1) First serial job
   a) Submit a serial job that:
      i) Is a serial (1 core) job
      ii) Emails you when is starts, ends and aborts
      iii) Has a maximum wall time of 2 minutes
   b) Watch your job run with the following command:
      i) “qstat -a -u $USER”
   c) Did you get the result emailed to your account

2) Second serial job
   a) Submit a serial job that:
      i) Is a serial (1 core) job
      ii) Emails you when is starts, ends and aborts
      iii) Has a maximum walltime of 2 minutes
      iv) Sleeps for 200 seconds
   b) Describe what will you think happen when this jobs runs?
   c) Run your job and see what happens

3) Third serial job (Job Names)
   a) Submit a serial job that:
      i) Is a serial (1 core) job
      ii) Has a maximum wall time of 2 minutes
      iii) Sleeps for 30 seconds
      iv) Is named “my-3rd-job”
   b) Look at your job in the output of the following commands
      i) “showq –u $USER”
      ii) “qstat -a –u $USER”
      iii) “jobinfo –j”

4) Job Arrays
   Submit a serial job array that:
   a) Has a maximum wall time of 2 minutes
   b) Sleeps 30 seconds
   c) Is named “my-array-job”
   d) Has 12 tasks
   e) Writes a files to $PBS_JOBNAME.output
   f) Look at your job with the following commands
      i) “showq –u $USER”
      ii) “qstat -a –u $USER”
      iii) “jobinfo –j”
5) **(Extra Credit) Job Arrays**
   
   Submit a serial job array
   
   a) Has a maximum wall time of 2 minutes
   b) Sleeps 30 seconds
   c) Is named "my-array-job"
   d) Has 12 tasks
   e) Writes a files to $PBS_JOBNAME
   f) Runs at most 2 jobs at once

6) **(Extra Credit) Job Arrays**
   
   Submit a serial job array that
   
   a) Sleeps 30 seconds
   b) Is named "my-array3-job"
   c) Has 4 tasks with indexes of: 1, 2, 7, -13
   a) Writes a files $PBS_JOBNAME
   b) Look at your job with the following commands
      
    i) “showq –u $USER”
    ii) “qstat –a –u $USER”
    iii) “jobinfo –j”
   d) Look at the output of the job,
      
    i) Notice that the negative number (-13) is turned positive.

7) MPI Jobs
   
   a) Edit the start-7.pbs job script replace the email address with your email address
   b) Submit the edited script start-7.pbs
   c) Look at the job with the following commands:
      
    i) “showq –u $USER”
    ii) “qstat –a –u $USER”
    iii) “jobinfo –j”
    iv) “checkjob <jobid>”
   d) Note how long it took to run
   e) Edit the start-q7.pbs script to user 4 processors
   f) Submit the edited script
   g) Look at the job with the following commands:
      
    i) “showq –u $USER”
    ii) “qstat –a –u $USER”
    iii) “jobinfo –j”
    iv) “checkjob <jobid>”
   h) Please list which nodes and cores the job is running on or scheduled to run on and how long it took to run.
8) OpenMP jobs  
   a) Submit a job  
      i) Asking for 1 node with 12 cores  
      ii) sleeps 60 seconds  
      iii) Has a maximum walltime of 2 minutes  
      iv) Look at the job with the following commands:  
          • “showq –u $USER”  
          • “qstat –a –u $USER”  
          • “jobinfo –j”  
          • “checkjob <jobid>”  
   v) Please list which nodes and cores the job is running on or scheduled to run on.

9) (Extra Credit) Hybrid Jobs  
   Submit a job that:  
   a) Asking for 4 nodes with 4 cores  
   b) sleeps 60 seconds  
   c) Has a maximum walltime of 2 minutes  
   d) Look at the job with the following commands:  
       • “showq –u $USER”  
       • “qstat –a –u $USER”  
       • “jobinfo –j”  
       • “checkjob <jobid>”  
   ii) Please list which nodes and cores the job is running on or scheduled to run on.

10)(Extra Credit) Jobs and features  
   a) Submit a serial job  
      i) Asks for older cpus: “L5420” feature  
      ii) Look at the job with the following commands:  
          • “showq –u $USER”  
          • “qstat –a –u $USER”  
          • “jobinfo –j”  
          • “checkjob <jobid>”  
   b) Is your job taking a long time to be run? Why?

11) Jobs and memory (mem,pmem)  
   a) Take the start-q11.pbs script and edit  
      i) ask for: pmem=12000mb  
      ii) replace the email address with your email address  
   b) Submit a job from the script you edited. Look at the job with the following commands:  
      i) “showq –u $USER”  
      ii) “qstat –a –u $USER”  
      iii) “jobinfo –j”
iv) “checkjob <jobid>”

c) How much memory does this job use?

