

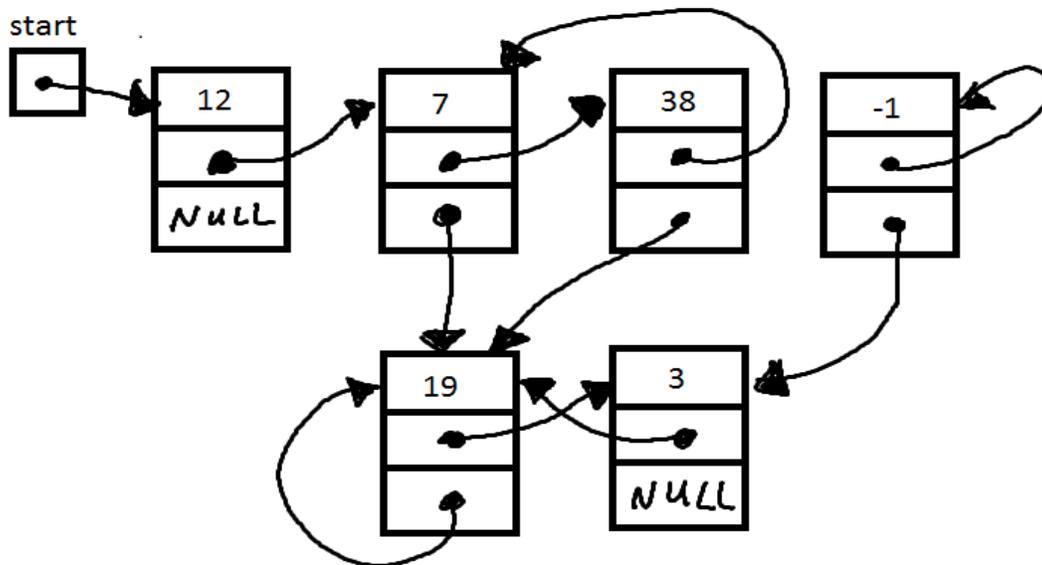
Consider the following data structure and function:

```
typedef struct Node_type Node;
```

```
struct Node_type{  
    int data;  
    Node* next;  
    Node* other;  
};
```

```
Node* makeNode(int dataIn, Node* nextIn, Node* otherIn){  
    Node* newNode = (Node*) malloc(sizeof(Node));  
    newNode->data = dataIn;  
    newNode->next = nextIn;  
    newNode->other = otherIn;  
    return newNode;  
}
```

a) Write a sequence of statements, using the Node data type, to generate the following:



b) Draw the memory picture (similar above) that corresponds to the outcome of the following lines of code:

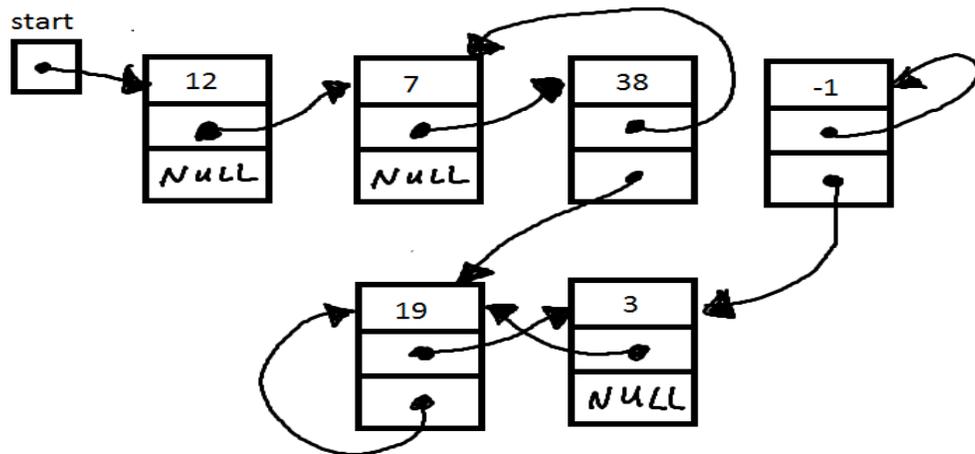
```

Node start = {17, NULL, NULL}; // note: start is different than the
                               // given in part (a)
Node* node1 = makeNode(1, NULL, NULL);
Node* node2 = makeNode(2, NULL, NULL);
Node* node3 = makeNode(3, NULL, NULL);
Node* node4 = makeNode(4, NULL, NULL);
Node* node5 = makeNode(5, NULL, NULL);

start.next = node2;
node2->next = node3;
node3->next = node4;
node3->other = node5;
node4->next = node1;
node5->next = node4;
node1->next = node2;
node1->other = start.next->next->next;
node3->other->other = node2;

```

c) Given the (Node*) start and the following memory diagram, write one line of code that will change the data field that is 19 to -19.



d) in the memory diagram above (assuming all we have is the variable start), are there any memory leaks?