



IT210 Odd 2021 Mini Project

---

## LEKON - HOSTEL MANAGEMENT

<<DO NOT DELETE THIS TABLE FORMAT>>

<<The following points are mandatory, you may add additional points if needed>>

<<Upload the word file in the Moodle, not the PDF>>

<<Each of the Group members in the same group needs to submit the SAME report in the Moodle, any one will be evaluated>>

### *What is the Application?*

Hostel management is a real life problem like when i am came to the college with my friends i am a hosteller i want a room to stay in the college so i go to the hostel office for room allotment there allocating the rooms to the students there asking the details and there allocating room number wise sequentially . I observed the way there allocating the rooms it is like that for every room there want to fill the details of students and allocate the room to him after there close that room section and open next new room to allocate to new students and if any student want to vacate the room there just erase the details of the student and putting it for new room to allocate to some other new student and if there search to know the student details there just enter the room number or roll number the all the details of the student are displaying on the screen and there updating the rooms by erasing the previous student in the room and allocating the room to the new student .the

**NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA**  
**NH 66, Srinivas Nagar, Surathkal, Mangaluru, Karnataka 575025, India**



**IT210 Odd 2021 Mini Project**

software use in my hostel office for allocation of rooms to the students is our mini project



***What is a Possible Solution?***

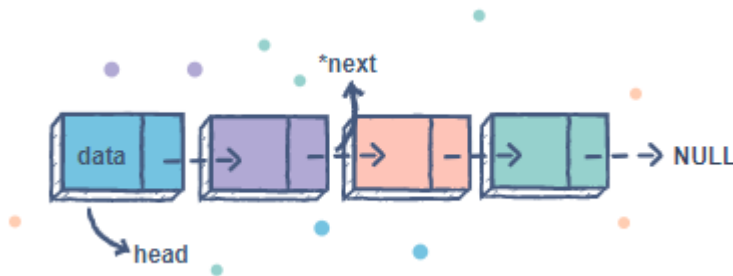
*(Minimum 150 words or more, include figure and tables if needed)*

We are creating the hostel management system by using the Basic linked list in c - language . We are taking all the functionalities of the hostel office as the functions in our program. For allocating the rooms for the students we are taking each node as the one room we are filling the details of the student in the nodes data and inserting another node for the new room allotment in the sequentially . by using traverse we can go to each node



### IT210 Odd 2021 Mini Project

so we can go through each student details respective rooms



insert(rollnumber, name, phone, roomno) :

this function use to inserting every node and node data to the head it is like allocating next room to the new student

search(roomno) :

this function use to searching the student details by taking the room number as the key by traversing every node in the linked list

Delete(roomno) :

this function use to vacate the already allocated room by erasing the details of previous student details and reading for new allotment here key is room number

update(roomno) :

this function use to update or reallocate the already allocated room to the new student by overwriting the data in the nodes

search1(rollnumber) :

this function use to searching the student details by taking the roll number as the key by traversing every node in the linked list

display() :

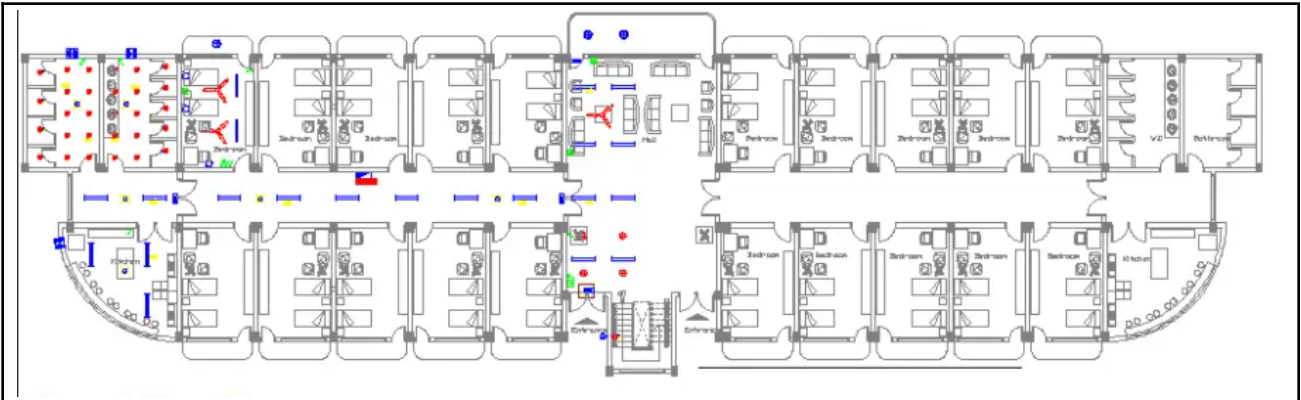
this function use to display all the students details who allocated the room by traversing through each node in the linked list

by observing from above we take all the functionalities of hostel office as the user defined functions in the our hostel management program

**NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA**  
**NH 66, Srinivas Nagar, Surathkal, Mangaluru, Karnataka 575025, India**



**IT210 Odd 2021 Mini Project**



***Which Data Structure to Use for the Solution (Details of the Data Structure)***

*(Minimum 150 words or more, include figure and tables if needed)*

We use the simple singly linked list for creating the hostel management program. A **singly linked list** is a type of linked list that is unidirectional, that is, it can be traversed in only one direction from head to the last node (tail). Each element in a linked list is called a **node**. A single node contains data and a pointer to the next node which helps in maintaining the structure of the list. The first node is called the **head**; it points to the first node of the list and helps us access every other element in the list. The last node, also sometimes called the **tail**, points to NULL which helps us in determining when the list ends.

You can determine and retrieve a specific node either from the front, the end, or anywhere in the list.

The worst case **Time Complexity** for retrieving a node from anywhere in the list is  $O(n)$ .

**NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA**  
**NH 66, Srinivas Nagar, Surathkal, Mangaluru, Karnataka 575025, India**



**IT210 Odd 2021 Mini Project**



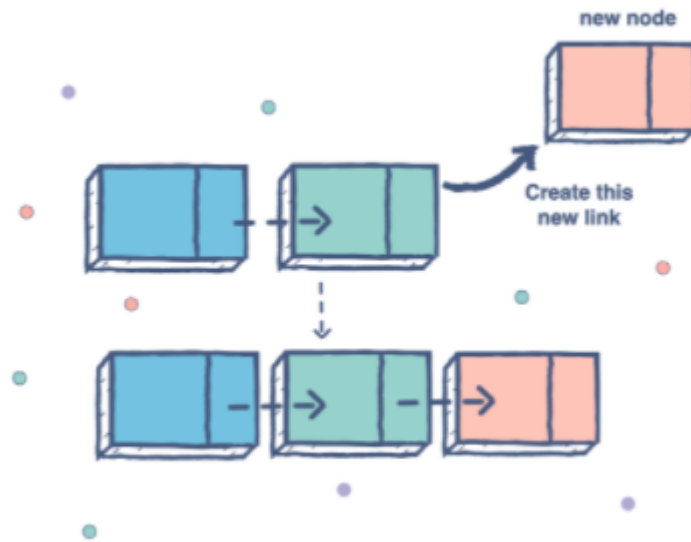
You can add a node at the front, the end or anywhere in the linked list.

The worst case **Time Complexity** for performing these operations is as follows:

- Add item to the front of the list:  $O(1)$
- Add item to the end of the list:  $O(n)$
- Add item to anywhere in the list:  $O(n)$



### IT210 Odd 2021 Mini Project

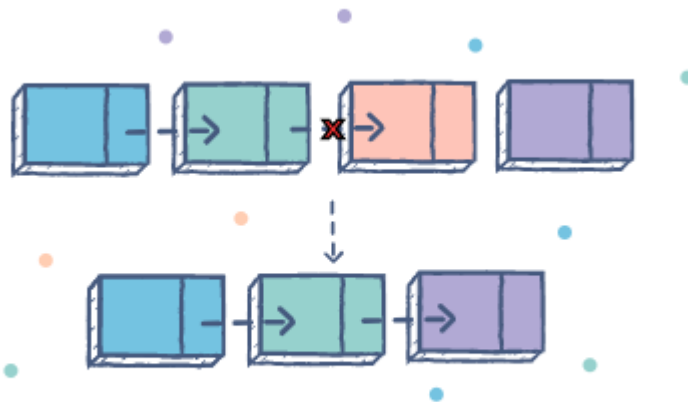


You can remove a node either from the front, the end or from anywhere in the list. The worst case **Time Complexity** for performing this operation is as follows:

