Day-3

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

struct node {
    int data;
    struct node *next;
};

struct node *head = NULL;
struct node *tail = NULL;

void int_beg(int num) {
    struct node *n = (struct node*)malloc(sizeof(struct node));
    n->data = num;
    n->next = NULL;
    if (head == NULL) {
        head = n;
        tail = n;
    } else {
        n->next = head;
        head = n;
    }
}

void int_end(int num) {
    struct node *n = (struct node*)malloc(sizeof(struct node));
    n->data = num;
    n->next = NULL;
    if (head == NULL) {
        head = n;
        tail = n;
    } else {
        tail->next = n;
        tail = n;
    }
}

void int_mid(int num, int mid_data) {
    struct node *t = head;
    while (t != NULL) {
        if (t->data == mid_data) {
            struct node *n = (struct node*)malloc(sizeof(struct node));
            n->data = num;
            n->next = t->next;
            t->next = n;
            if (t == tail) {
                tail = n;
            }
            break;
        }
        t = t->next;
    }
}

void del_beg() {
    if (head == NULL) {
```

```c
        return;
    }
    struct node *temp = head;
    head = head->next;
    free(temp);
}

void del_end() {
    if (head == NULL) {
        return;
    }
    struct node *prev = NULL;
    struct node *current = head;
    while (current->next != NULL) {
        prev = current;
        current = current->next;
    }
    if (prev != NULL) {
        prev->next = NULL;
    }
    if (current == head) {
        head = NULL;
    }
    tail = prev;
    free(current);
}

void del_mid(int mid_data) {
    struct node *prev = NULL;
    struct node *current = head;
    while (current != NULL && current->data != mid_data) {
        prev = current;
        current = current->next;
    }
    if (current != NULL) {
        if (current == head) {
            head = head->next;
        } else {
            prev->next = current->next;
        }
        if (current == tail) {
            tail = prev;
        }
        free(current);
    }
}

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

void search(int key) {
    struct node *t = head;
    while (t != NULL) {
        if (t->data == key) {
```

```c
        }
        t = t->next;
    }

}

void sort() {
    struct node *current = head;
    struct node *index = NULL;
    int temp;
    if (head == NULL) {
        return;
    }
    while (current != NULL) {
        index = current->next;
        while (index != NULL) {
            if (current->data > index->data) {
                temp = current->data;
                current->data = index->data;
                index->data = temp;
            }
            index = index->next;
        }
        current = current->next;
    }
}

int find_max() {
    int max = head->data;
    struct node *t = head->next;
    while (t != NULL) {
        if (t->data > max) {
            max = t->data;
        }
        t = t->next;
    }
    return max;
}

int find_min() {
    int min = head->data;
    struct node *t = head->next;
    while (t != NULL) {
        if (t->data < min) {
            min = t->data;
        }
        t = t->next;
    }
    return min;
}

int main() {
    printf("name=kongara sai\n");
    printf("reg no=192365025\n");
    int_beg(3);
    int_beg(5);
    int_beg(9);
    int_mid(3, 6);
    int_end(5);
    display();
    printf("----\n");
```

```c
    del_beg();
    del_end();
    del_mid(3);
    display();

    printf("Search for 6: ");
    search(6);
    if (head != NULL)
        printf("Element found\n");
    else
        printf("Element not found\n");

    sort();
    printf("Sorted List: ");
    display();

    printf("Maximum value in the list: %d\n", find_max());
    printf("Minimum value in the list: %d\n", find_min());

    return 0;
}
```
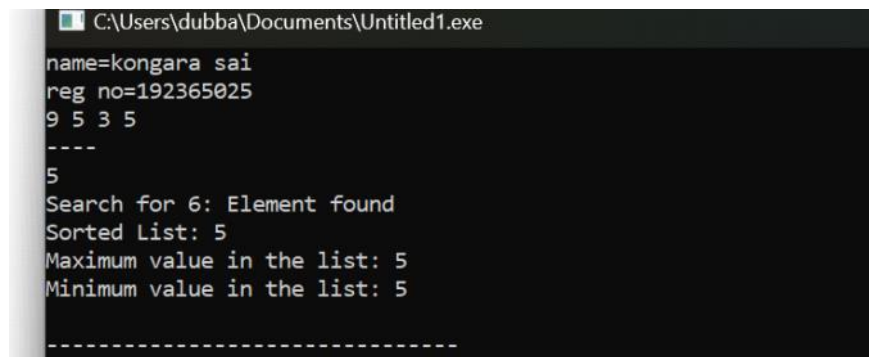


```
C:\Users\dubba\Documents\Untitled1.exe

name=kongara sai
reg no=192365025
9 5 3 5
----
5
Search for 6: Element found
Sorted List: 5
Maximum value in the list: 5
Minimum value in the list: 5

--------------------------------
```