

# Hostel Counselling System

Name: Hemaksh Chaturvedi

Registration number: 19BCE2222

Mail id: [hemakshamit.chaturvedi2019@vitstudent.ac.in](mailto:hemakshamit.chaturvedi2019@vitstudent.ac.in)

Slot: B2

---

## Abstract:

**The data structure and algorithm that I have chosen is an Array. In this project I will be explaining what an array is, its implication and uses in the real life, its limitations and a way of overcoming them.**

**By overcome the limitations I will be creating a program for Hostel room allotment using Linked list, Dynamic Memory Allocation and Structures.**

## I. INTRODUCTION

An array is defined as the collection of similar type of data items stored at continuous memory locations. Since the array has a predefined memory-space we cannot add more elements to it if it is filled; neither can the empty spaces in the array be used.

Hence, to overcome these limitations I will be creating a program for Hostel room allotment using **Linked list, Dynamic Memory Allocation and Structures.**

A linked list is a sequence of data structures, which are connected together via links. Linked List is a sequence of links which contains items. Each link contains a connection to another link.

## II. OBJECTIVES

- To overcome the limitation of fixed memory space in an array
- To overcome the limitation of a single data type in an array
- To create a program overcoming these limitations

## III. METHOD USED

- To overcome the limitation of fixed memory space I have used another data structure called **Linked List**
- To overcome the limitation of a single data **Structure**, I have used structures which is a set of different data type.
- The program I am creating using the above mentioned features is called **Hostel Counselling System.**

## IV. AIM

To create a C Language program for a Hostel Counselling System by overcoming the limitations of an array by using Linked list and structure and Structures.

## V. PROGRAM OBJECTIVES

The program has the following objectives: -

- To ask for the students details namely: -
  - First name
  - Registration number
  - CGPA
  - Gender
- After taking these details, to form two separate rank lists, one for male and another for female on the basis of their CGPA.
- After the rank list is generated you can start the Hostel allotment.
- In the Hostel allotment: -
  - We first start with the Men's Hostel Allotment
  - Then go to the Women's Hostel Allotment
  - While allotting, the details of the student are displayed that is their

- name, registration number and CGPA.
  - They are then asked to choose the block they want to live in
  - They are then given the option to opt for an A.C. or Non – A.C. room.
  - After which they must choose the number of sharing they wish to live in that is, 1,2,3 and 6.
  - When they choose the number of sharing they are then allotted room number using a random number generator.
  - This continues until all students are allotted.
- You can then Exit the program.

## VI. SOURCE CODE

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

struct list
{
    char name[30];
    char reg_no[10];
    float CGPA;
    char gender;
    struct list *next;
}*front=NULL,*rear=NULL,*nw,*head=NULL,*tail=NULL;

void block_name()
{
    char b[5];
    printf("\nEnter which block you want: ");
    scanf("%s",&b);
    printf("\nThe block allotted to you is:- %s\n",b);
    return b;
}
```

```
int room_number()
{
    int max,min,n,sharing,type ;

    printf("\nThe following are the options for your
    hostel:- \n\t(1) A.C. room\n\t(2) Non - A.C. room\n
    ");

    scanf("%d",&type);

    if (type<3)
    {
        printf("\nEnter the number of sharing you want:
        \n\t(1) 1 bedded sharing\n\t(2) 2 bedded
        sharing\n\t(3) 3 bedded sharing\n\t(4) 6 bedded
        sharing\n ");

        scanf("%d",&sharing);

        if (sharing <5)
        {
            if (sharing==1)
            {
                max=1143;
                min=1005;
                n=(rand()%(max-min+1))+min;
            }

            if(sharing==2)
            {
                max=900;
                min=700;
                n=(rand()%(max-min+1))+min;
            }

            if (sharing==3)
            {
                max=320;
                min=112;
                n=(rand()%(max-min+1))+min;
            }

            if (sharing==4)
            {
                max=650;
```

```

        min=431;

        n=(rand()%(max-min+1))+min;
    }
}
else{
    printf("It is not an option. RETRY!!\n");
    room_number();
}
printf("\nThe room allotted to you is:- %d\n",n);
}
else
{
    printf("It is not an option. RETRY!!\n");
    room_number();
}
return n;
}

void display()
{
    struct list *temp=front;

    printf("\n\t\t ~~~RANK LIST FOR MEN'S
    HOSTEL ALLOTMENT~~~\n\n");

    while (temp->next!=NULL)
    {
        printf("\nName: %s\nRegistration Number:
        %s\nCGPA: %f\n",temp->name,temp-
        >reg_no,temp->CGPA);