- Remove item from the front of the list:  $O(1)$
- Remove item from the end of the list:  $O(n)$
- Remove item from anywhere in the list:  $O(n)$



### IT210 Odd 2021 Mini Project



#### *Justify Why This is the Best Data Structure for this Application?*

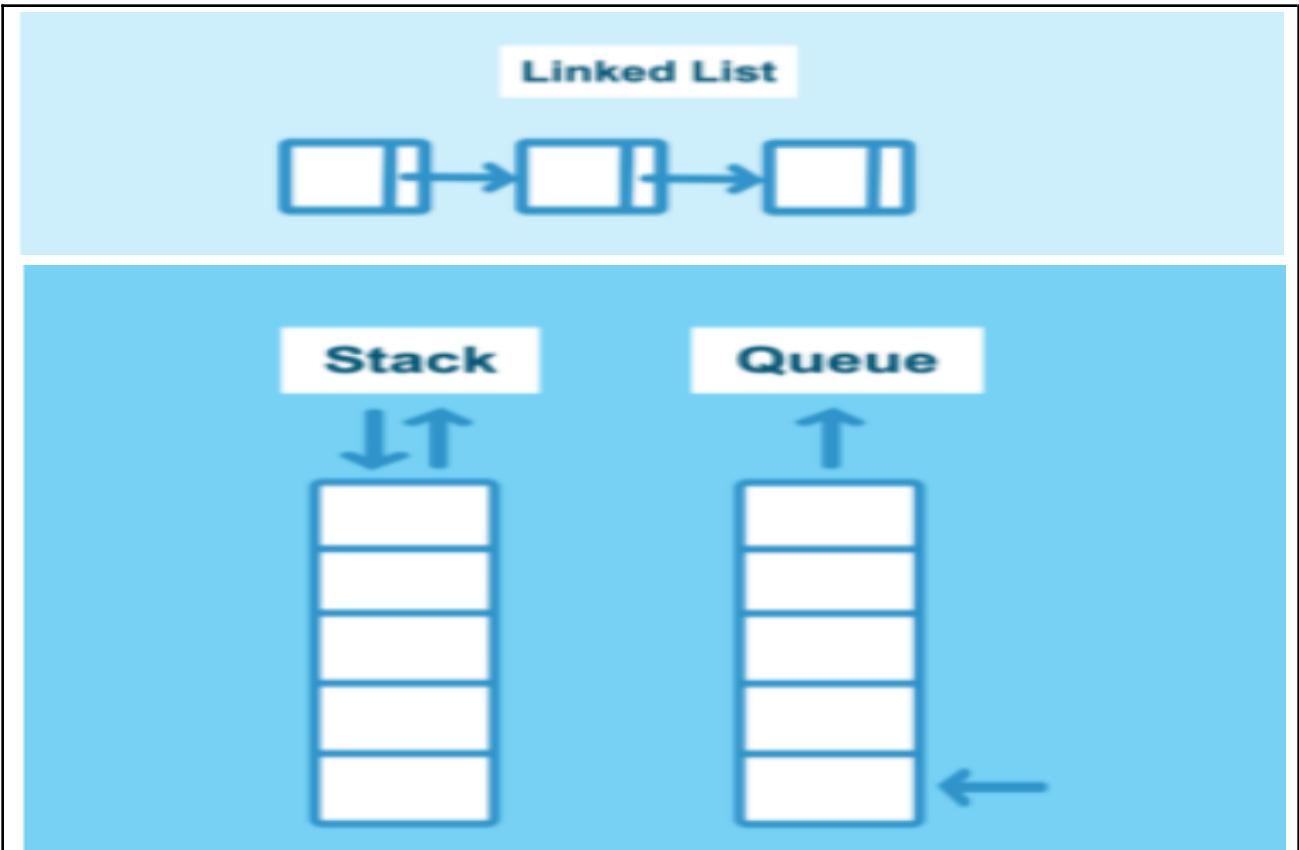
Here we use the singly linked list for creating the hostel management system for justifying this is the best data structure first we want to know the key difference between the singly linked list, stack and queue

stack is a linked list that allows insertion / removal only from its tail

queue is a linked list that allows insertion only at its tail and removal only from its head.



