Name: Siddhesh Gupte
UFID: 2975 - 1118
UF Email account: siddhesh.gupte@ufl.edu

Node(String ip_name, int ip_frequency);		
Description	Constructor for individual node	
Parameters	1. Input name 2. Input frequency	1. Name for the node eg: facebook 2. Starting frequency for the node.
Return value	None	

Node fibonacci_heap.insert(Node new_node);		
Description	Insert a new node into the fibonacci heap	
Parameters	1. New node	1.
Return value	Reference of the new node	

Node fibonacci_heap.remove_max()		
Description	Remove the max node from fibonacci heap	
Parameters	None	None
Return value	Reference of the removed max node	

Node fibonacci_heap.increase_frequency(Node node, int new_frequency)		
Description	Increase frequency of a node in the fibonacci heap	
Parameters	1. node 2. New frequency	1. Reference of Node whose frequency is to be increased 2. Value by which the frequency is to be increased
Return value	Reference of the node whose frequency was increased	

void fibonacci_heap.cascading_cut(Node node_to_cut)		
Description	Start a cascading cut from given node	
Parameters	1. Node to cut	1. Node from which the cascading cut starts in the upward direction
Return value	None	

Node fibonacci_heap.merge(Node node1, Node node2)		
Description	Merge two circular doubly linked lists	
Parameters	1. Node 1 2. Node 2	1. First node to merge 2. Second node to merge
Return value	Max node of the merged circular doubly linked lists	

Node fibonacci_heap.test_linkedlist(Node first)		
Description	Utility function to test if a linked list containing the input node is correctly circular	
Parameters	1. First	3. One node of the linked list to check
Return value	None	

Node fibonacci_heap.test_max_node()		
Description	Utility function to peek max node	
Parameters	None	None
Return value	None	

Node fibonacci_heap.test_max_node()	
Description	Utility function to peek max node

Parameters	None	None
Return value	None	