

Week 8:

Q1:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef struct Node * Nodeptr;
typedef struct Node {
    int data;
    Nodeptr rlink;
    Nodeptr llink;
}Node;

Nodeptr createNode() {
    Nodeptr temp = malloc(sizeof(Node));
    return temp;
}

void insert(Nodeptr *n,int val) {
    if(*n == NULL){
        *n = createNode();
        (*n)->data = val;
        (*n)->llink = (*n)->rlink = *n;
    }
    else{
        Nodeptr temp = *n;
        while(temp->llink != *n)
            temp = temp->llink;
        Nodeptr newNode = createNode();
        newNode->data = val;
        temp->llink = newNode;
        newNode->rlink = temp;
        newNode->llink = *n;
        (*n)->rlink = newNode;
    }
}

Nodeptr inputValue() {
    Nodeptr head;
    char str[100];
    int i;
    printf("Enter a long value: ");
    scanf("%s",str);
    Nodeptr n = createNode();
    n->llink = n->rlink = n;for(i=0;str[i];i++)
    insert(&n,str[i]-'0');
    return n;
}
```

```
Nodeptr calcSum(Nodeptr A, Nodeptr B) {
```

```
    int digit, sum, carry=0;
    Nodeptr head,r,R,a,b;
    a=A->rlink;
    b=B->rlink;
    head = createNode();
    head->llink = head->rlink = head;
    while(a!=A && b!=B){
        sum = a->data + b->data +carry;
        digit = sum%10;
        carry = sum/10;
        insert(&head,digit);
        a=a->rlink;
        b=b->rlink;
    }
    if(a!=A){
        r=a;
        R=A;
    }
    else{
        r=b;
        R=B;
    }
    while(r!=R){
        sum = r->data + carry;
        digit = sum%10;
        carry = sum/10;
        insert(&head,digit);
        r = r->rlink;
    }
    if(carry)
        insert(&head,carry);
    return head;
}
```

```
void display(Nodeptr *n) {
    for(Nodeptr temp=(*n)->rlink;temp!=*n;temp=temp->rlink)
        printf("%d",temp->data);
    printf("\n");
}
```

```
int main() {
    Nodeptr A,B,sum;
    A = inputValue();
    B = inputValue();
    sum = calcSum(A,B);
    printf("Sum : ");
    display(&sum);
    return 0;
}
```

Output:

```
student@dslab: ~/190905216-DSA/Programs/w8
File Edit View Search Terminal Help
student@dslab:~/190905216-DSA/Programs/w8$ gcc q1.c
student@dslab:~/190905216-DSA/Programs/w8$ ./a.out
Enter a long value: 12345678901234567890
Enter a long value: 12345678901234567891
Sum : 24691357802469135781
student@dslab:~/190905216-DSA/Programs/w8$
```

Q2:

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 10

typedef struct node
{
    int key;
    struct node *left, *right;
}* NODE;

typedef struct {
    NODE S[MAX];
    int tos;
}STACK;

NODE createNewNode (int item)
{
    NODE temp = (NODE)malloc(sizeof(struct node));
    temp->key = item;
    temp->left = temp->right = NULL;
    return temp;
}

void push (STACK *s, NODE n) {
    s->S[++(s->tos)] = n;
}
```

```

NODE pop (STACK *s) {
    return s->S[(s->tos)--];
}

```

```

void inorder (NODE root) {

    NODE curr;
    curr = root;
    STACK S;
    S.tos = -1;
    push(&S, root);
    curr = curr->left;
    while (S.tos != -1 || curr != NULL)
    {
        while (curr != NULL)
        {
            push(&S, curr);
            curr = curr->left;
        }
        curr = pop(&S);
        printf("%d\t", curr->key);
        curr = curr->right;
    }
}

```

```

NODE insert (NODE node, int key) {
    if (node == NULL)
        return createNewNode(key);
    if (key < node->key)
        node->left = insert(node->left, key);
    else if (key > node->key)
        node->right = insert(node->right, key);
    return node;
}

```

```

NODE minValueNode (NODE node) {

    NODE current = node;
    while (current && current->left != NULL)
        current = current->left;
    return current;
}

```

```

NODE deleteNode (NODE root, int key) {

    if (root == NULL)
        return root;
    if (key < root->key)
        root->left = deleteNode(root->left, key);
    else if (key > root->key)

```

```

        root->right = deleteNode(root->right, key);
    else
    {
        if (root->left == NULL){
            NODE temp = root->right;
            free(root);
            return temp;
        }
        else if (root->right == NULL)
        {
            NODE temp = root->left;
            free(root);
            return temp;
        }
        NODE temp = minValueNode(root->right);
        root->key = temp->key;
        root->right = deleteNode(root->right, temp->key);
    }
    return root;
}

```

```

void main() {

    NODE root = NULL;
    int k;
    printf("Enter the Root node:\t");
    scanf("%d", &k);
    root = insert(root, k);
    int ch;
    int flag=1;
    do
    {
        printf("\n-----\nEnter your choice:");
        printf("\n1. Insert\n2. Delete\n3. Display\n4. Exit\nYour Choice: ");
        scanf("%d", &ch);
        switch (ch)
        {
            case 1:
                printf("Enter element to be inserted:\t");
                scanf("%d", &k);
                root = insert(root, k);
                break;
            case 2:
                printf("Enter element to be deleted:\t");
                scanf("%d", &k);
                root = deleteNode(root, k);
                break;
            case 3:
                inorder(root);
                break;
            case 4:
                flag=0;

```

```

        printf("exiting.....");
default:
        printf("\nWrong Choice. Try Again..\n");
    }
} while (flag);
}

```

Output:

```

File Edit View Search Terminal Help
student@dslab:~/190905216-DSA/Programs/w8$ gcc q2.c
student@dslab:~/190905216-DSA/Programs/w8$ ./a.out
Enter the Root node: 5

-----
Enter your choice:
1. Insert
2. Delete
3. Display
4. Exit
Your Choice: 1
Enter element to be inserted: 3

-----
Enter your choice:
1. Insert
2. Delete
3. Display
4. Exit
Your Choice: 1
Enter element to be inserted: 9

-----
Enter your choice:
1. Insert
2. Delete
3. Display
4. Exit
Your Choice: 1
Enter element to be inserted: 1

-----
Enter your choice:
1. Insert
2. Delete
3. Display
4. Exit
Your Choice: 1
Enter element to be inserted: 7

-----
Enter your choice:
1. Insert
2. Delete
3. Display
4. Exit
Your Choice: 3
1      3      5      7      9

-----
Enter your choice:
1. Insert
2. Delete
3. Display
4. Exit
Your Choice: 2
Enter element to be deleted: 5

-----
Enter your choice:
1. Insert
2. Delete
3. Display
4. Exit
Your Choice: 3
1      3      7      9

-----
Enter your choice:
1. Insert
2. Delete
3. Display
4. Exit
Your Choice: 4
exiting.....

```