Stage	1	2	3	4	5	Total
Points	4	9	8	3	0	24
Score						

L4: Carthago delenda est

It is the year DCVIII AUC (ab urbe condita, from the founding of the city). The consul Scipio Africanus the Younger (Publius Cornelius Scipio Æmilianus Africanus Minor) has finally captured Carthage. The Senate of the Eternal City, persuaded by Cato the Elder (Marcus Porcius Cato), orders the complete destruction of the enemy city and the salting of its fields so that nothing may grow there again. However, salting such a large area is not as simple as it might seem. It was decided that Scipio would be responsible for salting the fields, while his fellow officeholder, Gaius (Gaius Livius Drusus), would handle transporting the salt to the site. Scipio wants to complete the task as quickly as possible and return to Rome for a well-deserved triumph. However, Gaius, envious of his success, will certainly try to delay the salt deliveries. To handle the political situation appropriately, Scipio has asked you to create a simulation of salting Carthage's fields, which will estimate how long the task will take.

Carthage has been divided into plots of size 10x10 actus (approximately 355x355 meters), or 50 iugera. It is assumed that one sack of salt is sufficient for 1 iugerum, so 50 sacks are required per plot. Once the salt is delivered by Gaius's porters, Scipio's laborers can begin salting. There are $1 \leq N \leq 20$ plots to salt, with one of Scipio's laborers working on each plot. Gaius commands $1 \leq Q \leq 10$ porters.

Note: Part of the initial code is included in the header file `common.h`. This file must not be modified (the server will reject modifications). It contains the implementation of several helper functions, which you should familiarize yourself with.

As always, using global and static variables is prohibited.

Stages:

1. $\boxed{\text{4 pts.}}$ Prepare signal handling for subsequent stages. In the `main` function, there are variables `do_work` along with a mutex. After receiving the `SIGINT` signal, `do_work` should be safely set to 0. Implement signal handling in a separate thread.

2. $\boxed{\text{9 pts.}}$ Implement the simulation of salt delivery to the fields. As per the task description, there are N plots (a shared array among threads), each represented by the number of delivered but unused sacks of salt and the number of already salted iugera. The program creates Q threads for porters. Each thread selects a random plot and then spends (5 + <field number>)ms transporting (sleeping) 5 sacks of salt there. Remember to ensure proper synchronization—each plot should have its own mutex, as two porters may reach it at the same time. For now, the program operates until receiving a `SIGINT` signal—after it is received, porters stop delivering sacks, their threads terminate, and the main process waits for all other threads to finish before ending.

3. $\boxed{\text{8 pts.}}$ Implement a laborer thread. A laborer on a plot takes a sack of salt (if available) and spends 15ms salting the field (sleeping), then prints to the terminal: `SERVVS <laborer number>: ADIPISCOR SALEM <number of fields already salted on this plot> ADHVC IVGERVM`. If no salt is available at a given plot, the thread waits for delivery, printing to the terminal every 0.1 seconds: `SERVVS <laborer number>:  EXSPECTO SALEM`. Ensure proper synchronization. To inform a waiting laborer about salt delivery, use a condition variable (hint: `pthread_cond_timedwait` might be slightly unintuitive. Check the manual. There's a helpful function written in the starter code). When a plot is fully salted, the thread writes to the terminal `SERVVS <laborer number>:  LABOR MEVS CONFICIO` and terminates. The main thread waits for laborer threads to finish; if all plots are salted, Scipio writes to the terminal `SCIPIO: VENI, VIDI, VICI, CONSPERSI`, signals other porter threads to stop working, waits for their completion, and exits the main thread. Hint: To signal the end of work from the main thread, you can simply send `SIGINT`.

4. $\boxed{\text{3 pts.}}$ Scipio has been informed that Gaius will attempt to limit the salting speed by citing the limited capacity of the port where ships with salt dock. Modify the porter threads—before moving salt to a plot, a porter must spend $5ms$ in the port (Gaius's port workers ensure that unloading salt from ships does not proceed too quickly). A maximum of 3 porters can enter the port at once (use a semaphore).

5. $\boxed{\text{0 pts.}}$ `GRADUS ADIUNCTUS`. Modify stage three so that numbers are displayed in the Roman numeral system.