

a) Create a Doubly Linked List.

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

struct Node {
    int data;
    struct Node* next;
    struct Node* prev;
};

struct Node* createNode (int data) {
    struct Node* newNode = (struct Node*) malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->next = NULL;
    newNode->prev = NULL;
    return newNode;
}

void insertNode (struct Node** head, int data) {
    struct Node* newNode = createNode(data);
    newNode->next = *head;
    if (*head != NULL) {
        (*head)->prev = newNode;
    }
    *head = newNode;
}

void printList (struct Node* node) {
    while (node != NULL) {
        printf("%d ", node->data);
        node = node->next;
    }
    printf("\n");
}

int main() {
    struct Node* head = NULL;
    insertNode(&head, 100);
    insertNode(&head, 200);
    insertNode(&head, 500);
    printf("Double Linked List is ");
    printList(head);
}
```

Output:

Doubly Linked List: 500 200

Insertion & Deletion

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

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

```
struct Node {
```

```
    int data;
```

```
    struct Node* prev;
```

```
    struct Node* next;
```

```
};
```

```
void insertNode(struct Node* existNode, int newData) {
```

```
    struct Node* newNode = (struct Node*) malloc(sizeof(struct Node));
```

```
    newNode->data = newData;
```

```
    newNode->prev = existNode->prev;
```

```
    newNode->next = existNode;
```

```
    existNode->prev->next = newNode;
```

```
    existNode->prev = newNode;
```

```
}
```

```
void delNode(struct Node** head, int value) {
```

```
    struct Node* current = *head;
```

```
    while (current != NULL && current->data != value) {
```

```
        current = current->next;
```

```
    }
```

```
    if (current->prev != NULL) {
```

```
current
```

```
        printf(value);
```

```
        return;
```

```
    }
```

```
    if (current->prev != NULL) {
```

```
        current->prev->next = current->next;
```

```
    } else {
```

```
        *head = current->next;
```

```
    }
```

```
    if (current->next != NULL) {
```

```
        current->next->prev = current->prev;
```

```
    }
```

```
    free(current);
```



```

void printList (struct Node* node) {
    while (node != NULL) {
        printf ("%d", node->data);
        node = node->next;
    }
    printf ("\n");
}

```

```

int main() {
    struct Node* head = NULL;
    int reqNodes;
printf ("Enter no. of nodes to be created: ");
scanf ("%d", &reqNodes);
    for (int i=1; i<=10; i++) {
        struct Node* newNode = (struct Node*) malloc (sizeof (struct Node));
        newNode->data = i;
        newNode->prev = NULL;
        newNode->next = head;
        if (head != NULL) {
            head->prev = newNode;
        }
        head = newNode;
    }
    printf ("Original List: ");
    printList (head);
    insertNode (head->next, 11);
    printf ("After inserting: ");
    printList (head);
    delNode (&head, 5);
    printf ("After Deleting: ");
    printList (head);
    return 0;
}

```

Output

~~Original List: 1 2 3 4~~

~~After Insertion: 4 11 3 2 1~~

~~After Deleting:-~~

Original List: 5 4 3 2 1

After Inserting: 5 11 4 3 2

After Deleting: 11 4 3 2 1

"C:\Users\User\Desktop\Data Structures\Stackss.exe"

10 20 30 40

10 30 40

Process returned 0 (0x0) execution time : 0.009 s

Press any key to continue.