## Problem 1

Write a program described below:

- The program creates several threads passing to their constructors consecutive letters; each thread prints from time to time its letter (the time interval between printing could be a random number of milliseconds from the interval $[100, 1000]$).
- All threads are started, but the first is suspended.
- Another thread, let's call it the main thread, from time to time (say, every 5 seconds) resumes the thread that is suspended and suspends the next one — cyclically, i.e., if the last one is resumed, the first is suspended.
- After several such cycles (for example, ten of them), the main thread stops all the threads which in turn print a message just before exiting.

**Important:**
All threads are created once, at the beginning of the program.
The methods **stop**, **resume**, **suspend**, and **destroy** from the **Thread** class are inherently unsafe and must not be used!

The output of the program could look as shown below. Here, five threads were launched, corresponding to letters 'a', . . . ,'e', but increasing the number of threads/letters should be a matter of modification of one line only.

```
ebdcebcdbedcbedcdbecbdedbce
Resuming a, suspending b: aedcaedcaecdeadceacdeacdeacade
Resuming b, suspending c: bdaebdebaaedbdbeaedabedbaebda
Resuming c, suspending d: cbeacebaceacbaecbabecabecbacea
Resuming d, suspending e: dbacdbacbadcbacddcbadacbdcabd
Resuming e, suspending a: ecbdecbdecbdecbdecdbecdbecbd
Resuming a, suspending b: aecdaecdeacdecadedcaedacdeace
Resuming b, suspending c: bdaedbaedbedabedabedaebdeabd
Resuming c, suspending d: cbaeacebabecacebaecbaebcaebceab
Resuming d, suspending e: dcabdcabdcadbcacdbabcdabcdab
Resuming e, suspending a: e
Thread b exits
Thread d exits
Thread a exits
Thread e exits
Thread c exits
```