

Assignment 2 Fall 2018

Due Date: Sunday 21st October 2018, 05:00 PM

Please keep in mind the following guidelines:

- Do not share your program code with anyone.
- Do not copy code from the internet.
- If you receive any assistance, mention the part of code in which you received assistance.
- You must be able to explain any part of your submitted code.
- All submissions are subject to automated plagiarism detection.

Submission:

You have to submit all the .cpp files containing source code. Zip all .cpp files into one file named as <your8DigitRollNumber>_Assignment2_CS200.zip and submit the zip file.

Total Marks: 60

Question 1 [40 marks]

You have to implement following two Array works you have to implement two separate classes

(A) Array work 1 [20 marks]

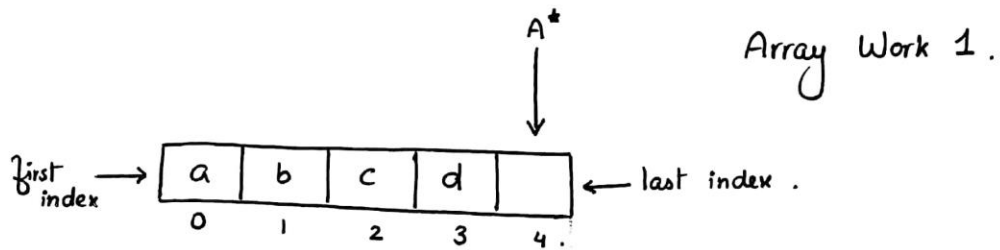
A char array in this class should be of the size given input by the user and must exhibit the pattern that the last most entered element must be the first to be taken out. That is the user can only see or get the top most element of the array. For this you need to implement the following functions

- Constructor that takes size as input
- **Char Add_element(char c)** : show error if array full otherwise adds element on top.
- **Char delete_element()**: show error if array empty otherwise return top most element from array and remove it from array.
- **Void peek()**: show top element without removing for this you need to use add_element and **delete_element** function only
- **Int size()** : return size of the array.
- **Bool is_empty()** : a Boolean function that returns true if array is empty.
- **Bool is_full()** : a Boolean function that returns true if array is full

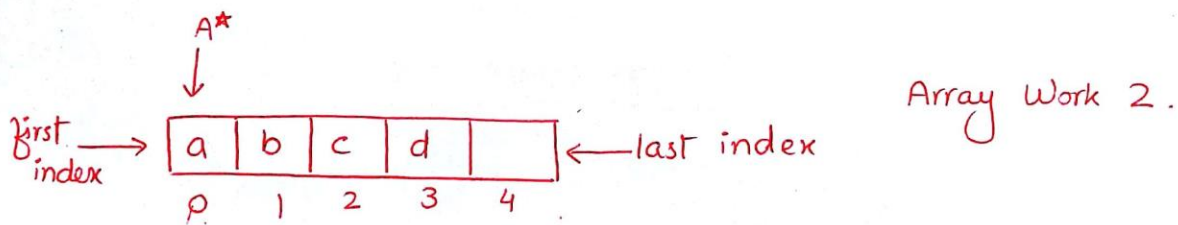
(B) Array work 2 [20 marks]

This works the same as “**Array work 1**” but the user can only see or get the first element of array. That is the add_element adds on the first location only pushing the others back, the delete_element removes the first element etc. implement all functions of “**Array work 1**” accordingly.

Following Diagrammatic representation illustrates the above code better.



- A^* points to top empty space.
- `add_element('e')` will enter 'e' on place 4.
- `delete_element()` will give us 'd'.
- `peek()` will show us 'd'.
- `size()` will give us 5



- A^* points to first index of array.
- `add_element('e')` pushes all elements one space ahead and puts 'e' at location 0.
- `delete_element()` will give us 'a' and bring all elements one space back.

Question 2 [10 marks]

Given the following infix notation expression, convert it into postfix notation and save it in “**Array work 1**” and “**Array work 2**”. Solve it separately.

$$3 + 4 / 5 + 6$$

Question 3 [10 marks]

Given the following infix notation expression, convert it into prefix notation and save it in “**Array work 1**” and “**Array work 2**”. Solve it separately.

$$3 - 8 * 5 \% 2$$

Following is a very good video illustrating how one can convert infix to prefix and postfix and solve them.

<https://www.youtube.com/watch?v=jos1Flt21is>

Following illustrations might help you out in making a better solution.

| Infix | Prefix | Postfix |
|-----------|---------|---------|
| $x+y$ | $+xy$ | $xy+$ |
| $x+y-z$ | $-+xyz$ | $xy+z-$ |
| $x+y*z$ | $+x*yz$ | $xyz*+$ |
| $x+(y-z)$ | $+x-yz$ | $xyz-+$ |

Examples

| Infix | Postfix |
|-----------------------|-----------------|
| $2+3*4$ | $234*+$ |
| $a*b+5$ | $ab*5+$ |
| $(1+2)*7$ | $12+7*$ |
| $a*b/c$ | $ab*c/$ |
| $(a/(b-c+d))*(e-a)*c$ | $abc-d+/ea-*c*$ |
| $a/b-c+d*e-a*c$ | $ab/c-de*ac*-$ |