### IT210 Odd 2021 Mini Project



Stacks are based on the LIFO principle, i.e., the element inserted at the last, is the first element to come out of the list

Queues are based on the FIFO principle, i.e., the element inserted at the first, is the first element to come out of the list.

but in the singly Linked list we can insert the element in the middle of the linked list. This is the key reason for selecting the singly linked list as our primary data structure because by using the stack or queue we cannot update or vacate the rooms in the middle hostel rooms . we can only able to change the front and rear rooms means 1st or last rooms by using the stack or queue

*Implementation Details (Experimental Setup)*



**NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA**  
**NH 66, Srinivas Nagar, Surathkal, Mangaluru, Karnataka 575025, India**



**IT210 Odd 2021 Mini Project**

*(150-200 words approx.)*

With the basic knowledge of operations on Linked Lists like insertion, search, traversing of elements in the Linked list, the hostel management and room allotment system can be created. Below are the functionalities explained that are to be implemented

**Allotting the Room:** It is as simple as creating a new node in the Empty Linked list or inserting a new node in a non-Empty linked list.

**Search according to roll number:** Search a Record is similar to searching for a key in the linked list. Here in the Hostel management key is the roll number as the roll number is unique for every student in the hostel.

**Search according to room number:** Search a Record is similar to searching for a key in the linked list. Here in the Hostel management key is the room number as the room number is unique for every room in the hostel.

**Vacating the room :** By traversing we can go to certain node and empty the allotted room details and ready it for new allotment

**Displaying**

**allotted rooms details :** by traversing we can Print all details elements of the Linked list.

**Updation of room :** by traversing we can go to certain node and reallocate the room to the new student like reassign

**CODE:**

**NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA**  
**NH 66, Srinivas Nagar, Surathkal, Mangaluru, Karnataka 575025, India**



**IT210 Odd 2021 Mini Project**

```
/* MINI PROJECT ON HOSTEL MANAGEMENT */
/* PROGRAM IN C-LANGUAGE */
/* LEKON - HOSTEL MANAGEMENT */
#include<stdlib.h>
#include<string.h>
#include<stdio.h>
struct Student
{
    int rollnumber;
    char name[100];
    char phone[100];
    int roomno;
    struct Student *next;
}
}*head;
void insert(int rollnumber, char* name, char* phone,int roomno)
{
    struct Student *temp;
    struct Student * student = (struct Student *) malloc(sizeof(struct Student));
    student->rollnumber = rollnumber;
    student->roomno = roomno;
    strcpy(student->name, name);
    strcpy(student->phone, phone);
    student->next = NULL;

    if(head==NULL){
        head = student;
        temp = head;
    }
    else{
        temp->next= student;
        temp = temp->next;
    }
}
```

**NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA**  
**NH 66, Srinivas Nagar, Surathkal, Mangaluru, Karnataka 575025, India**



**IT210 Odd 2021 Mini Project**

```
}
void search(int roomno)
{
    struct Student * temp = head;
    while(temp!=NULL){
        if(temp->roomno==roomno){
            printf("Room number      : %d\n", temp->roomno);
            printf("Name of the student: %s\n", temp->name);
            printf("Roll Number      : %d\n", temp->rollnumber);
            printf("Phone          : %s\n", temp->phone);
            return;
        }
        temp = temp->next;
    }
    printf("This room is not allotted to any student this room is empty");
}
void search1(int rollnumber)
{
    struct Student * temp = head;
    while(temp!=NULL){
        if(temp->rollnumber==rollnumber){
            printf("Room number      : %d\n", temp->roomno);
            printf("Name of the student: %s\n", temp->name);
            printf("Roll Number      : %d\n", temp->rollnumber);
            printf("Phone          : %s\n", temp->phone);
            return;
        }
        temp = temp->next;
    }
    printf("Student with roll number %d is not found !!!\n", rollnumber);
}
void update(int roomno)
{
    struct Student * temp = head;
```

**NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA**  
**NH 66, Srinivas Nagar, Surathkal, Mangaluru, Karnataka 575025, India**



