

Problem 1

Implement List using Array

The problem is to implement a **list of integers** using an array. Your implementation must support the following list operations. The operations should be implemented as efficiently as possible.

- insertItem:** First check whether the list is full or not. If not full, then insert a given integer at the end of the list and print the modified list. If full, then allocate a new memory whose size is **double** of the existing list size, copy the existing items to the newly allocated memory, deallocate the existing memory, insert a given integer at the end of the list and print the modified list.
- insertItemAtPos:** First check whether the list is full or not. If not full, then insert a given integer at a given position, move all items whose position was greater than or equal to given position one position right and print the modified list. If full, then allocate a new memory whose size is double of the existing list size, copy the existing items to the newly allocated memory, deallocate the existing memory, insert a given integer at a given position, move all items whose position was greater than or equal to given position one position right and print the modified list.
- deleteItem:** Delete the last Item from the list. If the length of the list becomes 0 after deletion, then print "Empty List" and return the deleted element. Otherwise, print the modified list and return the deleted element.
- deleteFirstItem:** Delete the first occurrence of a given integer from the list. If the given integer does not exist in the list, print "Element not Found" and return -1. Otherwise, print the modified list and return the deleted element.
- deleteLastItem:** Delete the last occurrence of a given integer from the list. If the given integer does not exist in the list, print "Element not Found" and return -1. Otherwise, print the modified list.
- clearList:** Delete all the elements from the list and print "Clear Completed".
- searchItem:** Determine if a given integer exists in the list. If the given integer exists, print "Found at position ____" and return the first occurrence position of the element. Otherwise, print "Not Found" and return -1.

Now you have to write main function that processes line of inputs according to the following format using your own implementation:

- ✓ Each line of input starts with a code letter: I for insert, D for delete, C for Clear and S for search.
- ✓ I followed by an integer indicates that you have to insert the given integer at the end of the list and print the modified list.
- ✓ I followed by two integer indicates that you have to insert the first given integer at the last given integer position, move all items whose position was greater than or equal to given position one position right and print the modified list.

- ✓ **D** indicates that you have to delete the last item from the list and print the modified list.
- ✓ **D** followed by a character **F** and an integer indicates that you have to delete the first occurrence of a given integer from the list and print the modified list.
- ✓ **D** followed by character **L** and an integer indicates that you have to delete the last occurrence of a given integer from the list and print the modified list.
- ✓ **C** indicates that you have to delete all the elements from the list and print "Clear Completed".
- ✓ **S** followed by an integer indicates that you have to find the first occurrence position of the integer in the list and print "Found at position ____".
- ✓ A special code letter **T** can be used to terminate the program.
- ✓ Assume that initially the list is empty.
- ✓ Follow the format of the sample input and output provided below.

Sample Input	Sample Output
I 2	2
I -1	2 -1
I 3	2 -1 3
I 4	2 -1 3 4
I 5 1	2 5 -1 3 4
I 5	2 5 -1 3 4 5
I 5	2 5 -1 3 4 5 5
D L 5	2 5 -1 3 4 5
D F 5	2 -1 3 4 5
D F 6	Element not Found
D L 6	Element not Found
S -1	Found at position 1
S 2	Found at position 0
C	Clear Completed
I 5	5
I 4	5 4
D	5
D	Empty List
T	

Problem 2

Balanced Parenthesis Problem

In this task, you will solve the balanced parenthesis problem using the list implemented in the previous problem. In this problem, a parenthesis structure will be given as input, for example "([()])". You may store this in a *string* variable. Your program will determine whether the input parenthesis is structurally correct, i.e., it forms a well-balanced parenthesis. The algorithm to check this is as follows:

1. Initialize your list at the beginning of your program.
2. Take input string from user.
3. Scan the characters of the input string one by one starting from the leftmost position. For each character, do any one of the following:
 - a. If it is a "(", "{", or "[", then insert the character to the list by using *insertItem* function. Do not worry about inserting characters to *int* type list. It will work perfectly!
 - b. If it is a ")", "}", or "]", then do not insert it. First, check whether the list is empty. If it is, then this indicates an unbalanced parenthesis structure and go to step 5 directly. If not empty, then delete one from the list by calling the *deleteItem* function. Then check whether the removed item is the left matching of the currently scanned character. If it is, then carry on scanning. If not, then it denotes an unbalanced parenthesis structure. You may stop scanning at this point and go to Step 5 directly.
4. After scanning of all characters, check the length of the list. If it is not 0, then it implies that the input parenthesis was not balanced.
5. Give your output as "Balanced" or "Not Balanced".
6. Clear your list by calling the *clearList* function.

Sample Input	Sample Output
([])	Balanced
[()]	Not Balanced
(O[])	Balanced
(O[]	Not balanced

Problem 3

Implement Sorted List using Array

The problem is to implement a sorted list of integers using an array. Your implementation must support the following list operations. The operations should be implemented as efficiently as possible.

insertItem: Insert a given integer in such a position that the list remains sorted and print the modified list.

deleteItem: Delete the first occurrence of a given integer from the list. If the given integer does not exist in the list, print "Element not Found". Otherwise, print the modified list.

findItem: Determine if a given integer exists in the list. If the given integer exists, print "Found at position ____ " and returns the first occurrence position of the element. Otherwise, print "Not Found".

Each line of input starts with a code letter: I for insert, D for delete and F for find. The code letter is followed by a blank space and then an integer, which is to be inserted, deleted or found. For each line of input, a single line should be printed as output as described above. A special code letter T can be used to terminate the program. Assume that initially the list is empty. Follow the format of the sample input and output provided below.

Sample Input	Sample Output
I 2	2
I -6	-6 2
D -6	2
D -2	Element not Found
I 1	1 2
I 3	1 2 3
F 2	Found at position 1
D 2	1 3
F 2	not found
I 4	1 3 4
I 3	1 3 3 4
D 3	1 3 4
T	

Other Requirements for the Assignment:

- ✓ You cannot use any advanced features of C library.
- ✓ You must use dynamic memory allocation.
- ✓ You must free unused memory where it is required.
- ✓ You may assume that the input will not contain any invalid characters.
- ✓ You must not use other's code. You must not share your code. You must not copy from any other sources such as web, friends, relatives, etc. In all cases, you will earn a -100% mark and will move closer to getting an "F" grade in the course.