

ΕΡΓΑΣΙΑ 1

conditions: terminatingMain, waitingMain, workersQueue

global μεταβλητές: terminatedWorkers, waitingWorkers

Worker:	Main:
<pre>While(1){ if(first!=1){ while(1){ cond_wait(workersQueue); if(num == -1){ terminatedWorkers++; if(terminatedWorkers == flag){ cond_signal(terminatingMain); } mutex_unlock return; } if(isready==0){ mutex_unlock break; } else{ cond_signal(workersQueue); } } } else{ first=0; } primetest(num); mutex_lock waitingWorkers++; isready = 1; if(waitingWorkers==1){ cond_signal(waitingMain); } }</pre>	<pre>for(workers){ scanf(temp); if(EOF){break;}} flag++; workers[i]->number=temp;workers[i]->isready=0; pthread_create(workers[i]); } while(1){ scanf(temp); if(EOF){break;}} mutex_lock if(waitingWorkers==0){cond_wait(waitingMain)} mutex_unlock mutex_lock for(flag){ if(workers[i]->isready==1){ workers[i]->number=temp;workers[i]->isready=0; waitingWorkers--; cond_signal(workersQueue); mutex_unlock break; } } } for(flag){ mutex_lock if(waitingWorkers==0){cond_wait(waitingMain)} mutex_unlock mutex_lock for(flag){ if(workers[i]->isready==1){ workers[i]->number=-1; workers[i]->isready=0; waitingWorkers--; cond_signal(workersQueue); mutex_unlock break; } } } mutex_lock if(terminatedWorkers<flag){ cond_wait(terminatingMain); } mutex_unlock</pre>