**IT210 Odd 2021 Mini Project**

```
while(temp!=NULL)
{
    if(temp->roomno==roomno)
    {
        printf("Record with room number %d Found !!!\n", roomno);
        printf("Enter new student name: ");
        scanf("%s", temp->name);
        printf("Enter new roll number : ");
        scanf("%d",&temp->rollnumber);
        printf("Enter new phone number: ");
        scanf("%s", temp->phone);
        printf("Updation Successful!!!\n");
        return;
    }
    temp = temp->next;
}
printf("room with this room number %d is not allocated to any body !!!\n", roomno);
}
void Delete(int roomno)
{
    struct Student * temp = head;
    while(temp!=NULL)
    {
        if(temp->roomno==roomno)
        {
            printf("Record with room number %d Found !!!\n", roomno);
            strcpy(temp->name," ");
            temp->rollnumber=0;
            strcpy(temp->phone," ");
            printf("Successfully Vacated the room with room number '%d' ,Now this room is
ready for new allotmentment !!!\n",roomno);
            return;
        }
    }
}
```

**NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA**  
**NH 66, Srinivas Nagar, Surathkal, Mangaluru, Karnataka 575025, India**



**IT210 Odd 2021 Mini Project**

```
        temp = temp->next;
    }
    if(roomno>temp->roomno)
    {
        printf("Already room number %d is vacate !!!\n",roomno);
    }
}
void display()
{
    struct Student * temp = head;
    while(temp!=NULL){
        printf("\nRoom number      : %d\n",temp->roomno);
        printf("Name of the student: %s\n", temp->name);
        printf("Roll Number      : %d\n", temp->rollnumber);
        printf("Phone          : %s\n", temp->phone);
        temp = temp->next;
    }
}
int main()
{
    head = NULL;
    int roomno=0;
    int choice;
    char name[100];
    char phone[100];
    int rollnumber;
    printf("\n\t\t\t\t\t *****");
    printf("\n\t\t\t\t\t **LEKON-HOSTEL MANAGEMENT**");
    printf("\n\t\t\t\t\t *****");
    printf("\n\n\nFOLLOW THE BELOW INSTRUCTIONS :- \n");
    printf("***** ** ***** ");
    printf("\n=> to Allot a room to the student - 1\n=> to search Room details      -
2\n=> to Vacate the room                - 3\n=> to update Room allotment      - 4\n=> to
```

**NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA**  
**NH 66, Srinivas Nagar, Surathkal, Mangaluru, Karnataka 575025, India**



**IT210 Odd 2021 Mini Project**

```
display all the Room details - 5\n=> to search for student details - 6\n=> to quit
- 7\n");
l:
printf("\nEnter Choice: ");
scanf("%d", &choice);
switch (choice)
{
    case 1:
        printf("\nAlloted Room number      : %d\n", ++roomno);
        printf("Enter name of the student : ");
        scanf("%s", name);
        printf("Enter roll number      : ");
        scanf("%d", &rollnumber);
        printf("Enter phone number      : ");
        scanf("%s", phone);
        insert(rollnumber, name, phone, roomno);
        goto l;
        break;
    case 2:
        printf("Enter room number to search: ");
        scanf("%d", &roomno);
        search(roomno);
        goto l;
        break;
    case 3:
        printf("Enter room number to vacate the room: ");
        scanf("%d", &roomno);
        Delete(roomno);
        goto l;
        break;
    case 4:
        printf("Enter room number to update: ");
        scanf("%d", &roomno);
        update(roomno);
        goto l;
}
```

**NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA**  
**NH 66, Srinivas Nagar, Surathkal, Mangaluru, Karnataka 575025, India**



**IT210 Odd 2021 Mini Project**

```
        break;
    case 5:
        display();
        goto l;
        break;
    case 6:
        printf("Enter roll number to search: ");
        scanf("%d", &rollnumber);
        search1(rollnumber);
        goto l;
        break;
    case 7: exit(0);

    default: printf("WRONG CHOICE IS ENTERED PLEASE TRY AGAIN");
            goto l;
        }
    }
```

**OUTPUT:**

**NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA**  
**NH 66, Srinivas Nagar, Surathkal, Mangaluru, Karnataka 575025, India**



