**Question 4**

State whether a stack is appropriate for each of the following tasks. Indicate why or why not.

***a. A word processor must remember a line of up to 80 characters. Pressing the Backspace key deletes the previous character and pressing CTRL and Backspace deletes the entire line. Users must be able to undo deletion operations. (6 marks)***

Using of stack is appropriate. When user deletes the previous character, it should be deleted starting from the character he or she last keyed in. This suits the stack that implements “Last-In-First-Out” basis. In order to have the undo function, the character or line that about to be deleted must be stored to a variable before carrying out the deletion operation. Based on the example in Figure 1, the characters are keyed in from the left to the right, where ‘P’ is the first character and ‘T’ is the last character. Each line in the word processor is a stack. When backspace is pressed, ‘T’ (Last-In) will be deleted first (First-Out) by POP operation. When CTRL and backspace is pressed, the POP operation will be repeated using a loop until all the nodes in the stack have been deleted. When user wants to undo, the deleted line or character that stored in a variable will be inserted to the Top of the stack by PUSH operation. If the user just deleted the character ‘T’ and he wants to undo, ‘T’ that stored in a variable will be inserted to the Top of the stack, which mean ‘T’ is inserted to its original position before it was deleted.

***b. Customers must wait one to three months for delivery of their new automobiles. The dealer creates a list that will determine the “fair” order in which customers should get their cars; the list is to be prepared in the order in which customers places their requests for a new car. (6 marks)***

Using of stack is not appropriate. Stack implements “Last-In-First-Out” basis while the list created by the dealer is implementing “First-Come-First-Serve” basis, which is actually a queue structure (“First-In-First-Out”). Based on the example in Figure 2, if the POP operation is carried on the stack, the last customer, “C8”, will be the one that get his order and being removed from the stack. By right, the first customer, “C1”, should get his order first as he was the one that place the order first. If the dealer wants the POP operation to remove the first customer, “C1”, he has to add the customers from the last one to the first one to the stack, which causes inconveniences. The right and suitable data structure to be used by dealer should be queue (refer Figure 3). Using queue structure, dealer can append the nodes when he received new order. When SERVE operation is carried out on the queue, the first customer, “C1”, will be removed from the queue and given the ordered item.

***c. You are required to search downward in a pile of magazines to locate the issue for last January. Each magazine was placed on the pile as soon as it was received. (6 marks)***

Using of stack is appropriate. When we are searching downward in a pile of magazine, we are actually searching from the uppermost magazine. This searching process is exhibiting “Last-In-First-Out” behaviour, which mean the newest magazines on the top of the pile will be searched first. Thus, using of stack structure that exhibits “Last-In-First-Out” behaviour is suitable. Based on Figure 4, “M8” is the latest received magazine that placed on the top of the pile. “M1” is the first magazine received and the following received magazines will be added to the stack by PUSH operation. Each new magazine added to the stack will be the “Top” of the stack. When searching through the pile of magazine, the “Top” will be searched first following by the next node and continue downward until the target magazine has been found.

***d. A programming team accepts jobs and prioritizes them based on urgency. (6 marks)***

Using of stack is not appropriate. This is because when we are prioritizing a series of jobs, we are actually choosing the most important one followed by the less important one. Based on Figure 5, “T1” is the most important job and being inserted to the stack first followed by less important one. As stack exhibits “Last-In-First-Out” behaviour, when the POP operation is carried out on the stack, the “Top” node, which is “T8” (the least important job), will be handled and removed from the stack. Supposedly, “T1” (the most important job) should be handled first. The right and suitable data structure to be used should be queue (refer Figure 6) as it exhibits “First-In-First-Out” behaviour. Using queue structure, the most important job (“T1”) will be handled and removed from the queue first when the SERVE operation is carried out on the queue. The team can also prioritize the jobs by choosing the most important job first with the use of queue structure.

***e. A line formed at a bus stop. (6 marks)***

Using of stack is not appropriate. Stack implements “Last-In-First-Out” basis while the line formed at the bus stop is actually a queue that implements “First-In-First-Out” basis. People who reached the bus stop first can get into the bus first. Based on Figure 7, “P1” is the first person who reached the bus stop. If POP operation carried out on the stack, “P8” will be the one to be removed from the line (stack) and get into the bus, instead of “P1”. The right and suitable data structure to be used should be queue (refer Figure 8). When SERVE operation is carried out on the queue, the first person who came to the bus stop, “P1”, will be removed from the line (queue) and get into the bus. The next node of the last node deleted will be the next to be removed if the SERVE operation carries out again. In queue structure, each SERVE operation will remove the head nodes, so the nodes that be added first (First-In) will be remove first (First-Out).