        temp=temp->next;
    }

    printf("\nName: %s\nRegistration Number:
    %s\nCGPA: %f\n",temp->name,temp-
    >reg_no,temp->CGPA);

    printf("\n\t\t ~~~RANK LIST FOR WOMEN'S
    HOSTEL ALLOTMENT~~~\n\n");

    temp=head;

```

```

        while (temp->next!=NULL)
        {
            printf("\nName: %s\nRegistration Number:
            %s\nCGPA: %f\n",temp->name,temp-
            >reg_no,temp->CGPA);

            temp=temp->next;
        }

        printf("\nName: %s\nRegistration Number:
        %s\nCGPA: %f\n",temp->name,temp->reg_no,temp-
        >CGPA);
    }

    void enqueue()
    {
        nw= (struct list*)malloc(sizeof(struct list));

        printf("\nEnter your Name: ");

        //scanf("%o[^\n]%*c",&nam);

        scanf("%s",&nw->name);

        printf("\nEnter you Registration number: ");

        scanf("%s",&nw->reg_no);

        printf("\nEnter your CGPA: ");

        scanf("%f",&nw->CGPA);

        printf("\nEnter your gender [Put 'M' for Male and
        'F' for Female]: ");

        scanf("%s",&nw->gender);

        if (nw->gender=='M' || nw->gender=='m')
        {
            if (front==NULL)
            {

                nw->next=NULL;

                front=nw;

                rear=nw;
            }
            else
            {

```

```

        rear->next=nw;

        rear=nw;

        rear->next=NULL;
    }
}

if(nw->gender=='F' || nw->gender=='f' )
{
    if (head==NULL)
    {

        nw->next=NULL;

        head=nw;

        tail=nw;
    }
    else
    {

        tail->next=nw;

        tail=nw;

        tail->next=NULL;
    }
}

void details()
{
    int k,f;

    while(f=1)
    {
        printf("\nEnter what you want to do:\n\t(1)
Enter student details\n\t(2) Exit\n");

        scanf("%d",&k);

```

```

        switch(k)
        {
            case 1:
                enqueue();

                break;

            case 2:
                return;
        }
    }

    /* Bubble sort the given linked list */

    void bubbleSort()
    {
        int swapped, i;

        struct list *ptr1;

        struct list *lptr = NULL;

        struct list *ptr2;

        struct list *lptr2=NULL;

        //SORTING MENS LIST

        /* Checking for empty list */

        if (front == NULL)

            return;

        else if (front->next==NULL)

            return;

        do
        {
            swapped = 0;

            ptr1 = front;

            while (ptr1->next != lptr)
            {

```

```

        if (ptr1->CGPA < ptr1->next->CGPA)
        {
            swap(ptr1, ptr1->next);
            swapped = 1;
        }
        ptr1 = ptr1->next;
    }
    lptr = ptr1;
}

while (swapped);

// SORTING WOMEN LIST

if (head == NULL)
    return;

else if (head->next==NULL)
    return;

do
{
    swapped = 0;
    ptr2 = head;

    while (ptr2->next != lptr2)
    {
        if (ptr2->CGPA < ptr2->next->CGPA)
        {
            swap(ptr2, ptr2->next);
            swapped = 1;
        }
        ptr2 = ptr2->next;
    }
    lptr2 = ptr2;
}

while (swapped);
}

/* function to swap data of two nodes a and b*/
void swap(struct list *a, struct list *b)
{
    float temp = b->CGPA;
    b->CGPA = a->CGPA;
    a->CGPA = temp;

    char temp2[10];
    strcpy(temp2,b->reg_no);
    strcpy(b->reg_no,a->reg_no);
    strcpy(a->reg_no,temp2);

    char temp3[30];
    strcpy(temp3, b->name);
    strcpy(b->name,a->name);
    strcpy(a->name,temp3);
}

void asking_for_mens_hostel()
{
    struct list *temp4=front;

    printf("\n\t\t ~~MEN'S HOSTEL
    ALLOTMENT~~\n\n");

    while (temp4->next!=NULL)
    {
        printf("\nName: %s\nRegistration Number:
        %s\nCGPA: %f\n",temp4->name,temp4->reg_no,temp4->CGPA);

        block_name();
        room_number();
        temp4=temp4->next;
    }
}

```

```

        printf("\nName: %s\nRegistration Number:
%s\nCGPA: %f\n",temp4->name,temp4-
>reg_no,temp4->CGPA);

```

