CS 211 Practice Final Exam

How many bytes will be dynamically allocated from the heap by each of the following statements?

Statement	Number of bytes from the **heap**
int* a = new int[5];	
int* b = new int(7);	
int c[4] = {1, 2, 3, 4};	
int* a = &c	

Write statements to release all dynamically allocated spaces back to the **heap**.

Given the following **infix** expressions, convert them into a **postfix** expression.
2+2*5-6
2/2+6*3

Evaluate the **postfix** expressions that you have found

Build a **binary search tree** using the following values in the order provided.

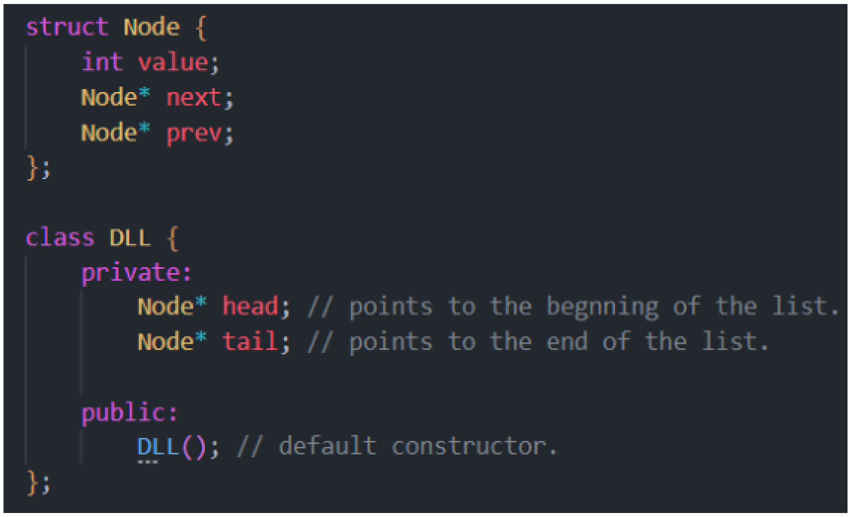
50, 40, 35, 44, 60, 70, 41, 69, 43, 49

In the following space, list the **pre-order, in-order, and post-order traversals** of the tree you have drawn from the previous part.

pre-order	
in-order	
post-order	

Implement a **palindrome** function using recursion using strings. **No loop is allowed**. The function shall return true if the string reads the same forwards and backward, otherwise the function shall return false. Assume all characters in the string are lowercase.

Given the following definition of the **Node** struct and **DLL** class:



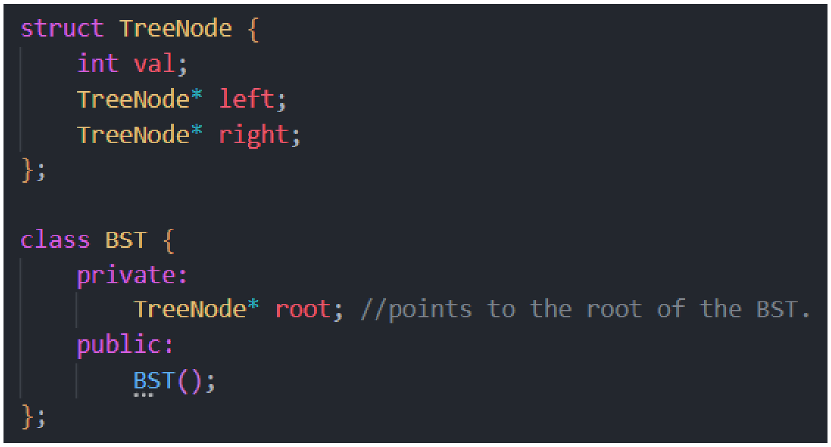
```
struct Node {
    int value;
    Node* next;
    Node* prev;
};

class DLL {
    private:
        Node* head; // points to the begnning of the list.
        Node* tail; // points to the end of the list.

    public:
        DLL(); // default constructor.
};
```

Implement the **reverse** private function using recursion with a doubly linked list. **No loop is allowed**. The function shall reverse the list's contents without dynamically creating new nodes. Also, implement the public function for the reverse as well.

