Ain Shams University
Faculty of Computer and Information Sciences
Data Structures Project

Title	Scheduler

Description	There are some workers. Every worker is able to do a different set of tasks/jobs. Such as the following example:
	
	`Worker A (can do): T2, T3`
	
	`Worker B : T1, T3, T4`
	
	`Worker C : T3, T5`
	
	And you have a list of tasks which must be done. For example, the list is something like: T1[2], T3[3], T5[4]. Where each task needs certain amount of time to be performed (the number between brackets).
	
	There's some constraints:
	
	1. Each task must be taken by one worker
	2. Several tasks can be taken concurrently
	3. But a worker can do only one task at the same time. (He/she is not available until finish the task)
	
	For the above example, we may have a schedule like this:
	
	`T1 --> Worker B`
	
	`T3 --> Worker C  T5 --> Worker C`
	
	As you may notice, the above schedule is not optimal. Because T5 has to wait worker C to finish T3. The total time needed for finishing all tasks is 7 hours. The total idle time is 3 hours.
	
	The following solution is better:
	
	`T1 --> Worker B  T3 --> Worker A  T5 --> Worker C`
	
	Because there's no wait. And total time needed is 4 hours.
	
	Now suppose that you know the worker-tasks matrix (what worker can do what tasks). Given a set of random order tasks with their times. You are asked to design a scheduler that automatically finds an idle worker for every task. And when finally all the tasks are done, there's a minimum waiting time and minimum total working time.

Ain Shams University
Faculty of Computer and Information Sciences
Data Structures Project

Group size	4 members.
Deliverables	1- A C++ application that performs the required task
Bonus extensions	Represent the problem Graphically
Mentor	T.A. Dina Khattab
Notes	