For Q&A tasks, fill your answers in the cell, or at least fill “A”, “P” or “N” in all the ”Your self-evaluation” cells.
A-all finished P-Partially finished N-None finished
If you mark the task “N”, it receives 25% of max possible points for the task (It does not apply to extra-point tasks).
If you leave it blank, it receives 0 points.
If your mark the task "A" or "P" but zero work is done for the task, then it receives 0 points.
Please leave "score" cells blank for course staff
If your code does not build, your code will not receive more than 50% of max possible points

Task1: git and project structure	**Max score (%)**	**Your answer/Your self-evalution**	**1st pass: autograder (NA for the task)**	**2nd pass: manual review**	**final score**
Your github repo should be named “CS371F21”;	10% total	Done			
Your github repo CS371F21 must be private, and you must add the instructor(jyuan2pace) as a collaborator;	missing any one - 5%	Done			
All your work for this project should be placed under directory “project1”, which is immediately under repo CS371F21;	missing two or more - 0%	Done			
Design and answers for task 2 should be immediately under a folder called task2, immediately under project1;		Done			
Source code for task 3 should be immediately under a folder called task3 immediately under project1;		Done			
This grading sheet should be completed and shared with the instructor		Done	0		**0**

Task2: design and code review	**Max score (%)**	**Your answer/Your self-evalution**	**1st pass: autograder (NA for the task)**	**2nd pass: manual review**	**final score**
Every class is shown in the class diagram	1				
Correct/reasonable associations and attributes	2				
Methods and atrributes have proper type and type modifier (static, private, public etc)	2	Done			
Good practice of PIE	5				
All unit tests are reviewed and commented with list states	10		0		**0**

Task3: Implementation	**Max score (%)**	**Your answer/Your self-evalution**	**1st pass: autograder**	**2nd pass: manual review**	**final score**
Is constructor implemented correctly?	5				
Is allocate implemented correctly? Does it support -- first fit? -- best fit? -- next fit? Does it use list of free blocks and used blocks? and uses own list to split/delete blocks?	30				
Is free implemented correctly? -- Does it compact free space? -- Does it give a nice error if you free a silly address? -- Use your own list to insert and consolidate blocks?	15				

Is size implemented correctly?	5				
Is max_size implemented correctly?	5				
Is free and used blocks tracked using **your own linked list**? -- Can your program output a list of used and free blocks in ascending order of offset ?	10				
Does your implementation devite signifantly from the design (task2)?	[-5, -10]				**0**
Penalty				**Total**	**0**
				Late penalty	
				Any member misses git commit	
Bonus				**Total**	**0**
				Extra tasks	
				Early bird	
Final total score					**0**