Given the following definition of **TreeNode** struct and **BST** class:

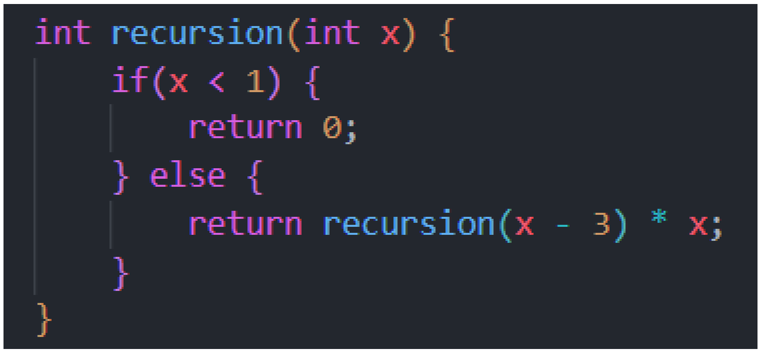


```
struct TreeNode {
    int val;
    TreeNode* left;
    TreeNode* right;
};

class BST {
    private:
        TreeNode* root; //points to the root of the BST.
    public:
        BST();
};
```

Implement the **search** private function using recursion for a Binary Search Tree (BST). **No loop is allowed**. The function shall return true if the value is within the BST, otherwise it shall return false. Also, implement the search public function for the BST class as well.

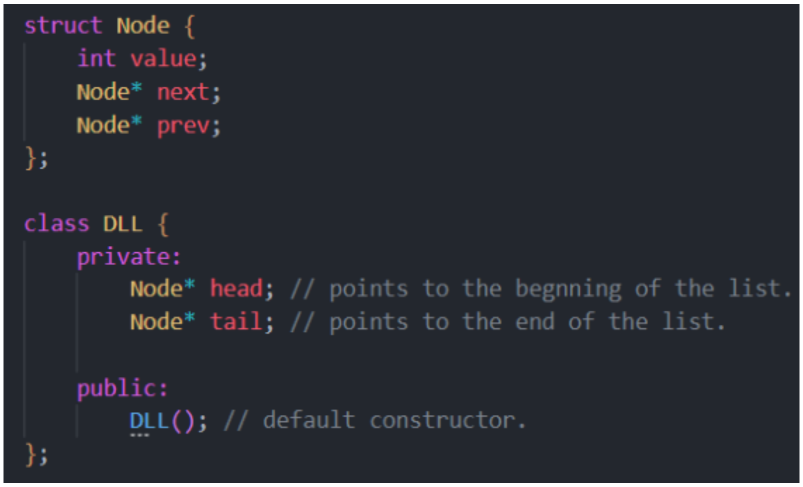
Given the following recursive function:



```
int recursion(int x) {
    if(x < 1) {
        return 0;
    } else {
        return recursion(x - 3) * x;
    }
}
```

In the following space, draw the contents of the call stack when the base case is met for the function all **recursion(8)**. For each stack frame, indicate the value of n. What value would be returned by the function call?

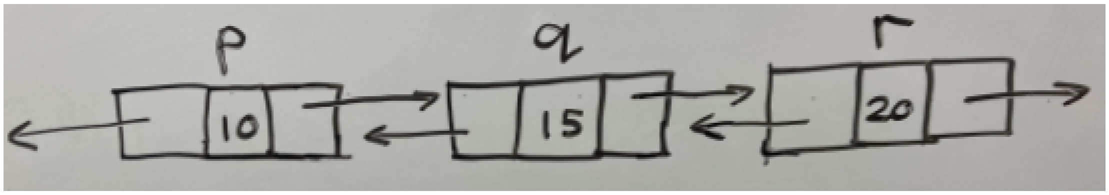
Given the following definition of the **Node** struct and **doubly linked list sequence**:



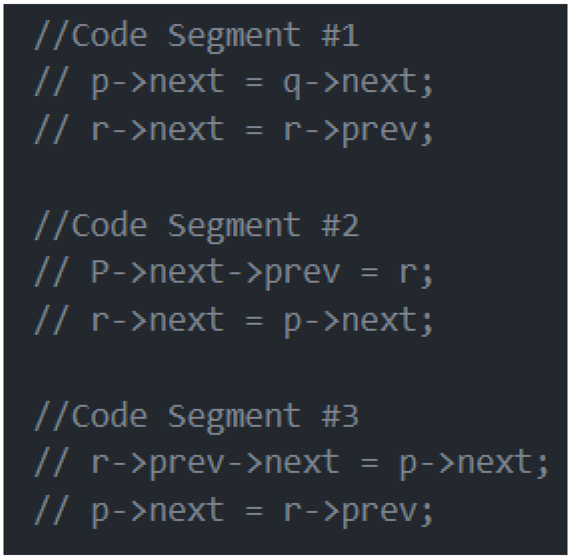
```
struct Node {
    int value;
    Node* next;
    Node* prev;
};

class DLL {
    private:
        Node* head; // points to the begnning of the list.
        Node* tail; // points to the end of the list.

    public:
        DLL(); // default constructor.
};
```



Draw diagrams illustrating how pointers in the linked sequence are adjusted in their code segments. Use the given doubly linked list for each code segment.



```
//Code Segment #1
// p->next = q->next;
// r->next = r->prev;

//Code Segment #2
// P->next->prev = r;
// r->next = p->next;

//Code Segment #3
// r->prev->next = p->next;
// p->next = r->prev;
```