Assignment 1 : Playing with Arraylist

Deadline: 10/11/2020 11:55 PM

Instruction:

1. Write your code in the c file named "static_arraylist.c".
2. Write an explanatory comment for every important block of code(a loop or a condition). Your marks will be reduced to half if you don't write the comments.
3. Avoid plagiarism. If you are found to adopt any unfair means you will get a straight 0.
4. Upload your code (only the c file) in elms.
5. Deadline is 10/11/2020 11:55 PM.

Task:

In this assignment you have to implement the arraylist. In the zip file you will find a c file named "static_arraylist.c". There you will see some function prototypes. Your task is to complete these functions. Description and sample input output are listed below.

1. int is_full() function returns 1 is the arraylist is full, and 0 otherwise.
2. int is_empty() function returns 1 is the arraylist is empty, and 0 otherwise.
3. int insert_first(int item) function inserts "item" to the beginning of the arraylist. Be careful not to cross the "size" of the arraylist**.** It should return 1 for a successful insertion, and 0 for an unsuccessful insertion. For example,

Sample input	Sample output
Arraylist: insert_first(10);	Returns 1 Arraylist: 10
Arraylist: 10 15 20 25 30 insert_first(35);	Returns 1 Arraylist: 35 10 15 20 25 30

4. int insert_last(int item) function inserts "item" to the end of the arraylist. Be careful not to cross the "size" of the arraylist. It should return 1 for a successful insertion, and 0 for an unsuccessful insertion. For example,

Sample input	Sample output
Arraylist: insert_last(10);	Returns 1 Arraylist: 10
Arraylist: 10 15 20 25 30 insert_last(35);	Returns 1 Arraylist: 10 15 20 25 30 35

5. int insert_at(int item, int pos) function inserts "item" at the position specified by "pos". Be careful not to cross the "size" of the arraylist. It should return 1 for a successful insertion, and 0 for an unsuccessful insertion. For example,

Sample input	Sample output	Explanation
Arraylist: 10 15 20 25 insert_at(35, 2);	Returns 1 Arraylist: 10 15 35 20 25	
Arraylist: 10 15 20 25 30 insert_at(5, 0);	Returns 1 Arraylist: 5 10 15 20 25 30	
Arraylist: 10 15 20 25 30 insert_at(5, -1);	Returns 0 Arraylist: 10 15 20 25 30	Neagtive index
Arraylist: 10 15 20 25 30 insert_at(5, 9);	Returns 0 Arraylist: 10 15 20 25 30	Index greater than length

6. int search(int item) function searches for the entry specified by “item”. If the entry is found the function should return the index of the entry and -1 if the entry is not found. For example,

Sample input	Sample output	Explanation
Arraylist: 10 15 20 25 30 search(25);	3	
Arraylist: 10 15 20 25 30 search(5);	-1	Entry not found

7. int remove_first() function deletes the first entry of the arraylist. It should return 1 for a successful deletion, and 0 for an unsuccessful deletion. For example,

Sample input	Sample output	Explanation
Arraylist: 10 15 20 25 30 remove_first();	Returns 1 Arraylist: 15 20 25 30	
Arraylist: 30 remove_first();	Returns 1 Arraylist:	
Arraylist: remove_first();	Returns 0 Arraylist:	Arraylist empty

8. int remove_last() function deletes the last entry of the arraylist. It should return 1 for a successful deletion, and 0 for an unsuccessful deletion. For example,

Sample input	Sample output	Explanation
Arraylist: 10 15 20 25 30 remove_last();	Returns 1 Arraylist: 10 15 20 25	
Arraylist: 10 remove_last();	Returns 1 Arraylist:	
Arraylist: remove_last();	Returns 0 Arraylist:	Arraylist empty

9. int remove_at(int pos) function deletes the entry at the position specified by “pos”. It should return 1 for a successful deletion, and 0 for an unsuccessful deletion. For example,

Sample input	Sample output	Explanation
Arraylist: 10 15 20 25 30 remove_at(2);	Returns 1 Arraylist: 10 15 25 30	
Arraylist: 10 15 20 25 30 remove_at(0);	Returns 1 Arraylist: 15 20 25 30	
Arraylist: 10 15 20 25 30 remove_at(-1);	Returns 0 Arraylist: 10 15 20 25 30	Negative index
Arraylist: 10 15 20 25 30 remove_at(9);	Returns 0 Arraylist: 10 15 20 25 30	Index greater than length

10. int remove_item_first(int item) function deletes the first occurrence of the entry specified by “item”. It should return 1 for a successful deletion, and 0 for an unsuccessful deletion. For example,

Sample input	Sample output	Explanation
Arraylist: 10 15 20 25 30 remove_item_first(20);	Returns 1 Arraylist: 10 15 25 30	
Arraylist: 10 15 20 15 30 remove_item_first(15);	Returns 1 Arraylist: 10 20 15 30	
Arraylist: 10 15 20 25 30 remove_item_first(35);	Returns 0 Arraylist: 10 15 20 25 30	Entry not found

11. int remove_item_last(int item) function deletes the last occurrence of the entry specified by “item”. It should return 1 for a successful deletion, and 0 for an unsuccessful deletion. For example,

Sample input	Sample output	Explanation
Arraylist: 10 15 20 25 30 remove_item_last(20);	Returns 1 Arraylist: 10 15 25 30	
Arraylist: 10 15 20 15 30 remove_item_last(15);	Returns 1 Arraylist: 10 15 20 30	
Arraylist: 10 15 20 25 30 remove_item_first(35);	Returns 0 Arraylist: 10 15 20 25 30	Entry not found

12. int remove_item_all(int item) function deletes all occurrences of the the entry specified by “item”. It should return 1 for a successful deletion, and 0 for an unsuccessful deletion. For example,

Sample input	Sample output	Explanation
Arraylist: 10 15 20 25 30 remove_item_all(20);	Returns 1 Arraylist: 10 15 25 30	
Arraylist: 10 15 20 15 30 remove_item_all(15);	Returns 1 Arraylist: 10 20 30	
Arraylist: 10 15 20 25 30 remove_item_first(35);	Returns 0 Arraylist: 10 15 20 25 30	Entry not found

13. int replace(int old_item, int new_item) function replaces all occurrences of entry specified by “old_item” with “new_item”. It should return 1 for a successful replacing, and 0 for no replacement. For example,

Sample input	Sample output	Explanation
Arraylist: 10 15 20 25 30 replace(20, 40);	Returns 1 Arraylist: 10 15 40 25 30	
Arraylist: 10 15 20 15 30 replace(15, 45);	Returns 1 Arraylist: 10 45 20 45 30	
Arraylist: 10 15 20 25 30 replace(35, 45);	Returns 0 Arraylist: 10 15 20 25 30	Entry not found

14. void clear() function deletes every entry of the arraylist and brings the arraylist to it’s initial state. For example,

Sample input	Sample output
Arraylist: 10 15 20 25 30 clear();	Arraylist:

Mark distribution:

Ques	1	2	3	4	5	6	7	8	9	10	11	12	13	14	**Total**
Mark	1	1	2	1	2	2	2	1	2	2	2	3	3	1	**25**