: 11 (queue size is: 3) 23:53:40
User 2 sent new print job with ID 6 sent to the printer queue, number of pages: 12 (print queue size: 4) 23:53:41
User 3 sent new print job with ID 7 sent to the printer queue, number of pages: 6 (print queue size: 5) 23:53:42
User 3 sent new print job with ID 8 sent to the printer queue, number of pages: 8 (print queue size: 6) 23:53:43
User 1 sent new print job with ID 9 sent to the printer queue, number of pages: 7 (print queue size: 7) 23:53:45
User 2 sent new print job with ID 10 sent to the printer queue, number of pages: 10 (print queue size: 8) 23:53:47
User 1 sent new print job with ID 11 sent to the printer queue, number of pages: 10 (print queue size: 9) 23:53:48
User 3 sent new print job with ID 12 sent to the printer queue, number of pages: 7 (print queue size: 10) 23:53:48
User 2 sent new print job with ID 13 sent to the printer queue, number of pages: 5 (print queue size: 11) 23:53:50
The printer finished printing the job with ID: 2, number of pages: 11 23:53:51
The printer started to print the job with ID: 3, number of pages: 13 (queue size is: 10) 23:53:51
The printer finished printing the job with ID: 3, number of pages: 13 23:54:04
The printer started to print the job with ID: 4, number of pages: 5 (queue size is: 9) 23:54:04
The printer finished printing the job with ID: 4, number of pages: 5 23:54:09
The printer started to print the job with ID: 5, number of pages: 11 (queue size is: 8) 23:54:09
The printer finished printing the job with ID: 5, number of pages: 11 23:54:20
The printer started to print the job with ID: 6, number of pages: 12 (queue size is: 7) 23:54:20
The printer finished printing the job with ID: 6, number of pages: 12 23:54:32
The printer started to print the job with ID: 7, number of pages: 6 (queue size is: 6) 23:54:32
The printer finished printing the job with ID: 7, number of pages: 6 23:54:39
The printer started to print the job with ID: 8, number of pages: 8 (queue size is: 5) 23:54:39
The printer finished printing the job with ID: 8, number of pages: 8 23:54:47
The printer started to print the job with ID: 9, number of pages: 7 (queue size is: 4) 23:54:47
The printer finished printing the job with ID: 9, number of pages: 7 23:54:54
The printer started to print the job with ID: 10, number of pages: 10 (queue size is: 3) 23:54:54
The printer finished printing the job with ID: 10, number of pages: 10 23:55:04
The printer started to print the job with ID: 11, number of pages: 10 (queue size is: 2) 23:55:04
The printer finished printing the job with ID: 11, number of pages: 10 23:55:14
The printer started to print the job with ID: 12, number of pages: 7 (queue size is: 1) 23:55:14
The printer finished printing the job with ID: 12, number of pages: 7 23:55:21
The printer started to print the job with ID: 13, number of pages: 5 (queue size is: 0) 23:55:21
The printer finished printing the job with ID: 13, number of pages: 5 23:55:26
End of the simulation ends 23:55:26
```

## Sample Run 8:

```
Please enter the max number of print jobs: 10
Please enter the min and max values for the waiting time period (in seconds) after creating a print job:
Min: 1
Max: 1
Please enter the min and max values for the number of pages in a print job:
Min number of pages: 1
Max number of pages: 1
Simulation starts 16:09:43
User 2 sent new print job with ID 1 sent to the printer queue, number of pages: 1 (print queue size: 1) 16:09:44
The printer started to print the job with ID: 1, number of pages: 1 (queue size is: 0) 16:09:44
User 1 sent new print job with ID 2 sent to the printer queue, number of pages: 1 (print queue size: 1) 16:09:44
User 3 sent new print job with ID 3 sent to the printer queue, number of pages: 1 (print queue size: 2) 16:09:44
User 2 sent new print job with ID 4 sent to the printer queue, number of pages: 1 (print queue size: 3) 16:09:45
The printer finished printing the job with ID: 1, number of pages: 1 16:09:45
User 1 sent new print job with ID 5 sent to the printer queue, number of pages: 1 (print queue size: 4) 16:09:45
The printer started to print the job with ID: 2, number of pages: 1 (queue size is: 3) 16:09:45
User 3 sent new print job with ID 6 sent to the printer queue, number of pages: 1 (print queue size: 4) 16:09:45
User 2 sent new print job with ID 7 sent to the printer queue, number of pages: 1 (print queue size: 5) 16:09:46
User 1 sent new print job with ID 8 sent to the printer queue, number of pages: 1 (print queue size: 6) 16:09:46
The printer finished printing the job with ID: 2, number of pages: 1 16:09:46
User 3 sent new print job with ID 9 sent to the printer queue, number of pages: 1 (print queue size: 7) 16:09:46
The printer started to print the job with ID: 3, number of pages: 1 (queue size is: 6) 16:09:46
User 2 sent new print job with ID 10 sent to the printer queue, number of pages: 1 (print queue size: 7) 16:09:47
The printer finished printing the job with ID: 3, number of pages: 1 16:09:47
The printer started to print the job with ID: 4, number of pages: 1 (queue size is: 6) 16:09:47
The printer finished printing the job with ID: 4, number of pages: 1 16:09:48
The printer started to print the job with ID: 5, number of pages: 1 (queue size is: 5) 16:09:48
The printer finished printing the job with ID: 5, number of pages: 1 16:09:49
The printer started to print the job with ID: 6, number of pages: 1 (queue size is: 4) 16:09:49
The printer finished printing the job with ID: 6, number of pages: 1 16:09:50
The printer started to print the job with ID: 7, number of pages: 1 (queue size is: 3) 16:09:50
The printer finished printing the job with ID: 7, number of pages: 1 16:09:51
The printer started to print the job with ID: 8, number of pages: 1 (queue size is: 2) 16:09:51
The printer finished printing the job with ID: 8, number of pages: 1 16:09:52
The printer started to print the job with ID: 9, number of pages: 1 (queue size is: 1) 16:09:52
The printer finished printing the job with ID: 9, number of pages: 1 16:09:53
The printer started to print the job with ID: 10, number of pages: 1 (queue size is: 0) 16:09:53
The printer finished printing the job with ID: 10, number of pages: 1 16:09:54
End of the simulation ends 16:09:54
```

**Please see the previous homework specifications for the other important rules and the submission guidelines**

**Last, but not the least, you are highly recommended to use Windows for this homework. In multithreaded applications, there are several dependencies to the underlying operating system. Thus the behavior that you see while running in a Windows computer could be totally different than MacOS. Your code will be tested in Windows and if it does not work, even if it works using Mac, we may not re-evaluate your code due to hectic grading period at the end of the semester. To sum up, use Mac at your own risk.**

Good Luck!  
Albert Levi, Ahmed Salem