12) Jobs and memory (mem, pmem)
   a) Take the start-q12.pbs script and edit it
      i) ask for: mem=12000mb
      ii) replace the email address with your email address
   b) Submit a job from the script you edited. Look at the job with the following commands:
      i) “showq –u $USER”
      ii) “qstat –a –u $USER”
      iii) “jobinfo –j”
      iv) “checkjob <jobid>”
   c) How much memory does this job use?

13) Jobs and memory (mem, pmem)
   a) Take the start-q13.pbs script and edit
      i) asks for: pmem=3000mb
      ii) replace the email address with your email address
   b) Submit a job from the script you edited. Look at the job with the following commands:
      i) “showq –u $USER”
      ii) “qstat –a –u $USER”
      iii) “jobinfo –j”
      iv) “checkjob <jobid>”
   c) How much memory does this job use?

14) Jobs and memory (appropriate resources)
   a) Create a job run the “cryptic” program edit the start-q14.pbs script
      i) Make sure your job emails you when is starts, ends and aborts
      ii) Make a guess and for enough RAM to run the program
   b) Submit your edited Job script, look at your running Job with the following commands, look at the memory used by your job
      i) “checkjob <jobid>”
      ii) “qstat -f <jobid>”
   c) Did your job run successfully? Or fail because of a lack of memory?
      i) If your job failed due to a lack of memory, increase the maximum memory requested and resubmit your job, and go back to point 14b
   d) Look at the email reporting on your job success, how much resources were reported used. Compare the memory used to the reported memory in 14c
   e) Edit job script and request an appropriate amount of memory to run the Job as learned in 11e.
   f) Submit your new job
   g) Verify that the jobs runs successfully.
15) *(Extra Credit)* Full nodes  
 a) Submit a job asking for that asks for  
   i) 4 procs  
   ii) Not to run on any nodes with other users  
       Useful if you are trying to debug your job  
 b) See if you can see which nodes your job is running on.  
   i) “checkjob <jobid>”  
   ii) “qstat -f <jobid>”

16) *(Extra Credit)* Full nodes  
 a) Submit a job asking for that asks for  
   i) 4 procs  
   ii) Not to run on any nodes with other jobs  
       Useful if you are trying to debug your job  
 b) See if you can see which nodes your job is running on.  
   i) “checkjob <jobid>”  
   ii) “qstat -f <jobid>”

17) *(Extra Credit)* Full nodes  
 a) Submit a job asking for that asks for  
   i) 4 procs  
   ii) Each task should run on a separate node.  
 b) See if you can see which nodes your job is running on.  
   i) “checkjob <jobid>”  
   ii) “qstat -f <jobid>”

18) *(Extra Credit)* *(Do only If you are part of multiple WestGrid projects)*  
 Multiple accounting groups  
 a) Submit a Job to a non default accounting group, that asks for 1 proc  
 b) Try to see which accounting group your job belongs to, use the checkjob command:  
   i) “checkjob <jobid>”

19) *(Extra Credit)* Software licenses and generic resources  
 a) Submit a job asking for that asks for  
   i) 1 proc  
   ii) 1 MATLAB license  
   iii) 1 Statistics_Toolbox license  
 b) Try to see which accounting group your job belongs to, use the checkjob command:
20) (Extra Credit) Job dependencies

a) Submit a serial job named 20.a, that has
   i) Walltime of 2:00
   ii) Sleeps 120 seconds
b) Submit a serial job 20b waits until job 15a is done
   i) Walltime of 2:00
   ii) Sleeps 120 seconds
c) Look at job 15.b with checkjob
d) Run the command “showq –u $USER”
e) Verify that Job 15.a complete before 15.b starts

21) (Extra Credit) Job epilogue

a) look at the epilogue.script file provided
b) Submit a job that runs the provided epilogue script named “epilogue.script”
   after the job is done as epilogue.

22) (Extra Credit) Job using temporary directory

a) Submit a job that runs in the temporary directory used no more than
   1000mb of space,

23) (Extra Credit) Job environment variables.

a) Submit a serial job that prints the queue that the job was ran in.

24) Basic Job info

a) Use the “Jobinfo –j” and “qstat -t -u <username>” commands to find out how
   many jobs your have running, queued, in hold state or complete.

b) Use the “showq” command what does active, eligible, blocked jobs mean,
   how many jobs are in each category

c) Use the “showq –b” command to see how many jobs are in what state?

25) Examining a job

a) Start a Job

b) Examine its priority with ”jobinfo –i” or “showq -i”

c) run qstat -f <jobid > and determine how much RAM the Job asks for/used

d) run checkjob –v –v <> determine and examine the result , what is its priority
26) Priority, Fairshare, and allocations
   a) Start a Job, what is its priority relative to other jobs.
   b) What is your research groups allocation and usage
   c) Which person in your group has used the most resources, have they used more than their share?
   d) When were the most Jobs ran by your group

27) Cluster info
   a) How many idle nodes are on the cluster “mdia –n”
   b) How many cpus and memory are not being used on the cluster?
   c) How many cores with 2 GB Ram are not being used on the cluster?
   d) How many nodes are set offline on a cluster.