Assignment # 1

Question 1

Q1:Write a function that receives a string consisting of several lines of text and returns arrays indicating top 10

longest unique words and the number of occurrences of each unique word in the text along with their size.

You cannot use string date type; however, you can use char \*.

class swapper

{

public class Node

{

public int detail;

public Node next;

// constructor

public Node(int value, Node next2)

{

detail = value; next = next2;

}

// print list from this

// to last till null

public void printList()

{

Node node = this;

while (node != null)

{

Console.Write(node.detail+" ");

node = node.next;

}

Console.WriteLine();

}

}

// adding a node in the beginning

static Node push(Node head\_ref1, int new\_detail)

{

// allocate node

(head\_ref) = new Node(new\_detail, head\_ref1);

return head\_ref1;

}

static Node swappingnodes(Node head\_ref1, int m, int n)

{

Node head=head\_ref1;

// Nothing to do if m and n are same

if (m == n)

return null;

Node o = null, p = null;

// search for m and n to store their pointer in o and p

while (head\_ref1.next!=null)

{

if ((head\_ref1.next).detail == m)

{

o = head\_ref1;

}

else if ((head\_ref1.next).detail == n)

{

p = head\_ref;

}

head\_ref1 = ((head\_ref1).next);

}

// if o and p is not null swap current pointer and next pointer of these

if (o != null && p != null)

{

Node temp = a.next;

o.next = p.next;

p.next = temp;

temp = o.next.next;

o.next.next = p.next.next;

p.next.next = temp;

}

return head;

}

// Driver code

public static void Main()

{

Node Enter = null;

// linked list is: 11,12,13,14,15,16,17

Enter=push(Enter, 17);

Enter=push(Enter, 16);

Enter=push(Enter, 15);

Enter=push(Enter, 14);

Enter=push(Enter, 13);

Enter=push(Enter, 12);

Enter=push(Enter, 11);

Console.Write("Linked list without swapping ");

Enter.printList();

Enter=swappingnodes(Enter, 14, 12);

Console.Write("Linked list with swapping ");

Enter.printList();

}

}