

[Home](#) / [My courses](#) / [10 - PRG C++](#) / [Linked List / Свързани списъци](#) / [13-10 Doubly Linked List](#)

Started on Wednesday, 23 February 2022, 12:02 PM

State Finished

Completed on Sunday, 27 February 2022, 2:07 PM

Time taken 4 days 2 hours

Marks 16.00/16.00

Grade **10.00** out of 10.00 (**100%**)



Question 1

Correct

Mark 4.00 out of 4.00

It is necessary to create a doubly linked list with the structure described below:

```
struct NODE {
    int data;
    NODE* prev;
    NODE* next;
};
```

This list is loaded with data from the main part of the program,
but it is necessary to write a function ***reverseList*** which *reverses the order of the elements in the list* .
The function doesn't return anything, it only reverses the order of the elements as:

from 5 <==> 2

to 2 <==> 5

It has one parameter: the double pointer to the head of the list.

```
void reverseList(NODE** head)
```

Answer: (penalty regime: 10, 20, ... %)

```
1 void reverseList(NODE** hd)
2 {
3     NODE* left = *hd;
4     NODE* right = *hd;
5     while (right->next != nullptr)
6         right = right->next;
7     while (left != right && left->prev != right)
8     {
9
10        swap(left->data, right->data);
11        left = left->next;
12        right = right->prev;
13    }
14 }
15 }
```

Test	Expected	Got

Passed all tests! ✓

Correct

Marks for this submission: 4.00/4.00.

Question 2

Correct

Mark 4.00 out of 4.00

It is necessary to create a doubly linked list with the structure described below:

```
struct NODE {  
    int data;  
    NODE* prev;  
    NODE* next;  
};
```

Don't declare the structure in the solution.

This list is not loaded with data from the main part of the program.

It is necessary to write a function **displayList** which displays all the items in the list, separated by " <==> ".

The function does not return anything. It has one parameter: pointer to the head.

Answer: (penalty regime: 10, 20, ... %)

```
1 void displayList(NODE* hd)  
2 {  
3     while (hd->next != nullptr)  
4     {  
5         cout << hd->data << " <==> ";  
6         hd = hd->next;  
7     }  
8     cout << hd->data << endl;  
9 }
```

Test	Expected	Got

Passed all tests! ✓

Correct

Marks for this submission: 4.00/4.00.



Question 3

Correct

Mark 4.00 out of 4.00

The structure below represents a node of a doubly linked list:

```
struct NODE {  
    int data;  
    NODE* prev;  
    NODE* next;  
};
```

Don't declare the structure in the solution.

You have to write the function **insert** which inserts an item in front of the list.

The function returns nothing and has two parameters: a pointer to pointer to the Head and the value of the element (integer).

Note_0: your function should update the head to point to the new element!

Note_1: the list will contain elements when your function gets called.

Answer: (penalty regime: 10, 20, ... %)

```
1 void insert(NODE** head, int a)  
2 {  
3     NODE* cat = newNode(a);  
4     cat->next = *head;  
5     (*head)->prev = cat;  
6     (*head) = cat;  
7 }
```

Test	Expected	Got

Passed all tests! ✓

Correct

Marks for this submission: 4.00/4.00.

Question 4

Correct

Mark 4.00 out of 4.00

It is necessary to create a doubly linked list with the structure described below:

```
struct NODE {
    int data;
    NODE* prev;
    NODE* next;
};
```

Declare the structure in the solution.

This **list is not loaded with data** from the main part of the program,

but it is necessary to write a function **newNode** which add item to the list /this node is the first element in the list/ .

The function returns the **pointer to the inserted element of the list** (the head is equal to tell).

It has one parameter: the value of the element.

Answer: (penalty regime: 10, 20, ... %)

```
1 struct NODE {
2     int data;
3     NODE* prev;
4     NODE* next;
5 };
6 NODE* newNode(int n)
7 {
8     NODE* cnt = new NODE;
9     cnt->data = n;
10    cnt->prev = cnt->next = nullptr;
11    return cnt;
12 }
```

Test	Expected	Got

Passed all tests! ✓

Correct

Marks for this submission: 4.00/4.00.

◀ [13-8 Search element](#)

Jump to...

[13-9 Circular Linked List ▶](#)