```

        block_name();
        room_number();

```

```

    }

```

```

void asking_for_womens_hostel()

```

```

{
    struct list *temp4=head;

```

```

    printf("\n\t\t~~WOMEN'S HOSTEL
ALLOTMENT~~\n\n");

```

```

    while (temp4->next!=NULL)
    {

```

```

        printf("\nName: %s\nRegistration Number:
%s\nCGPA: %f\n",temp4->name,temp4-
>reg_no,temp4->CGPA);

```

```

        block_name();
        room_number();
        temp4=temp4->next;
    }

```

```

        printf("\nName: %s\nRegistration Number:
%s\nCGPA: %f\n",temp4->name,temp4-
>reg_no,temp4->CGPA);

```

```

        block_name();
        room_number();

```

```

    }

```

```

int main()

```

```

{

```

```

    int R,j,m,k;

```

```

    printf("\n\t\tWelcome to Hostel
Counseling.\n\n");

```

```

    while (j=1)
    {

```

```

        printf("\n\t(1) Fill Student information\n\t(2)
Rank List\n\t(3) Choose Hostel Block \n\t(4) Exit");

```

```

        printf("\nEnter your choice: ");

```

```

scanf("%d",&m);

```

```

switch(m)

```

```

{

```

```

    case 1:

```

```

        details();
        break;

```

```

    case 2:

```

```

        printf("\n=====");

```

```

        bubbleSort();
        display();

```

```

        printf("\n=====");

```

```

        break;

```

```

    case 3:

```

```

        bubbleSort();
        asking_for_mens_hostel();
        asking_for_womens_hostel();
        break;

```

```

    case 4:

```

```

        exit(0);

```

```

    }

```

```

}

```

```

}

```

## VII. OUTPUT

- Taking Student information

```
Select "C:\Users\Hemaksh\Desktop\Programs\C\Hostel Counselling systm using II\Hostel.exe"

Welcome to Hostel Counseling.

(1) Fill Student information
(2) Rank List
(3) Choose Hostel Block
(4) Exit
Enter your choice: 1

Enter what you want to do:
(1) Enter student details
(2) Exit
1

Enter your Name: Hemaksh

Enter you Registration number: 19BCE2222

Enter your CGPA: 8.53

Enter your gender [Put 'M' for Male and 'F' for Female]: m

Enter what you want to do:
(1) Enter student details
(2) Exit
```

- Generating Rank list

```
Select "C:\Users\Hemaksh\Desktop\Programs\C\Hostel Counselling systm using II\Hostel.exe"

(1) Fill Student information
(2) Rank List
(3) Choose Hostel Block
(4) Exit
Enter your choice: 2

=====
~~RANK LIST FOR MEN'S HOSTEL ALLOTMENT~~

Name: Hemaksh
Registration Number: 19BCE2222
CGPA: 8.530000

Name: Manav
Registration Number: 19BCE0642
CGPA: 8.200000

~~RANK LIST FOR WOMEN'S HOSTEL ALLOTMENT~~

Name: Shreya
Registration Number: 19BCE1002
CGPA: 9.000000

=====
(1) Fill Student information
(2) Rank List
(3) Choose Hostel Block
(4) Exit
```

- Room Allotting

```

"C:\Users\Hemaksh\Desktop\Programs\C\Hostel Counselling systm using ll\Hostel.exe"
=====
(1) Fill Student information
(2) Rank List
(3) Choose Hostel Block
(4) Exit
Enter your choice: 3

      ~~~MEN'S HOSTEL ALLOTMENT~~~

Name: Hemaksh
Registration Number: 19BCE2222
CGPA: 8.530000

Enter which block you want: J

The block allotted to you is:- J

The following are the options for your hostel:-
(1) A.C. room
(2) Non - A.C. room
1

Enter the number of sharing you want:
(1) 1 bedded sharing
(2) 2 bedded sharing
(3) 3 bedded sharing
(4) 6 bedded sharing
1

The room allotted to you is:- 1046

```

## VIII. CONCLUSION

This program makes the conduction of Hostel Counselling easy and efficient. It provides the user with the options to take student details, generate two separate rank lists one for women and another for women for the separation of room allotment and also allows him/her with the ability of Hostel block and room allotment on the student's preference. The objective of this program was to develop a code in which we show the implementation of Linked list while overcoming the limitations of an array such as fixed memory space and single data type.

The program achieves the projects objectives.

## IX. VIDEO EXPLAINING MY PROJECT

<https://www.loom.com/share/8de97a9e77e84c39bcf4d6356688e120>

## X. ACKNOWLEDGMENT

I would like to thank my professor Mrs Seetha R. in helping me make this project and guiding me through its process.

## XI. REFERENCES

- Introduction to algorithms – Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein
- Algorithms by Robert Sedgewick & Kevin Wayne