**IT210 Odd 2021 Mini Project**

```
*****
**LEKON-HOSTEL MANAGEMENT**
*****

FOLLOW THE BELOW INSTRUCTIONS :-
*****
-> to Allot a room to the student - 1
-> to search Room details       - 2
-> to Vacate the room           - 3
-> to update Room allotment     - 4
-> to display all the Room details - 5
-> to search for student details - 6
-> to quit                      - 7

Enter Choice: 1

Alloted Room number      : 1
Enter name of the student : kamal
Enter roll number        : 234
Enter phone number       : 98765432

Enter Choice: 1

Alloted Room number      : 2
Enter name of the student : vivek
Enter roll number        : 56789
Enter phone number       : 0987654321

Enter Choice: 1

Alloted Room number      : 3
Enter name of the student : akshita
Enter roll number        : 876
Enter phone number       : 09875678432

Enter Choice: 2
Enter room number to search: 2
Room number              : 2
Name of the student: vivek
Roll Number              : 56789
Phone                    : 0987654321

Enter Choice: 3
Enter room number to vacate the room: 1
Record with room number 1 Found !!!
Successfully Vacated the room with room number '1' ,Now this room is ready for new allotment !!!

Enter Choice: 5

Room number              : 1
Name of the student:
Roll Number              : 0
Phone                    :

Room number              : 2
Name of the student: vivek
Roll Number              : 56789
Phone                    : 0987654321

Room number              : 3
Name of the student: akshita
Roll Number              : 876
Phone                    : 09875678432
```



**NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA**  
**NH 66, Srinivas Nagar, Surathkal, Mangaluru, Karnataka 575025, India**



**IT210 Odd 2021 Mini Project**

```
Enter Choice: 4
Enter room number to update: 1
Record with room number 1 Found !!!
Enter new student name: srinivas
Enter new roll number : 6789
Enter new phone number: 3456789023
Updation Successful!!!

Enter Choice: 5

Room number      : 1
Name of the student: srinivas
Roll Number      : 6789
Phone            : 3456789023

Room number      : 2
Name of the student: vivek
Roll Number      : 56789
Phone            : 0987654321

Room number      : 3
Name of the student: akshita
Roll Number      : 876
Phone            : 09875678432

Enter Choice: 6
Enter roll number to search: 876
Room number      : 3
Name of the student: akshita
Roll Number      : 876
Phone            : 09875678432

Enter Choice: 7
```

***Complexity Analysis***  
*(100 words approx.)*

Let 'n' be the number of students allocated in n rooms

**Allotting the Room :**

if we want to allocate n rooms for n students we want to traverse till the nth room  
=> Worst Time Complexity :  $O(n)$

**Search according to roll number:**

if we want to search details of a student according to a roll number who allocate the last room we want to traverse till the nth room that is the worst case

=> Worst Time Complexity :  $O(n)$

**NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA**  
**NH 66, Srinivas Nagar, Surathkal, Mangaluru, Karnataka 575025, India**



**IT210 Odd 2021 Mini Project**

if we want to search details of a student according to a roll number who allocate the 1st room then no need of traverse that is the best case

=> Best Time Complexity :  $O(1)$

**Search according to room number:**

if we want to search details of a student according to a room number who allocate the last room we want to traverse till the nth room that is the worst case

=> Worst Time Complexity :  $O(n)$

if we want to search details of a student according to a room number who allocate the 1st room then no need of traverse that is the best case

=> Best Time Complexity :  $O(1)$

**Vacating the room :**

if we want to vacate the last room we want to traverse till the nth room and remove all the details that is the worst case

=> Worst Time Complexity :  $O(n)$

if we want to vacate the 1st room then no need of traverse that is the best case

=> Best Time Complexity :  $O(1)$

**Displaying allotted rooms details :**

we want to display all the room details means till the n rooms then we want to traverse till the nth node

=>here all Time Complexity :  $O(n)$

**Updation of room :**

if we want to update the nth room details we want to traverse till the nth room and update the details that is the worst case

=> Worst Time Complexity :  $O(n)$

if we want to update the 1st room then no need of traverse that is the best case

=> Best Time Complexity :  $O(1)$

***Observations and Conclusions***

*(100 words approx.)*

**NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA**  
**NH 66, Srinivas Nagar, Surathkal, Mangaluru, Karnataka 575025, India**



**IT210 Odd 2021 Mini Project**

Finally we can allocate  $n$  rooms for  $n$  students in college. In our program we not delete any node we just vacate the node by replacing previous data with NULL why because deleting the node means deleting the room it's not possible just we want to erase the previous room data and make it ready for new allotment if we want to delete the node we again change the next room person to the before room like we want shift till the last in real world this work become complex and not possible that's why we vacate the room means erasing the details put it for the new allotment so other node means no other room students not disturbed this is the thing we observe in our program, with our program we did maximum functionalities like hostel office software. we concluded that we want to update our program by implementing the some other functionalities like issues in hostel rooms, resolving them, mess allocation, fee payments and feedback from the students